JOURNAL OF PROPERTY RESEARCH
2021, VOL. 38, NO. 1, 48–70
https://doi.org/10.1080/09599916.2020.1832558

Routledge
Taylor & Francis Group

OPEN ACCESS

Predicting property prices with machine learning algorithms

Winky K.O. Ho[a], Bo-Sin Tang[b] and Siu Wai Wong[c]

[a]Department of Real Estate and Construction, The University of Hong Kong, Hong Kong, China; [b]Department of Urban Planning and Design, The University of Hong Kong, Hong Kong, China; [c]Department of Building and Real Estate, The Hong Kong Polytechnic University, Hong Kong, China

ABSTRACT
This study uses three machine learning algorithms including, support vector machine (SVM), random forest (RF) and gradient boosting machine (GBM) in the appraisal of property prices. It applies these methods to examine a data sample of about 40,000 housing transactions in a period of over 18 years in Hong Kong, and then compares the results of these algorithms. In terms of predictive power, RF and GBM have achieved better performance when compared to SVM. The three performance metrics including mean squared error (MSE), root mean squared error (RMSE) and mean absolute percentage error (MAPE) associated with these two algorithms also unambiguously outperform those of SVM. However, our study has found that SVM is still a useful algorithm in data fitting because it can produce reasonably accurate predictions within a tight time constraint. Our conclusion is that machine learning offers a promising, alternative technique in property valuation and appraisal research especially in relation to property price prediction.

ARTICLE HISTORY
Received 27 February 2020
Accepted 1 October 2020

KEYWORDS
Machine Learning algorithms; SVM; RF; GBM; property valuation

1. Introduction

In the contemporary digital era, useful information of the society can be retrieved from a wide variety of sources and stored in the form of structured, unstructured and semi-structured formats. In the analysis of economic phenomena or social observations, advancement of innovative technology makes it possible to systematically extract the relevant information, transform them into complex data formats and structures, and then perform suitable analyses. Because of these new circumstances, traditional data processing and analytical tools may not be able to capture, process and analyse highly complex information in the social and economic worlds. New techniques have been developed in response to the treatment of the colossal amount of available data.

Machine learning is one of the cutting edge techniques that can be used to identify, interpret, and analyse hugely complicated data structures and patterns (Ngiam & Khor, 2019). It allows consequential learning and improves model predictions with a systematic input of more recent data (Harrington, 2012; Hastie et al., 2004). Contemporary research on machine learning is a subset of artificial intelligence (AI) that attempts to train computers with new knowledge through input of data, such as texts, images, numerical

CONTACT Bo-sin Tang bsbstang@hku.hk

values, and so on, and support its interaction with other computer networks. According to Feggella (2019), machine learning is about 'the science of getting computers to learn and act like humans do, and improve their learning over time in autonomous fashion, by feeding them data and information in the form of observations and real-world interactions'.

Machine learning can be broadly categorised into three types, namely supervised learning, unsupervised learning and semi-supervised learning. In essence, supervised machine learning algorithms intend to identify a function that is able to give accurate out-of-sample forecasts. For example, in property research, if an investigator intends to make forecast of housing prices y_i from its physical, neighbourhood and accessibility characteristics x_{ij} from a sample of n apartments, one can assume $\mathcal{L}(\widehat{y_i}, y_i)$ to be the prediction loss function. A machine learning algorithm will look for a function $\hat{f}$ that produces lowest expected prediction loss $E_{(y_i,x_{ij})}\left[\mathcal{L}\left(\hat{f}\left(x_{ij}\right), y_i\right)\right]$ on the *test* data from the same distribution (Mullainathan & Spiess, 2017). A model trained through supervised learning is said to be successful if it can make predictions within an acceptable level of accuracy. Examples of supervised learning include linear regression and support vector machine.

Unsupervised learning is a branch of machine learning that handles unlabelled training data. These data do not have any kind of label that can reveal their classification. When compared to supervised learning, unsupervised learning methods are applied to discover the relationships between elements in a dataset without labels. Because of this nature, unsupervised learning can discover hidden structures within data that may not be easily observed. To put it simply, it is best adopted if data scientists want to uncover patterns that may not have been identified theoretically prior to the investigation. The usefulness of unsupervised learning is to model the underlying structure or distribution in the data with a view to learning more about the data. Unsupervised learning problems can be categorised into clustering and association (i.e. discover rules that describe a large portion of data) problems. Examples of unsupervised learning algorithms include k-means for clustering problems, neural network for analysing sequences, and Apriori algorithm for association rule learning problems.

Semi-supervised learning refers to the case when there are a large amount of input data (X) and only some of the data are labelled (Y). In this case, data scientists can use unsupervised learning techniques to discover and learn about the data structure. On the contrary, data scientists can also adopt supervised learning techniques to make predictions for the unlabelled data, train the model with supervised learning algorithm, and make predictions on the test data. In the real world, most data analyses fall into this category.

Against this background, the mechanism of machine learning is drastically different from conventional econometric techniques in social and economic analyses. In econometrics research, economic models are built up to focus on a number of parameters that represent the effects of change in demand and/or supply factors. Researchers have to correct for heterogeneity and attempt to obtain heteroskedasticity-consistent standard errors and covariances for the estimated parameters (Einav & Levin, 2014). They also have to evaluate the robustness of the results by estimating several alternative model specifications. In contrast, machine learning approaches mostly put emphasis on the out-of-sample predictions, and rely on the data-driven model selection to unveil the features with the most explanatory power (Vespignani, 2009). These approaches work with uncertainty associated with incomplete information and/or measurement errors, and intend to model uncertainty via the tools and techniques from probability, such as the Bayesian models. Commonly used machine learning techniques include lasso (Mullainathan & Spiess, 2017), decision tree (Mullainathan & Spiess, 2017), random forest (Varian, 2014; Wang & Wu, 2018), support vector machine (Gu et al., 2011; Liu & Liu, 2010; Shinda & Gawande, 2018; Vilius & Antanas, 2011; Wang & Wu, 2018; Xie & Hu, 2007; Zurada et al., 2011; Zhong et al., 2009), gradient boosting machine (Ahmed & Abdel-Aty, 2013; Zhang & Haghani, 2015), *K* nearest neighbours (Mukhlishin et al., 2017), and artificial neural network (Limsombunchai et al., 2004; McCluskey et al., 2012; Mohd et al., 2019; Zurada et al., 2011).

Advanced machine learning techniques have already been applied in many domains, such as face recognition (Vapnik & Lerner, 1963), speech analysis (Jelinek, 1998; Jurafsky & Martin, 2008; Rabiner & Juang, 1993; Schroeder, 2004), gene identification (Golub et al., 1999; Krause et al., 2007), protein study (Cai et al., 2003; Rausch et al., 2005), tumour detection (Furey et al., 2000; Guyon et al., 2002; Ramaswamy et al., 2001) and diabetics (Yu et al., 2010). It is also increasingly popular to apply machine learning techniques in the appraisal and prediction of property values. The merit of this technique is that it allows more recent property data or transactions to possibly 'correct' the parameters of existing model estimations, which were based on earlier observations (Baldominos et al., 2018; Hastie et al., 2004; Hausler et al., 2018; Rafiei & Adeli, 2016; Rogers & Girolami, 2011; Sun et al., 2015).

In this paper, we choose three algorithms as examples to illustrate how machine learning algorithms can be utilised to predict housing prices with the input of conventional housing attributes. We use these techniques for a number of reasons. First, SVM is used because it works well with high dimensional data, semi-structured and unstructured data, and is seen as a more powerful tool to make accurate predictions (Hassanien et al., 2018). In the commercial world, the use of SVM is very popular in predicting a company's sale volume and revenue. Second, due to many decision trees participating in the process, random forest (RF) is used because it can reduce over-fitting of data. Previous property research has also demonstrated that random forest is a robust algorithm which provides accurate predictions (Mohd et al., 2019; Mullainathan & Spiess, 2017; Pérez-Rave et al., 2019). Third, gradient boosting machine (GBM) is also used because it is considered as an emerging machine learning algorithm with highly flexibility. It also provides better accuracy than many other machine learning algorithms, as suggested by Kaggle Data Science Competition (Kaggle, 2019).

This paper is organised as follows. Section 2 reviews the use of machine learning and the application of SVM, RF and GBM algorithms. Section 3 describes these methodologies and examines the algorithms, optimisation and parameters. Section 4 describes the data, its definitions and sources. Section 5 presents our empirical results based on these algorithms, and compares their results. The last section concludes the paper.

2. Literature review

Our literature review section comprises three parts. We first start our discussions about SVM. First, Support Vector Machine (SVM) has been developed from the Statistical

Learning Theory by Vapnik (1995) based upon the principle of structural risk minimisation. As an advanced machine learning algorithm, it has been extensively used to perform data classification, clustering and prediction. SVM is a supervised learning technique that can be considered as a new method for training classifiers based on polynomial functions, radial basis functions, neural networks, splines or other functions (Cortes & Vapnik, 1995). SVM uses a hyper-linear separating plane to create a classifier. If a simple separating plane fails, SVM can also perform a non-linear transformation of the original input space into a high dimensional feature space. Those separating planes should have a maximal margin to contain all the points with a very small error, and to provide spaces for new coming data (Rychetsky, 2001). Moreover, SVM can also be used to perform regression on continuous and categorical variables. In this case, it is termed as support vector regression (SVR).

SVM has been used by Boser et al. (1992) to recognise handwriting patterns. It has then become very popular in a wide variety of applications, including handwriting recognition, face detection and text classification. In property research, Li et al. (2009) have used support vector regression (SVR) to forecast property prices in China using quarterly data from 1998 to 2008. In their study, five features were selected to predict the property prices as the output variable of the SVR. Based on conventional evaluation criteria, such as the mean absolute error (MAE), the mean absolute percentage error (MAPE) and the root mean squared error (RMSE), they concluded that the SVR model is an excellent technique to predict property prices.

Rafiei and Adeli (2016) have used SVR to determine whether a property developer should build a new development or stop the construction at the beginning of a project based on the prediction of future housing prices. Using data from 350 condos built in Tehran (Iran) for the period between 1993 and 2008, their study trained a model with 26 features, such as ZIP code, gross floor area, lot area, estimated construction cost, duration of construction, property prices of nearby housing developments, currency exchange rate and demographic factors. Their results have demonstrated that SVR is an appropriate method to make predictions of housing prices since the prediction loss (error) is as low as 3.6% of the test data. The estimation results, therefore, provide valuable inputs to the decision-making of property developer.

Second, random forest approach was first proposed by Breiman (2001) by blending classification and regression tree (Breiman et al., 1984) and bootstrap aggregation (Breiman, 1996). It is an ensemble classifier or regressor that employs multiple models of *T* decision trees to obtain better prediction performance. This method creates many trees, and utilises a bootstrap technique to train each tree from the original sample set of training data. It searches for a random subset of features to obtain a split at each node. The bootstrap technique for each regression tree generation, together with the randomly selected features partitioned at each node, reduce the correlations between the generated regression trees. Hence, Random Forest averages the prediction responses to reduce the variance of the model errors (Pal, 2017).

Wang and Wu (2018) employ 27,649 housing assessment price data from Airlington county, Virginia USA during 2015, and suggest that Random Forest outperforms Linear Regression in terms of accuracy (see also Muralidharan et al., 2018). Based on a set of features, such as number of beds, floor level, building age and floor area, Mohd et al. (2019) utilise several machine learning algorithms to predict housing prices in Petaling

Jaya, Selangor, Malaysia. Their study compares the results using Random Forest, Decision Tree, Ridge Regression, Linear Regression and LASSO, and concludes that Random Forest is the most preferred one in terms of overall accuracy, as evaluated by the root mean squared error (RMSE).

Using 1,970 housing transaction records, Koktashev et al. (2019) attempt to estimate the housing values in the city of Krasnoyarsk. Their study considers the number of rooms, total area, floor, parking, type of repair, number of balconies, type of bathroom, number of elevators, garbage disposal, year of construction and the accident rate of the house as features. It applies random forest, ridge regression and linear regression to predict property prices. Their study concludes that random forest outperforms the other two algorithms, as evaluated by the mean absolute error (MAE).

Third, the idea of boosting algorithms has been developed by Breiman (1997, 1998) as numerical optimisation techniques in function space, and these algorithms can provide a solver to many classification and regression problems. Gradient boosted classifiers belong to a group of machine learning algorithms that combine many weak learning models together to create a strong predictive model, typically decision trees. Gradient boosting models have gained their popularity because they are effective in classifying complex datasets. These methods have recently been used to win many data science competitions, such as (Kaggle, 2019).

In property research, Park and Bae (2015) develop a housing price prediction model with machine learning algorithms, such as C4.5, RIPPER, Naïve Bayesian, and AdaBoost, and compare their performance in terms of classification accuracy. Their study aims to help property sellers or real estate agents make rational decisions in property transactions. The experiments demonstrate that the RIPPER algorithm, based on accuracy, consistently outperforms the other models in the performance of housing price prediction.

Satish et al. (2019) utilise several machine learning algorithms, namely linear regression, LASSO and gradient boosting algorithms, to predict house prices. Their results show that LASSO regression algorithm outperforms other algorithms in terms of accuracy. Utilising a sample of 89,412 housing transaction records from three counties in Los Angeles, California, Huang (2019) employs both linear and non-linear machine learning methods to predict the log error of the home prices. Incorporating features such as the number of bedrooms, number of bathrooms, estate condition, completion year, total tax assessed value, and so on, the study compares the results using the methods of linear regression, decision tree, gradient boosting, random forest, and support vector machine. Surprisingly, all these methods tend to underestimate the Zestimate prediction error. The study concludes that Zestimate has already provided a relative accurate housing price prediction.

Machine learning algorithms generate many benefits in research. On the one hand, machine learning techniques provide more flexible and sophisticated estimation procedures to manage and analyse a tremendously huge amount of data. On the other hand, large datasets sometimes contain extremely complex relationships among variables that linear model estimations in traditional estimation techniques, such as ordinary least square, may not be able to identify. Many machine learning algorithms, such as decision trees and support vector machine (SVM), allow investigators to capture and model complex relationships (Varian, 2014). There are many ways that we can make use of machine learning to perform data analysis, such as analysing demographic data, texts, and images, and then using them to make predictions.

There are several advantages of using machine learning methodologies to estimate property prices. First, unlike conventional econometrics models, machine learning algorithms do not require our training data to be normally distributed. Many statistical tests rely on the assumption of normality. If the data are not normally distributed, those statistical tests will fail and become invalid. Second, we will use the iterative process of stochastic gradient descent in SVM algorithms and grid search in RF or GBM. These processes used to take a long time in the past, but they can now be quickly completed with the high-speed computational power of modern computers, and thus the use of this technique today is less costly or time-consuming. Third, the estimation errors can be driven to the minimum level for both training and test data sets.

3. Methodology

3.1. Support vector machine

SVM is mostly used as a classifier to find hyperplane in an n-dimensional space (the number of features) that can separate objects of different categories while maximising the distance between data points of each category to the separating hyperplane (Noble, 2004). This classification method can be achieved within the same feature space or by projecting the features into a higher dimensional plane. Noble (2006) suggests that data are more easily separable in higher-dimensional spaces because there exists a kernel function for any data set to be linearly separated. However, the task of identifying the kernel function is normally completed by trial and error.

SVM can also be used as a regression method, namely support vector regression (SVR), that input x can be mapped onto an m-dimensional feature space employing nonlinear mapping, to construct a linear model in the feature space. A parameter $\in$ is introduced to ensure that the learned function does not deviate from the output variable by a value larger than the epsilon for each instance. The first step is to apply SVR to train the model with a supervised learning method. Equation (1) is specified as follows (For the detailed discussions of SVR, please see Basak et al., 2007; Mu et al., 2014):

$$\min \frac{1}{2} w^2 + C \sum_{i=1}^{n} \left(\xi_i + \xi_i^* \right).$$

$$s.t.\{ \begin{array}{c} y_i - f(x_i, w) \leq \in + \xi_i^* \\ f(x_i, w) - y_i \leq \in + \xi_i \\ \xi_i, \xi_i^* \geq 0, i = 1, , 2, \ldots, n \end{array} \tag{1}$$

where w^2 is the model complexity; $\in$ is the insensitive loss function; ξ_i is the slack variable that measures the degree of misclassification of the data point i; C is the cost parameter in the objective function. ξ_i is a non-zero and will be multiplied by a cost parameter C. Then the optimisation becomes a tradeoff between the maximum distance and the cost penalty.

The optimisation can be transformed into dual problem, and the solutions are specified as Equation (2):

$$f(x) = \sum_{i=1}^{n_{SV}} \left(\alpha_i - \alpha_i^*\right) K(x_i, x) s.t. \begin{cases} 0 \leq \alpha_i^* \leq C \\ 0 \leq \alpha_i \leq C \end{cases} \tag{2}$$

where n_{SV} is the number of support vectors; $K(x_i, x)$ is the kernel function, as stated in Equation (3).

$$K(x_i, x) = \sum_{j=1}^{m} g_j(x) g_j(x_i) \tag{3}$$

When we estimate the SVM, our data set is divided into k= 5 equal subsets (folds) where each fold will be used as a test set at some point of time. We start with the first subset (k = 1) to test the model, while the rest are used to train the model. In the second iteration, the second fold (k = 2) is taken as the test data, while the rest are taken as the train data. The process will repeat until each fold has been taken as the test data. Each iteration will give a R^2 score, and then we can compute their mean value in a bid to determine the overall accuracy of our model.

This resampling procedure is called cross-validation designed for evaluating machine learning models on a subsample or training set in machine learning terminology (say, 80% of the whole sample). This procedure attempts to utilise a training data set with a view to estimating how well the model generally predicts and then to make predictions utilising the test data set (the remaining 20% of the whole sample). Its goal is to define how many observations should be used to test the model in the training phase in order to minimise the issues like overfitting (the model performs well on training set but poorly on the test set) and underfitting (the model performs poorly in both training and test sets) As a result, data scientists can obtain some insights on how the model can be generalised to an independent data set. By employing k-fold cross-validation, we build a k different models so that all our data can be utilised for both training and testing while evaluating our algorithms on unseen data.

After computing the predicted values for our training data set by training our models in the form of supervised learning method with Python, we obtain $\theta = \left(X^T X\right)^{-1}\left(X^T y\right)$ that minimises the cost value for the training set. We then plug in the coefficients (or weights) into our models, using the test data to see whether our estimations are still accurate for the test data, as evaluated by the mean square error (*MSE*), root-mean-square error (*RMSE*), mean absolute percentage error (*MAPE*) and the coefficient of determination R^2. These three performance metrics (Equations (4)–(6)) range between 0 and ∞, and each of which indicates that the fit is perfect when it is estimated to be 0.

$$MSE = 1 \sum_{\mathrm{i}=1}^{m} \left(h\left(x^{(i)}\right) - y^{(i)}\right)^2 \tag{4}$$

$$RMSE = \sqrt{1 \sum_{\mathrm{i}=1}^{m} (h(x^{(i)}) - y^{(i)})^2} \tag{5}$$

$$MAPE = 100\% \sum_{\mathrm{i}=1}^{m} \left| \frac{\left(h\left(x^{(i)}\right) - y^{(i)}\right)}{y^{(i)}} \right| \tag{6}$$

where $h\left(x^{(i)}\right)$ represents the predicted value of property; $y^{(i)}$ represents the actual value of property; and m represents the number of observations in the test data.

Estimating the SVM will provide us with a R^2 for test set of our model. Following this stage (usually labelled as model validation) will be the estimation of stochastic gradient descent (SGD) to minimise the errors (prediction loss in machine learning terminology) while predicting the property prices. It will be even better if the training data size is very large since employing stochastic gradient descent will approximate the true gradient by performing the parameters update of a single training example at a time. Actually, it is a variant of gradient descent (GD).

SGD updates the weights based on the sum of accumulated errors over all sample $x^{(i)}$ through Δw. Equation (7) implies that we update the weights with a single training sample at one time instead of the full training set, and this algorithm performs the update for each training data set. The update will continue until the algorithm converges.

$$\Delta w = -\alpha \quad J = \alpha\left(y^{(i)} - \tau\left(w^T x\right)^{(i)}\right)x^{(i)} \tag{7}$$

where $-\alpha \quad J$ is the stochastic gradient descent. To implement the stochastic gradient descent, we need to specify a learning rate α. In this study, we choose an adaptive learning rate which does not have to be a fixed value. In fact, it is a small initial value, and increases according to the identification procedure during the training process. Equation (8) specifies the adaptive learning rate used in our estimation. It starts with an arbitrary initial value $e(0)$, and the learning rate $e(t)$ will converge to zero within a finite time period.

$$\dot{\alpha} = \gamma(I+2)|e| - v\gamma\alpha, \ 0 < \gamma, v \tag{8}$$

where γ and v are slightly larger than zero. Once we have computed the predicted values for our training data set, we can obtain $\theta = \left(X^T X\right)^{-1}\left(X^T y\right)$ that minimises the cost value for the training set. The estimated weights can then be plugged into our models employing the test data to obtain the weight for each feature, and R^2 for the test data. Then, we can see whether our estimations are still accurate for the test data, as evaluated by the MSE, RMSE and MAPE. These criteria are is shown in Equations (4)–(6).

3.2. Random forest

Random Forest is a supervised learning algorithm that employs ensemble learning method for classification and regression. It runs n number of regression trees, and combines them into a single model to make more accurate prediction than one single tree. RF constructs many decision trees at training, and predictions from all trees are combined to make the final prediction. Employing random sampling with replacement (bagging in machine learning terminology), RF helps data scientists reduce the variance associated with those algorithms that have high variance, typically decision trees. Given a training set of feature X and output Y, bagging repeatedly selects a random sample of the training set for β times $(b = 1, 2, \ldots, \beta)$ and fits the trees to these samples.

For every tree, we obtain a sequence of instances which are randomly sampled replacement from the training set. Each sequence of instances corresponds to a random vector $\emptyset_k$ forming a specific tree. Since all the sequences will not be exactly the same, the decision trees constructed from them will also be slightly variant. De Aquino Afonso et al. (2020) suggest that the prediction of the K-th tree for an input X can be represented by Equation (9):

$$h_k(X) = h(X, \emptyset_k), \quad k \in \{1, 2, \ldots, K\} \tag{9}$$

where K is the number of trees. As a tree splits, each of which randomly selects features to avoid correlations among features. Alpaydin (2009) points out that a node S can be split into two subsets, $S1$ and $S2$ by selecting a threshold c that minimises the difference in the sum of squared errors.

$$SSE = \left(\sum_{i \in S_i} \left(v_i - \frac{1}{|S_1|} \sum_{i \in S_1} v_i \right)^2 + \sum_{i: i \in S_2} \left(v_i - \frac{1}{|S_2|} \sum_{i \in S_2} v_i \right)^2 \right) \tag{10}$$

By following the same decision rules, we can predict any subtree as the mean or median output of instances. Finally, we can obtain the final prediction as an average of each tree's output, as stated in Equation (11):

$$h(X) = \frac{1}{K} \sum_{i=1}^{K} h_k(X) \tag{11}$$

3.3. Gradient boosting machine

Gradient boosting is a machine learning technique that produces a prediction model in the form of an ensemble of weak prediction models, typically decision trees. The concept of boosting is to transform a feature that is not strong in predicting outcome Y (weak learner) into a strong learner. After identifying weak learners in each regression tree, boosting will give an equal weight to each observation while focusing on the errors between the fitted values predicted by the weak learner and the actual values. With gradient boosting, this model employs the error rate to compute the gradient of the error function, adopting the gradient to determine how to tune the model parameters in order to reduce the error with each iteration. This process will be carried out m times, as specified by data scientists.

Under the gradient boosting method, Y is assumed to be a real value. This method attempts to make an approximation $\left(\hat{F}(x)\right)$ to the weighted sum of functions from weak learners $(h_i(x))$:

$$\hat{F}(x) = \alpha + \sum_{i=1}^{M} \theta_i h_i(x) \tag{12}$$

Under the empirical risk minimisation principle, we can employ Equation (13) to search for an approximation that minimises the mean value of the loss function employing the training set. We can start with a model with a constant function, and then transform it into Equation (14):

$$F_0(x) = \underset{\theta}{\operatorname{argmin}} \sum_{i=1}^{n} L(y_i, \theta) \tag{13}$$

$$F_m(x) = F_{m-1}(x) + \underset{h_m \in \mathcal{H}}{\operatorname{argmin}} \left[\sum_{i=1}^{n} L(y_i, F_{m-1}(x_i) + h_m(x_i)) \right] \tag{14}$$

where $h_m \in \mathcal{H}$ is a base learner function. Swathi and Shravani (2019) suggest that although the best function h at each step for the loss function L cannot be specified, data scientists can apply a steepest descent step to this minimisation problem. It is assumed to be the set of arbitrary differentiable function on, the model can be updated with the following equations:

$$F_m(x) = F_{m-1}(x) - \theta_m \sum_{i=1}^{n} \quad {}_{F_{m-1}} L(y_i, F_{m-1}(x_i)) \tag{15}$$

$$\theta_m = \underset{\theta}{\operatorname{argmin}} \sum_{i=1}^{n} L(y_i, F_{m-1}(x_i) - \theta \quad {}_{F_{m-1}} L(y_i, F_{m-1}(x_i))) \tag{16}$$

where the derivatives are taken with respect to the functions F_i for $i \in \{1, \ldots, m\}$, θ_m is the step length.

However, this approach will offer an approximation to be minimisationa problem only. Estimation of GBM can provide a R^2 for the test set of a model, and its performance can be evaluated by MSE, RMSE and MAPE, respectively. Finally, we can use grid search to improve the accuracy of our model by tuning the hyperparameters.

To conclude, we summarise the advantages and disadvantages of support vector machine, random forest and gradient boosting machine in Table 1.

4. Data sources

Machine learning algorithms are used to predict property prices in a housing district of Hong Kong. Conventional housing attributes that influence housing prices are included in the models. The variables in our price estimation models are defined as follows:

LRP_i represents the transaction price (total consideration) of residential property i, which is measured in HK dollars, in natural logarithm.

$LGFA_i$ is defined as the floor area of property i, in natural logarithm.

$LAGE_i$ represents the age of residential property i in years, which can be measured by the difference between the date of issue of the occupation permit and the date of transaction, in natural logarithm.

FL_i is the actual floor level of property i.

$LCENTRAL_i$ is a proxy for accessibility of residential property i. It measures the time spent on travelling from a specific housing estate to the Central District, which is measured in minutes, and in natural logarithm.

$E_i, S_i, W_i, N_i, NE_i, SE_i, SW_i \& NW_i$ are the orientation of property i are facing. Orientation is divided into eight categories: East, South, West, North, Northeast, Southeast, Southwest and Northwest, respectively. It represents the direction property

Table 1. Advantages and disadvantages of SVM, RF and GBM.

Support Vector Machine	1. Works well with unstructured and semi structured data. 2. Works well with high dimensional data. 3. Solve any complex problem with appropriate kernel. 4. The risk of overfitting is less.	1. Longer training time is required than linear regression. 2. Weights for the features are not meaningful. 3. Not perform well when the number of features exceeds the number of training set. 4. Not suitable for large data set.
Random Forest	1. Provide reliable estimates for feature importance. 2. Provide efficient estimates for the test set even without hyperparameter tuning. 3. Provide accurate predictions. 4. Works well with missing values by substituting the feature appearing the most in a node for them.	1. Longer computation time than SVM. 2. A lot of computer memory are required. 3. Overfit with noisy classification and regression. 4. Not easy to interpret.
Gradient Boosting Machine	1. Provide accurate prediction. 2. Provide several hyper-parameter tuning options which make the function fit flexible. 3. Resolves the problems of multicollinearity where the correlations among the features are high. 4. Work well with missing data.	1. Longer computation time than SVM. 2. A lot of computer memory are required. 3. Sensitive to outliers. 4. Not easy to interpret.

Sources: Authors' summary and adaptations from UC Business Analyst (2018) and Corporate Finance Institute (2020).

i is facing, respectively; for they equal 1 if a property is facing a particular direction, 0 otherwise. The omitted category is Northwest so that coefficients may be interpreted relative to this category.

Table 2 presents the descriptive statistics for the data employed in this paper. To minimise the influence of external factors on property prices, our dataset only includes the property transaction records in the secondary market from 14 private residential estates located in a highly dense residential district called Tseung Kwan O district in Hong Kong (Figures 1 and 2). All these estates are either located very close to or on the top of metro stations. Our housing transaction records are obtained from the public domain. Relevant information includes data about total consideration of transacted properties stated in the sale and purchase agreements, date of transaction, physical and locational characteristics of the apartments. Incomplete and abnormal records are removed from our dataset. This data preparation process results in 39,554 housing transactions from June 1996 through August 2014. The price variable is defined in real terms which is calculated by deflating the series by the monthly price deflator series (with the base year of 1999–2000 = 100). This price deflator is published by the Rating and Valuation Department, Hong Kong Special Administrative Government.

5. Results and discussions

In machine learning, it is a general practice to rely on a correlation matrix to decide what features to be incorporated into our models. Table 3 presents the correlations between LRP_i and each feature, and shows that housing floor area has the largest correlation with prices, followed by property age, travelling time to Central District and floor level. In comparison, correlations between each orientation and prices are very close to zero and so they are excluded from our estimation. Then, we present the data visualisation to portray the relationship between property prices and each selected feature in Figure 3. All the charts exhibit the expected relationship between the features and the housing prices.

Table 2. Descriptive Statistics.

	LRP	LGFA	LAGE	FL	LCENTRAL	E	S	W	N	NE	SE	SW	NW
Count	39,544	39,544	39,544	39,544	39,544	39,544	39,544	39,544	39,544	39,544	39,544	39,544	39,544
Mean	14.85932	6.54152	1.73383	27.31249	3.33165	0.04223	0.03606	0.04188	0.03285	0.22497	0.18595	0.22345	0.21262
Std	0.28629	0.21346	0.73451	15.60989	0.07801	0.20112	0.18645	0.20031	0.17825	0.41756	0.38907	0.41656	0.40917
Min	13.43933	6.12249	0.22314	1	3.21888	0	0	0	0	0	0	0	0
25%	14.64568	6.38351	1.19392	14	3.29584	0	0	0	0	0	0	0	0
50%	14.84376	6.51915	1.87487	27	3.3673	0	0	0	0	0	0	0	0
75%	15.03858	6.72623	2.32435	40	3.4012	0	0	0	0	0	0	0	0
Max	17.01961	7.19519	3.06619	76	3.52636	1	1	1	1	1	1	1	1
Skew	0.49481	0.47397	0.46611	0.16563	0.13502	4.55244	4.97695	4.57433	5.24194	1.31740	1.61448	1.32786	1.40475

Figure 1. Residential District under Study.

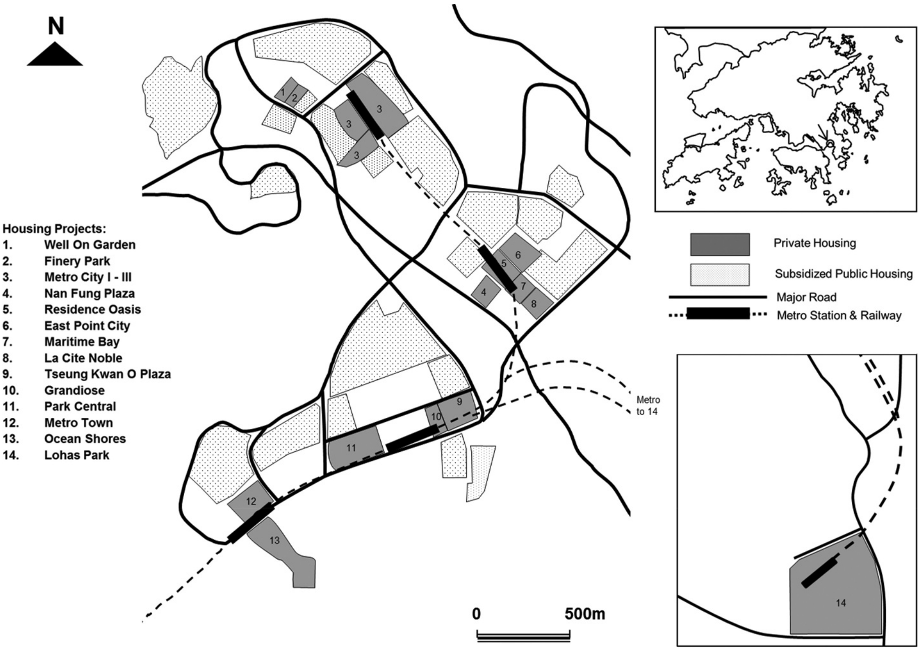

Figure 2. Location Plan of Private Housing Estates under Study.

Table 4 presents the estimations of R^2 for the test set under SVM, RF and GBM algorithms, together with the performance metrics: MSE, RMSE and MAPE. Our

Table 3. Correlation Matrix.

	LRP	LGFA	LAGE	FL	LCENTRAL	E	S	W	N	NE	SE	SW	NW
LRP	1.00000	0.84094	−0.39962	0.28535	−0.37285	0.01138	0.10823	0.00623	−0.03440	−0.03490	0.05026	−0.00662	−0.04842
LGFA	0.84094	1.00000	−0.27203	0.14673	0.08690	0.01028	0.09553	0.00038	−0.00276	−0.06329	0.08272	−0.07403	0.011373
LAGE	−0.39962	−0.27203	1.00000	−0.20894	0.14738	0.01629	0.02560	0.01336	0.04687	0.00636	−0.04232	0.00219	−0.01511
FL	0.28535	0.14673	−0.20894	1.00000	−0.11582	−0.00154	−0.00618	0.02043	−0.01241	0.00330	−0.01001	−0.00679	0.01204
LCENTRAL	−0.37285	−0.08690	0.14738	−0.11582	1.00000	−0.02565	−0.02770	−0.03984	−0.01120	−0.05598	0.10577	−0.05232	0.05944
E	0.01138	0.01028	0.01629	−0.00154	−0.02565	1.00000	−0.04062	−0.04390	−0.03870	−0.11313	−0.10036	−0.11264	−0.10912
S	0.10823	0.09553	0.02560	−0.00618	−0.02770	−0.04062	1.00000	−0.04044	−0.03565	−0.10421	−0.09244	−0.10375	−0.10051
W	0.00623	0.00038	0.01336	0.02043	−0.03984	−0.04390	−0.04044	1.00000	−0.03853	−0.11264	−0.09992	−0.11215	−0.10864
N	−0.03440	−0.00276	0.04686	−0.01241	−0.01120	−0.03870	−0.03565	−0.03853	1.00000	−0.09929	−0.08808	−0.09886	−0.09577
NE	−0.03490	−0.06329	0.00636	0.00330	−0.05598	−0.11313	−0.10421	−0.11264	−0.09929	1.00000	−0.25749	−0.28900	−0.27997
SE	0.05026	0.08272	−0.04232	−0.01001	0.10577	−0.10036	−0.09244	−0.09992	−0.08808	−0.25749	1.00000	−0.25637	−0.24836
SW	−0.00662	−0.07403	0.00219	−0.00679	−0.05232	−0.11264	−0.10375	−0.11215	−0.09886	−0.28900	−0.25637	1.00000	−0.27875
NW	−0.04842	0.01373	−0.01511	0.01204	0.05944	−0.10912	−0.10051	−0.10864	−0.09577	−0.27997	−0.24836	−0.27875	1.00000

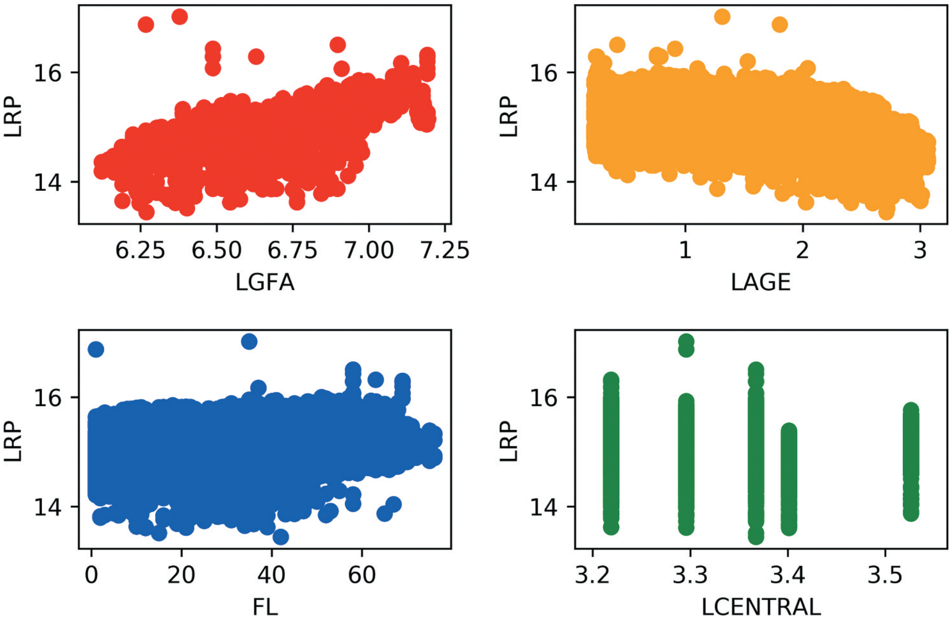

Figure 3. Data visualisation.

Table 4. Estimated results based on SVM, RF and GBM.

	R^2	MSE	RMSE	MAPE
Support Vector Machine	0.82715	0.01422	0.11925	0.54467%
Random Forest	0.90333	0.00795	0.08918	0.32270%
Gradient Boosting Machine	0.90365	0.00793	0.08903	0.32251%

discussion first starts with SVM. The best R^2 score for a linear SVR model $(C = 1)$ is 0.82702 for the test set while the intercept and estimators are as follows:

$Intercept$: 14.85848; $LGFA_i$: 0.21911; $LAGE_i$: −0.03580; FL_i : 0.03334; $LCENTRAL_i$: -0.08108

The results are then evaluated by the MSE, RMSE and MAPE criteria. These values are estimated to be 0.01423, 0.11929 and 0.54571%, demonstrating that SVM fits our data very well. Following this is the estimation of stochastic gradient descent. Employing 5 CV, it takes 37 epochs to converge in 0.05 seconds. The intercept and estimators in the SGD model are as follows:

$Intercept$: 14.85929; $LGFA_i$: 0.21966; $LAGE_i$: −0.03618; FL_i : 0.03323; $LCENTRAL_i$: -0.07880

Stochastic gradient descent provides an explanatory power R^2 of 0.82715 for the test set. The model is also evaluated by the same MSE, RMSE and MAPE criteria which are estimated to be 0.01422, 0.11925 and 0.54467%, respectively, implying a reasonable good fit. It is quite tempting to interpret the estimated weight for each feature in terms of elasticity. However, it is not the case in stochastic gradient descent. The estimated weights should not be considered as the magnitude of features' effects on real estate prices. In an iterative algorithm, the 'optimal' weight for each feature is indeed obtained by fine tuning an arbitrary value on a function, and going down its slope step by step until it reaches the lowest point of the function. Hence, our focus should concentrate on the explanatory power R^2 and the error minimisation of the model.

Based on our estimation results, Figure 4 portrays the scatterplot of property prices in logs LRP_i and the residuals. It shows that SVM fits the data very well most of the time. The model underperforms with high bias for extreme values in LRP_i (Note the green dots and orange dots indicating values are larger than 16.0 for housing prices). Figure 4 further shows the relationship between actual prices LRP_i and their predicted values $\widehat{LRP}_i$. We can see that, most of our predicted values follow closely to the red line, demonstrating that our model fits our data sufficiently well.

Using random forest, our base model provides a R^2 that is as high as 0.89690 for the test set while the MSE, RMSE and MAPE are estimated to be 0.00848, 0.09210 and 0.33348%, respectively. In random forest, the estimation of SGD is not feasible because some of the hyperparameters are not continuous, such as the number of estimators. Rather, we can employ the grid search to minimise the model errors by tuning the hyperparameters.

For hyperparameter tuning, we perform many iterations of the entire fivefold cross-validation process, each of which sets different model settings, such as the number of estimators, min samples split, min sample leaf, max features, and max depth. Then, we compare all these models, pick up the best one, train it on the training set, and then evaluate on the test set.

By using the grid search with 5 CV, we obtain a R^2 of 0.90333 for the test set (slightly higher than the base model). The MSE, RMSE and MAPE are estimated to be 0.00795, 0.08918 and 0.32270%, which are lower than those of the base model. Hence, we take the results based on grid search as the best results associated with random forest.

Figure 4 shows the scatterplot of property prices in logs LRP_i and the residuals in RF. It is noticeable that two dots (whose values are larger than 16 for housing prices) are lying far away from the clustering, our model fits the data very well most of the time. The

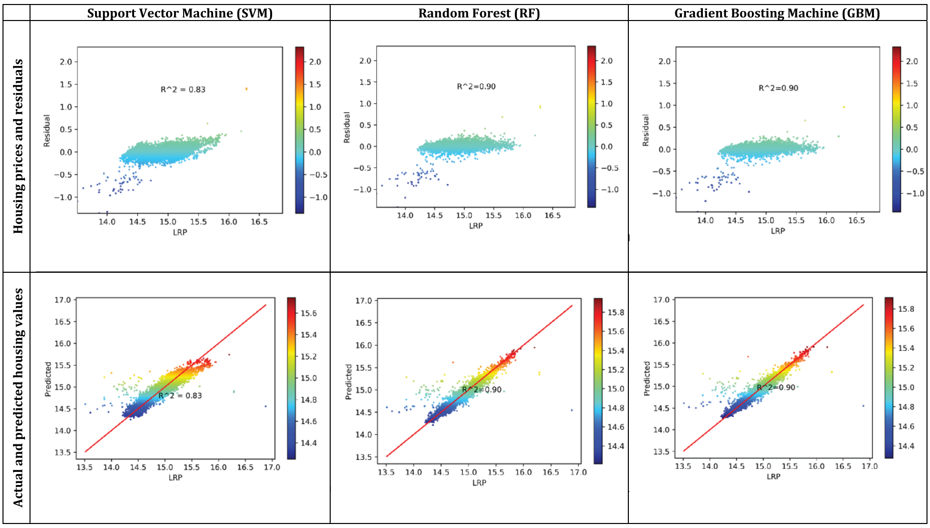

Figure 4. Comparison of Model Estimations.

model underperforms with high bias for extreme values in LRP_i (i.e. one yellow dot and one blue dot). We observe that most of our predicted values are lying very close or even on the red line, implying a very good fit of the data. Figure 4 shows that most of the predicted values lie closer to the red line in RF than SVM, implying that RF fits our data better than the SVM.

Lastly, we discuss the results based on the gradient boosting machine (GBM), more precisely gradient boosting regression. In the base model, GBM provides a R^2 of 0.89431 for the test set while the MSE, RMSE and MAPE are estimated to be 0.00870, 0.09325 and 0.35700%. It is very obvious that GBM fits our data reasonably well. In order to obtain even better results, we then employ the grid search with 5 CV to minimise the model errors by tuning the hyperparameters, including subsample, learning rates, the number of estimators, min samples split, min sample leaf, max features, and max depth. We then obtain a R^2 as high as 0.90365 for the test set while the MSE, RMSE and MAPE are estimated to be 0.00793, 0.08903 and 0.32251%, respectively. The results based on grid search are the best results associated with gradient boosting regression.

For most of the time, GBM also fits the data very well (see Figure 4). However, the model does not perform well with high bias for some extreme values in LRP_i in which two dots (whose values are larger than 16 for housing prices) are lying far away from the cluster. Again, the model has achieved very good model fitting since most of the predicted values are lying very close or around the red line, and its performance is slightly better than those of RF in relation to the MSE, RMSE and MAPE criteria.

Finally, we compare the results of these estimation techniques. In terms of explanatory power, this study shows that RF and GBM have equally good performances. In terms of error minimisation, GBM is slightly better than RF with respect to the three performance metrics while it outperforms SVM. Hence, we demonstrate that RF and GBM are very powerful tools for making accurate prediction of property prices, since the performances of these two algorithms are similar to each other. In comparison, the performance of SVM is found to be below that of RF and GBM in all aspects. However, this does not imply that SVM is necessarily inferior.

The existing literature has demonstrated mixed results about which algorithm is superior in making predictions. Previous studies suggest that artificial neural network (ANN) has a better performance than RF (see Masias et al., 2016; Mimis et al., 2013), but other studies demonstrate that both models have comparable predictive power and almost equally applicable in making predictions (Ahmad et al., 2017). In another study about the forecast of daily lake water levels, RF is found to exhibit the best results when compared to linear regression, SVM and ANN with respect to R^2 and RMSE (Li et al., 2016). Utilising RF, stochastic gradient boosting and SVM to predict genomic breeding values, Ogutu et al. (2011) conclude that boosting has had the best performance, followed by SVM and RF. In this study, SVM performs better than RF (see also Caruana & Niculescu-Mizil, 2006).

Hence, it is difficult to conclude which algorithm is the best in all cases because the results tend to be data-specific. What criteria should property valuers consider when choosing an algorithm to make predictions? We suggest that the choice of algorithm depends on a number of factors including, the size of data set, computing power, time constraint and researcher's knowledge about machine learning. If the data set is large and keeps increasing, SVM should be a good choice because its computation procedure takes less time than the other algorithms, especially when computing speed is a major concern. If the accuracy of prediction is the priority consideration, we recommend property valuers to utilise ANN, RF or GBM, although their computations often take much longer times than SVM. Needless to say, it is always a good practice to use more than one algorithms to compare the predictions.

6. Conclusions

Improvement in computing technology has made it possible to examine social information that cannot previously be captured, processed and analysed. New analytical techniques of machine learning can be used in property research. This study is an exploratory attempt to use three machine learning algorithms in estimating housing prices, and then compare their results.

In this study, our models are trained with 18-year of housing property data utilising stochastic gradient descent (SGD) based support vector regression (SVM), random forest (RF) and gradient boosting machine (GBM). We have demonstrated that advanced machine learning algorithms can achieve very accurate prediction of property prices, as evaluated by the performance metrics. Given our dataset used in this paper, our main conclusion is that RF and GBM are able to generate comparably accurate price estimations with lower prediction errors, compared with the SVM results.

First, our study has shown that advanced machine learning algorithms like SVM, RF and GBM, are promising tools for property researchers to use in housing price predictions. However, we must be cautious that these machine learning tools also have their own limitations. There are often many potential features for researchers to choose and include in the models so that a very careful feature selection is essential.

Second, many conventional estimation methods produce reasonably good estimates of the coefficients that unveil the relationship between output variable and predictor variables. These methods are intended to explain the real-world phenomena and to make predictions, respectively. They are used for developing and testing theories to perform causal explanation, prediction, and description (Shmuell, 2010). Based on these estimates, investigators can interpret the results and make policy recommendations. However, machine learning algorithms are often not developed to achieve these purposes. Although machine learning can produce model predictions with tremendously low errors, the estimated coefficients (or weights, in machine learning terminology) derived by the models may sometimes make it hard for interpretation.

Third, the computation of machine learning algorithms often takes much longer time than conventional methods such as hedonic pricing model. The choice of algorithm depends on consideration of a number of factors such as the size of the data set, computing power of the equipment, and the availability of waiting time for the results. We recommend property valuers and researchers to use SVM for making forecasts if speed is a primary concern. When predictive accuracy is a key objective, RF and GBM should be considered instead.

To conclude, the application of machine learning in property research is still at an early stage. We hope this study has moved a small step ahead in providing some methodological and empirical contributions to property appraisal, and presenting an