International Journal of Strategic Property Management
ISSN: 1648-715X / eISSN: 1648-9179
2022 Volume 26 Issue 5: 362–384
https://doi.org/10.3846/ijspm.2022.17909

COMBINATION OF MACHINE LEARNING-BASED AUTOMATIC VALUATION MODELS FOR RESIDENTIAL PROPERTIES IN SOUTH KOREA

Jengei HONG[1], Woo-sung KIM [2*]

[1] *School of Management and Economics, Handong Global University, Pohang, Republic of Korea*
[2] *School of Business, Konkuk University, Seoul, Republic of Korea*

Received 05 June 2022; accepted 25 October 2022

Abstract. The applicability of machine learning (ML) techniques has recently been expanding to include automatic real estate valuation models. The main advantage of this technique is that it can better capture complexity in the value determination process. Therefore, the performance of these techniques is shown to be superior to conventional models. In this paper, the latest ML algorithms (i.e., support vector machine, random forest, XGBoost, LightGBM, and CatBoost algorithms) are examined as automatic valuation models, and several combination methods are proposed to improve the models' predictive power. We applied ML models to approximately 57,000 records on apartment transactions, which were provided by South Korea's Ministry of Land, Infrastructure, and Transport, that occurred in Seoul in 2018. The results are as follows. First, ML-based predictors (especially, the latest decision tree-based algorithms) are more performative than conventional models. Second, the prediction error from a model can be partially offset by another model's error, which implies that an efficient averaging of the predictors improves their predictive accuracy. Third, the models' relative performance may be relearned by the ML algorithms, which means that they can also be used to recommend which algorithm should be selected for making predictions.

Keywords: automatic valuation model, mass appraisal, machine learning (ML) techniques, combined approach, decision tree-based algorithms.

Introduction

In a perfect market with homogeneous commodities, price changes would be observable, and price valuations would be accurate (Chau et al., 2005). However, a residential property is a spatially immobile, durable, expensive, and highly heterogeneous commodity (McClusky et al., 2000). Also, various factors such as market segmentation and government intervention can affect the real estate prices, so it has been a challenge to construct accurate and dependable automatic valuation models that capture those characteristics causing the complexity. In Glumac and Rosiers (2021), the purpose of the valuation model is "to provide a credible, reliable, and cost-effective estimate of market value as of a given point in time". Nevertheless, for practical purposes, such as local tax estimates, portfolio risk assessment, insurance risk assessment, lending risk, and land planning, appraising real estate accurately is essential. Moreover, international banking standards, such as the Basel accords, require real estate to be valued more frequently than before (Hong et al., 2020; Pi-ying, 2011). Because of its importance, studies have been conducted on the development of automated valuation models and their applications (Wang & Li, 2019).

The classical automated valuation method is the hedonic pricing model using multiple linear regression analysis. The advantage of this model is its simplicity and computational tractability: It is not difficult to calculate and interpret the regression coefficients of a linear regression model to explore how the value of a property varies with its attributes. Wang and Li (2019) state that "[b]ecause the target of mass appraisal is a large number of properties, and the valuation results need to be explained to the public, the basic needs are convenient operation and simple understanding". Because of these advantages, studies have constructed valuation models based on the hedonic pricing model. However, it has been pointed out in the literature that the model oversimplifies real estate market behavior to increase interpretability and analytical tractability. Most models assume a separable preference, perfect competition, market equilibrium, and an integrated market–qualities that do not reflect the characteristics

*Corresponding author. E-mail: *kimws@konkuk.ac.kr* of the actual market (Chau & Chin, 2003; Malpezzi, 2003; Sheppard, 1999). These assumptions weaken the model's predictive accuracy. Zurada et al. (2011) established that this model might yield imprecise coefficients because of functional form misspecification, multicollinearity, and non-linearity.

In this situation, the applicability of non-parametric machine learning (ML) methods has been rapidly growing because of significant advances in computational and data collection techniques, and numerous valuation models applying ML techniques have been proposed for real estate appraisal (Antipov & Pokryshevskaya, 2012; Čeh et al., 2018; Fan et al., 2006; Pace & Hayunga, 2020; Selim, 2009). A main strength of the ML technique is its predictive power. In most studies, the predictive power of the technique is reported to be significantly greater than that of the hedonic model, assuming a functional form. The technique does not require the model to take a pre-specified form. The non-parametric ML technique does not explicitly express the relationship between variables (i.e., it is less interpretable), but it can be constructed independently without specific assumptions about the relationship between the independent variables and between the independent and dependent variables. Therefore, attempts have been made to construct valuation models using the ML technique. For information on ML-based mass appraisal real estate models, please refer to the review paper by Wang and Li (2019).

Previous computer science research has reported that a more accurate predictor can be obtained when compared with individual models when the combination of models is properly constructed because they can compensate for errors caused by each model's characteristics. In other fields, such as computer science, various methods of comprising a committee (combining predictors) have been introduced, including averages (Taniguchi & Tresp, 1997), weighted averages (Krogh & Vedelsby, 1995; Merz & Pazzani, 1996), aggregation by NNs (Liu, 2005; Verikas et al., 2002), and probabilistic aggregation (Kittler et al., 1998). The design of the combination, which models to select, and how to combine them are important because the predictive accuracy of the combined model may further decline due to predictors with low predictive power. Although there are many studies (i.e., the aforementioned literature) on prediction or classification problems using combined models in other fields, there are few studies on developing an automated valuation model using the combination of predictors. Our research is motivated by the lack of results. Thus, the goal of our research is to investigate the feature of combined predictors for real estate mass appraisal based on a proper volume or coverage of data.

In this paper, the combination of predictive models is proposed to obtain a model with a high level of predictive accuracy. Three methods of combining predictive models are presented and examined. The first method is that of performing predictions by averaging the values obtained from each predictive model. The results show that the performance of averaging predictors was better than the average of the performance measure of predictors combined; its predictive performance approximate to the most performative model. The second is that of assigning different weights according to each model's performance. If the performance of a model is too poor, the model may be automatically excluded from the combination. In the third method, a model with a high level of predictive power is selected by re-learning the pattern of the errors. That is to say, after learning training data through various ML techniques, the error is derived by calculating the difference between the predicted value and the actual value when each algorithm is applied. The ML technique is then reapplied to learn which model exhibits the highest predictive power for an observation with a certain characteristic. This is done to select a highly predictive model according to the characteristics of each property. We demonstrate that the combination of models obtained by parameterizing the difference in performance of each model or by re-learning the error pattern of each model provides a superior result for mass appraisals. The algorithms employed in this paper are the support vector regression (SVR), Random Forest (RF), XGBoost, LightGBM, and CatBoost models. The SVR and RF algorithms have been widely used in other studies, but XGBoost, LightGBM, and CatBoost are recently proposed algorithms, and few studies have used the algorithms for mass appraisal problems.

Our contributions are as follow. First, as mentioned in many previous studies, the ML-based predictors are more performative than the ordinary least squares (OLS) regression-based predictor is. While the mean absolte percentage error (MAPE), R^2 values, coefficient of dispersion (COD) of the OLS-based predictor were 11.864, 0.898 and 11.892, respectively, the performance measure of the ML-based predictors was found to be superior. Especially, the recently developed boosted tree-based algorithms (i.e., XGBoost, LightGBM, and CatBoost) show sufficiently accurate performance for them to be applied to practical mass real estate appraisal. The MAPE, R^2 values and COD of the most performative model, CatBoost model, were 4.485, 0.978 and 4.487, respectively. The standard deviation of the absolute percentage error was also 11.429 for the OLS-based predictor, whereas 4.924 for CatBoost, which showed that the percentage error of ML-based predictor had less variability. Also, the probability of the CatBoost model predictions deviating more than 25% from the actual price was only 0.843%, while that for OLS-based predictions was 9.525%. Hong et al. (2020) mentioned that "It is important to obtain an accurate estimation of the value of a house whose market price is not observed in order to construct a reliable house price index or to conduct a successful mass appraisal". Our results suggests that the ML based predictors could be an alternative to the conventional OLS-based predictors in the development of mass appraisal models or house price indices. Second, we demonstrate that the predictive performance of the combination of single predictors could be improved. The result indicates that it is helpful to improve the predictive accuracy by taking a simple average of the predictors. The method of simply taking the average of the prediction values of single predictors shows similar performance to the best predictor, and the MAPE value, R^2 values, and COD values were 4.505, 0.977, and 4.504, respectively. The standard deviation of the absolute percentage error was found to be 4.918. The percentage of prediction error exceeding 50% and 100% of the actual value was 0.052% and 0% in the naive averaging model, whereas 0.087% and 0.08% in CatBoost model. In addition, we demonstrate that the combination of models obtained from the weighted averaging or by re-learning the error pattern provides more accurate predictions. The MAPE value, R^2 values, and COD values of the combined model constructed using the weighted average and ML-based voting techniques were 4.408, 0.978, 4.407 and 4.378, 0.977, 4.389, respectively. This result is evidence that if the prediction errors from single predictors can be eliminated in the combined model, the predictive performance may be improved. Moreover, the major advantage of these methods is their reliability. From a practical point of view, it will be an important issue to decide which technique to apply when employing machine learning techniques for mass appraisal problems. The combined model presented in this study learns the performance of single predictors according to features. After that, when estimating the price of a real estate, prediction is performed by selecting and using appropriate techniques based on the learned performance of single predictors considering the features of the real estate. Thus, it is not necessary to search for a model with the best predictive power because the combined model assigns higher weights to the most suitable ML-based models or makes them recommended to be used. Even if a model that has poor performance and uses specific data were to be incorporated into the combined model, it would be given either low weights or no recommendation for use. Hence, it will have a negligible effect on the model's predictive performance. We expect that this approach will help practitioners of mass appraisals to utilize various ML models with less apprehension. Third, the accuracy could be further increased by synthesizing the two combination methods, namely, the weighted average method and the error pattern re-learning method. The MAPE and R^2 values of the model were 4.3142% and 97.84%, respectively, which were superior to those of the other models.

The remainder of this paper is organized as follows. In Section 1, we review relevant previous literature. Previous studies on mass appraisal models and related techniques are presented. In Section 2, the techniques and data analysis process used in this study are introduced. Individual ML techniques such as SVR, RF, XGBoost, LightGBM and CatBoost used in this study and methods of combining predictive models are presented. Our data set and basic statistics are described in Section 3. Section 4 provides the results. Conclusions are summarized in the last section.

1. Literature review

In this section, literature related to this study is presented. Especially, we focus on the literature related to hedonic models and machine learning techniques frequently used in mass appraisal models. Note that there are various methods for estimating the market price of real estate. Pagourtzi et al. (2003) reviews various methods including comparable method, investment method, profit method, residual method, time series analysis, etc. Although these methods are proven methods that have been studied for a long time, some of them are not suitable for mass appraisal, either because of the need for human effort (qualitative judgement) or the characteristics of the methodology. For example, in the case of the comparable method, the characteristics and selling prices of recently sold properties in a similar area are investigated in order to evaluate the value of a certain property (subject property). The expert evaluates the subject property by adjusting the price of the properties sold in consideration of the difference between the characteristics of the properties sold and the characteristics of the subject property. Since the method is based on expert judgment, it requires human effort and is not suitable for mass appraisal. In the case of time series analysis, it has been successfully used in estimating the price of land in some studies (Hannonen, 2005), but other studies mentioned that it was difficult to construct a model that considers various attributes that affect house prices including location, property attributes and environmental amenities. In Raymond (1997) and Adamczyk and Bieda (2015), time series analysis was employed for real estate appraisal. In Pagourtzi et al. (2003), it was mentioned that "it is in any case difficult to identify the most appropriate proxy for the price index in the real estate market, since this heterogeneous sector includes different types and classes of building …" about the study of Raymond (1997). Also, in Adamczyk and Bieda (2015), it was stated that "Estimation of the market value by this method (time series analysis) is a big challenge. The assumptions adopted in this work (the scaling of the attributes …) … in reality, the valuation by forecasting methods using time series would be very laborious". In the market, real estate with various characteristics is traded at non-uniform time intervals, so it is difficult to scale it when applying time series analysis. For a study on various valuation models, please refer to Pagourtzi et al. (2003), Gabrielli and French (2021), Binoy et al. (2022).

1.1. Hedonic pricing models

The hedonic pricing model based on multiple regression analysis is the most widely used model for estimating the price of real estate in the recent past. The hedonic model originated from a study by Lancaster (1966) and its theoretical foundation was developed in Rosen (1974). In the model, real estate is assumed to be a heterogeneous commodity containing a bundle of characteristics that provide utilities. Thus, when a customer purchases a property, the customer gains a package of the characteristics in it, and these characteristics can be converted into utilities. From this point of view, various characteristics can affect the price of real estate, and several studies have been conducted to explore the factors that affect the price (Chau & Chin, 2003). The characteristics that are commonly included in the model are those related to the structural characteristics of the real estate, such as the number of rooms and toilets. In Malpezzi (2003), to estimate the price of a house, the type and number of rooms, floor area, availability and type of heating and cooling system, the age of the property were considered. Fletcher et al. (2000) and Li and Brown (1980) found that the number of rooms and floor area of a house had a positive effect on its price. In Kain and Quigley (1970), it was found that there was a negative relationship between the age of the house and the price. In addition, various studies have used the hedonic model to analyse the effects of accessibility to public transportation systems or CBD (Dubin & Sung, 1990; Hong et al., 2020; Kryvobokov & Wilhelmsson, 2007), socio-economic variables and local government service (Huh & Kwak, 1997) on prices. Sims et al. (2008) discusses the impact of a wind farm on house prices using a hedonic pricing model. Regarding the studies on real estate appraisal using the hedonic model, please refer to Malpezzi (2003), McMillan et al. (1980), Chau and Chin (2003), Dubin and Sung (1990), and Huh and Kwak (1997).

The major strength of the hedonic models is their interpretabilty (Wang & Li, 2019). It is not difficult to understand and interpret the relationship between each attribute (characteristic) and price through the regression coefficients of the obtained model. Wang and Li (2019) mentioned that one of the advantages of the hedonic model is that it is easy to explain the results of price estimation to the public. This interpretability comes from a prespecified form of the model, and some studies have criticized the model for oversimplifying the complexity or nonlinearity of the real world (Chau & Chin, 2003; Sheppard, 1999). In Hong et al. (2020), it was stated that "the functional form of the conventional hedonic pricing model is based on the simplification of household's preferences and strict assumptions about the housing. The model depends on the assumption that the effects from each attribute are separable and constant, which implies a separable preference, perfect competition, market equilibrium, and an integrated market". Due to the simplification, in practice, the prediction accuracy of the model may decrease. In Zurada et al. (2011), it was mentioned that "failures [that] would result in untenable or imprecise coefficients caused by functional form misspecification, interaction among variables, multicollinearity, and non-linearity problems". Nonlinear models have been proposed to solve these problems (Feng et al., 2021; Yeap & Lean, 2020), and studies on estimating prices using the nonparametric machine learning algorithm have been conducted.

1.2. Non-parametric machine learning (ML) methods and combined approach

In order to solve the above-mentioned shortcomings of the hedonic model, automated valuation models based on nonparametic ML techniques have been proposed. Regarding the advantages of the technique, Hong et al. (2020) mentioned that "the main advantage of the proposed method is that it constructs the model, while exploring the complexity, without the modeler explicitly describing it". Although the ML-based valuation models have a disadvantage in that their interpretability is lower than that of linear regression-based existing models, most studies have revealed that they have higher predictive accuracy. Thus, various machine learning techniques were applied in the study to develop real estate mass appraisal models. The ML methods most often used in the literature are the neural network (NN) and random forest (RF) models. For NN models, refer to Zhou et al. (2018), McCluskey et al. (2012), McCluskey and Anand (1999), Do and Grudnitski (1992), Limsombunchai (2004), Selim (2009), Torres-Pruñonosa et al. (2021) and Deaconu et al. (2022). The RF approach was employed in Yilmazer and Kocaman (2020), Hong et al. (2020), Dimopoulos et al. (2018), Antipov and Pokryshevskaya (2012), Ho et al. (2021), Bogin and Shui (2020) and Guo et al. (2020). Kok et al. (2017) used a regression tree model to evaluate multi-family assets in California, Florida, and Texas in the United States. Various tree-based regression techniques are examined to analyze the properties in Dallas, Texas, the United States, by Pace and Hayunga (2020). The k-nearest neighbor approach was employed to value the property prices in London (Bellotti, 2017) and in Szczeci (Gnat, 2021). Support vector machine technique was adopted to forecast residential housing price in Taipei city by Chen et al. (2017) and Lee and Chen (2016). Sing et al. (2022) recently employed the boosting tree ensemble technique to predict the price of real estate in Singapore. For more information on ML-based mass appraisal real estate models, please refer to the review paper by Wang and Li (2019).

In computer science field, in order to improve the predictive performance the concept of construct a better predictor by combining several predictors has been proposed. Various methods of comprising a committee (combining predictors) have been introduced. Krogh and Vedelsby (1995) and Merz and Pazzani (1996) proposed methods for constructing a combination of neural networks to improve predictive power. The methodology was applied to the computer hardware set data in the UCI repository and the bodyfat data set provided by Carnegie Mellon University for validation (Merz & Pazzani, 1996). In Taniguchi and Tresp (1997), neural networks with different hyperparameter values were combined. Four averaging methods (simple averaging, bagging, variance-based weighting and variance-based bagging) were examined and applied to Breast Cancer data and german stock index data (DAX data). The combination of multiple predictors has been employed in various studies dealing with handwritten digit recognition problem, medical diagnosis problem, and face and voice recognition problems (Liu, 2005; Verikas et al., 2002; Kittler et al., 1998; Verikas et al., 1999), and it has been shown to have better predictive performance than individual predictors. Although the combined method is used in regression and classification problems in various fields, to our best knowledge, it has not been used in real estate appraisal problems. Thus, the objective of our study is to investigate the feature of combined predictors for the appraisal problems based on a proper volume or coverage of data. Also, the latest ML algorithms (XGBoost, LightGBM, and CatBoost algorithms) are examined as automatic valuation models.

2. Methodology

2.1. Process of the model

This section illustrates the process of constructing an improved appraisal machine by combining predictors. A common task required to build models by applying ML techniques is to divide a dataset into several parts. In the case of training a single predictor, this is simply treated by randomly dividing the entire dataset into in-samples (training set) and out-samples (test set). Then, initially, the model is built by learning the data from the in-samples (training set). After the model is constructed, the data from the test set (which was not used in the estimation of the model) will be used to evaluate the accuracy (predictive power) of the mass appraisal model. Cross-validation techniques are frequently used to reduce bias and to more accurately estimate model performance.

However, an additional set is required when we design the procedure to construct the combination of single ML algorithm-based predictors. This is because the predictive powers of each single predictor are employed in the process of combining the predictors. In other words, the single predictors are initially trained on a dataset (training set) and each model is constructed. Then, another dataset (test dataset), which is independent of the training dataset, is used to evaluate each predictor. The predictive power of each single predictor is calculated based on the test dataset and the trained models are integrated to construct a combined model based on the predictive powers. For example, by observing the predictive powers of each predictor, it is possible to determine which predictors' predictions are accurate in a case. In this case, by assigning different weight values to each model, a combined model can be created to perform prediction with a weighted average of the prediction values from the single predictors. As the data in the test set are used in the construction of the combined model, another dataset is needed to evaluate the predictive power of the combined model. To avoid confusion, the set used to train and evaluate single predictors is defined as Set 1, and the test set used to evaluate the combined model is defined as Set 2. In other words, the single predictors are trained and evaluated by applying the 5-fold cross-validation technique to the data of Set 1. Then, a combined model is constructed using the prediction results (Set 1), and the combined model is assessed using Set 2.

Therefore, we divide the plot of analysis into two parts, called first- and second-stage analyses. In the first-stage analysis, we examine the single ML algorithm models only. The single predictors are trained, and the performances of the single predictors are measured and analyzed (on the entire data set excluding Set 2). A 5-fold cross-validation technique is employed. In the second-stage analysis, we construct the combinations of the single-algorithm-based predictors on the basis of the result obtained in the first-stage analysis. To discuss the features of combined predictors, the predictive accuracy of all the single-algorithm predictors and the combined predictors are estimated with another test set (Set 2).

Figure 1 shows a flowchart illustration of our model. The samples are first randomly divided into in-samples and out-samples at a ratio of 8:2. The in-samples are (Set 1) used to train and evaluate the single predictors, and the out-sample (Set 2) is used to evaluate the combined model. The single predictors are trained and evaluated on the in-samples (Set 1) based on a 5-fold cross-validation technique in the first stage analysis. In this step, the hyperparameters of each algorithm are optimized. In the second-stage analysis, the combination of the predictors is constructed based on the predictive performance on Set 1 and evaluated using Set 2. There are three models of combining the predictors suggested and discussed: naïve averaging, weighted averaging with parameterization, and ML-based voting. We evaluate and analyze the performances of all predictors (including the single-algorithm predictors and the combined model) on Set 2.

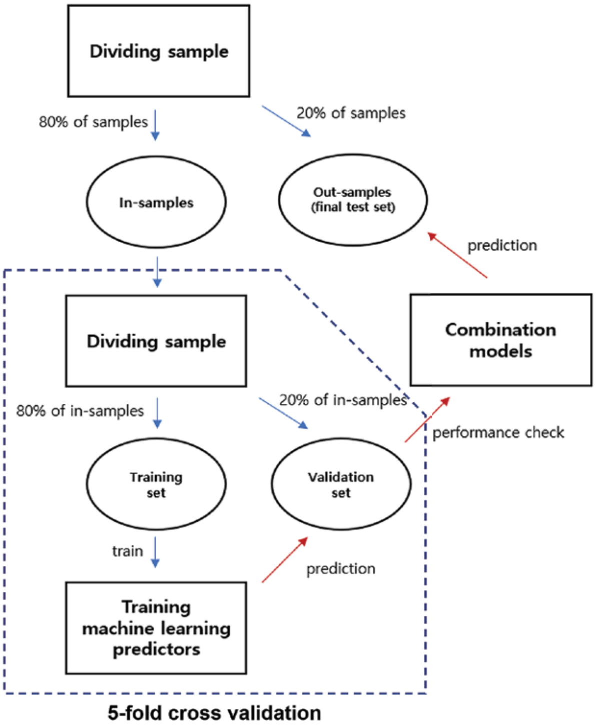


Figure 1. Flowchart illustrating our model

The features employed in our study are determined based on the feature selection methods of decision tree-based models. Feature selection methods of tree-based machine learning models fall under the category of embedded methods. They are implemented by algorithms that have their own built-in feature selection methods. The importance of each feature can be calculated, which means the ability of that feature to increase the purity of leaves in the tree structure. Hong et al. (2020) employed the random forest technique for a mass appraisal model of real estate and determined the list of variables to improve predictive power using recursive feature elimination. The features used in this study were determined using the same procedure (recursive feature elimination) used in the study of Hong et al. (2020).

2.2. The combination models

2.2.1. Naïve averaging

In naïve averaging, the arithmetic mean of the value predicted by each algorithm is used to predict the outcome variable of a sample. This might be the simplest combination method. The naïve averaging predictor for sample *j*, AVG_j, can be expressed as follows:

$$AVG_j = \frac{1}{n}\sum_{i=1}^{n} z_{i,j}, \tag{1}$$

where $z_{i,j}$ is the predicted value of the outcome variable of sample *j* calculated using the *i*-th (single) predictor, and *n* is the number of single predictors.

The naïve averaging model assumes that the same weights are assigned to all the combined predictors. This implies that the naïve averaging predictor does not use any information about the predictive performance of the combined model. The pros and cons of this method come from its simplicity. The advantage is that this method can be immediately applied to the combined predictors without any additional testing. This feature can be particularly virtuous when the volume or coverage of the data is limited. One weak point is that, while the predictive performance of the combined predictors can vary, the naïve averaging eliminates them. Nevertheless, the predictive performance of this combined predictor is often higher than that of each predictor obtained from a single algorithm is, particularly if the residuals due to each algorithm's calculation method are idiosyncratic. This is because when the error-generating patterns of the models are independent (i.e., not similarly biased), their deviation could be offset using simple averaging.

2.2.2. Weighted averaging with parameterization

The weighted averaging model allocates different weights to each predictor through the process of averaging. When the predictors exhibit different levels of performance, we may conjecture that some of the algorithms are better (or worse) fitted to capture the complexity embodied in the analysis objective. Therefore, in the averaging of predictors, weighting based on their relative performance could be helpful in improving the accuracy of the combined predictor.

There are many ways to calculate the weights. In this paper, we suggest that the parameterization of the minimum performance gap is an efficient method. One issue about setting a minimum performance model may arise, especially when some algorithms are not suitable for the prediction. In that case, as the inferiority of a model might pull down the predictive power of the combined model, we need to eliminate the predictions from those algorithms.

One intuitive way to deal with this issue is to define a threshold for the allowable performance gap. The performance gap can be defined as the gap between the MAPE of a predictor and the MAPE of the most performative one. If the performance gap is greater than the threshold, we can discard the predictor in the combination process by setting zero weight on the predictor. Moreover, for the predictors within the threshold gap, their performance gaps can be used as the parameters for smoothing the weights. If we standardize the weight on the most performative predictor as 1, the weight for the *i*-th predictor, ϕ_i, can be defined as follows:

$$\phi_i = \begin{cases} \dfrac{\theta - \left(MAPE_i - \overline{MAPE}\right)}{\theta}, \; if\; MAPE_i - \overline{MAPE} < \theta, \\ 0, otherwise \end{cases} \tag{2}$$

where θ is the parameter for the cut-off threshold, which is a positive scalar, and $MAPE_i$ and $\overline{MAPE}$ are the MAPE of *i*-th single-algorithm predictor and the lowest MAPE among the predictors, respectively.

Then, the weighted averaging predictor can be obtained as follows:

$$WVG_j = \sum_{i=1}^{n} \frac{\phi_i}{\phi} z_{i,j}, \tag{3}$$

where $\phi = \sum_{k=1}^{n} \phi_k$. Note that $z_{i,j}$ was defined as the predicted value of the outcome variable of sample *j* by using the *i*-th (single) predictor.

The parameter θ represents the degree of dependence on the most performative predictor. When θ decreases (increases), it narrows (widens) the boundary allowing predictors to be used in the combination and provides more (less) weight on the most performative predictor. The weighted averaging predictor falls between the naïve averaging predictor and the most performative single-algorithm predictor because $\lim_{\theta \to \infty} WVG_j \to AVG_j$ and $\lim_{\theta \to 0} WVG_j \to z_{k,j}$. Figure 2 shows how the size of the weight is smoothed with the relative performance, by providing an example.

In this example, the horizontal axis indicates the performance gap of the model, $\left(MAPE_s - \overline{MAPE}\right)$. The vertical axis indicates the weight of the model, ϕ_s. The weight is illustrated assuming the cut-off threshold, θ, is 3. The main

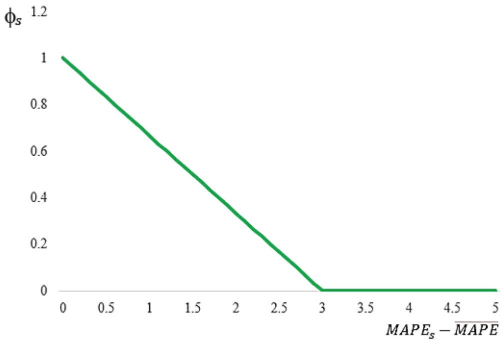


Figure 2. An example of the relationship between the performance gap and model weighting (θ = 3)

advantage of this parameterization method is its simplicity and intuitiveness. The modeler is only required to set a single parameter, θ, and then, the parameter determines not only which algorithms are used but also the relative value of the weights. Since unsuitable algorithms would eventually be eliminated from the combined predictor using this approach, employing various algorithms during the first-stage prediction is less harmful if the threshold is well-optimized. So, this combination model can be expected to reduce modeling cost in the practice of mass appraisal. This advantage is remarkable because it is exceedingly difficult for a model designer to know in advance which algorithm is suitable for analysis due to the nature of ML method.

2.2.3. Machine learning (ML)-based voting and averaging

In the averaging approaches (the naïve averaging and weighted averaging), it can be expected that the noises from a ML predictor can be diluted by noise from other predictors. Another approach could be employing the ML algorithms to learn the relative performance of the predictors and making predictions by voting. The rationale underlying this approach is that a certain predictor might capture the complexity of the samples in terms of specific types or situations better than the other predictors might. We expect that a more powerful combined model can be obtained if the pattern of the predictive power can be analyzed using the ML algorithms.

After the single-algorithm predictors are trained on the training set, their predictive performance is compared in Set 1. Then, we select the predictor with the highest predictive power for each observation and replace it with the existing outcome variable, price. In other words, a new dataset consisting of the features of the existing data and the predictor with the highest predictive power (outcome variable) is created. The most performative predictor might be different for each observation as each one has distinctive characteristics. On the basis of the data, we build ML classifiers predicting which ML algorithm performs better with the given features of the sample. Thus, the ML algorithm is used to predict the predictor with the highest predictive power based on the features of observation.

To determine which algorithms are best using Set 1, all algorithms used to build the single predictors can be used. Therefore, we use support vector machine (SVM), RF, XGBoost, LightGBM, and CatBoost algorithms to calculate a predictor with an elevated level of predictive power for each observation. Each of the trained classifiers recommends one algorithm for each sample. Note that there are five classifiers recommending which algorithm would be the best. To make predictions, the recommendation results from the five classifiers (it may be different) should be integrated. Here, we combine the results through soft voting (i.e., averaging the predicted values). For example, if two classifiers recommend the RF algorithm for a certain sample, and three classifiers recommend the XGBoost algorithm, the predicted value is (2/5)*(predicted value obtained from RF algorithm)+(3/5)*(predicted value obtained from XGBoost).

2.3. Machine learning (ML) algorithms

In this section, the ML algorithms used as single predictors are briefly introduced.

2.3.1. Support vector regression (SVR)

The SVM model was first introduced by Boser et al. (1992) to address binary classification issues, which can be viewed as the task of separating two classes in the feature space, and the main goal of SVM is to determine a linear classifier, or a so-called "hyperplane", that separates the data into classes. The classification problems is formulated as a convex optimization problem with the goal of establishing the hyperplane to maximize the distance from the plane to the nearest data point in each class. The concept of the hyperplane in two dimensions is depicted in Figure 3.

The SVM concept has been extended toward solving various regression problems (Smola & Schölkopf, 2004). Whereas SVM is used to classify data into binary classes, SVR is a generalization of SVM that is used to predict real values. In SVR, the input value is first mapped in a

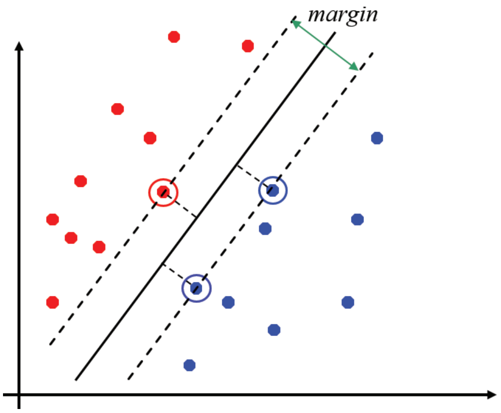


Figure 3. The hyperplane of the support vector machine (SVM) model

high-dimensional feature space, and then, the function associated with the mapped value is found (Lu et al., 2009). A convex, e-insensitive loss function called "e-tube" is employed to formulate the optimization problem in order to determine the tube that best approximates the function that balances the complexity of the model and the prediction error. As in SVM, the support vectors in SVR are crucial factors that affect the shape of the tube. SVR has been applied to real estate valuation problems in previous studies (Chen et al., 2017; Lee & Chen, 2016; Han & Clemmensen, 2014; Lin & Chen, 2011).

2.3.2. Decision tree (DT) model

The RF, XGBoost, LightGBM, and CatBoost algorithms used in this paper are based on the decision tree (DT) model. Accordingly, we now introduce the decision tree (DT) model and briefly discuss how each technique is derived from it. DT models are predictive models based on the form of a tree structure that can be applied to both classification and prediction problems. A tree structure consists of decision nodes and leaf nodes, and each decision node has two or more branches based on a feature with a threshold. For example, we consider a DT model with several nodes in Figure 4.

An arrow represents a branch, and a diamond and a square represent a decision node and a leaf node, respectively. In the classification problem, when a new observation arrives, it is classified in a node based on a feature (x_i) with a threshold (t_i). In Figure 4, the value of feature x_1 of a new observation is greater than t_1, the tree takes the observation to the left branch (otherwise, it goes to the right branch). Then, the value of feature x_2 of the observation is compared with threshold t_2. The classification of an observation is finished by following the branches from the root node to a leaf node. Each leaf node represents a class. For a regression problem, the average values are applied to the divided subspaces defined by the leaf nodes after following the branches, as in classification. Thus, a DT model is a predictive model that works by recursively partitioning the feature space. The advantages and application of DT models have been presented in Safavian and Landgrebe (1991), Myles et al. (2004) and Song and Lu (2015).

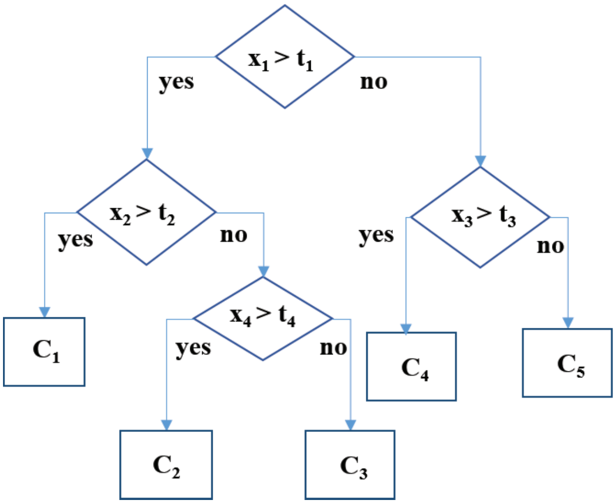


Figure 4. An illustration of the decision model

2.3.3. Random forest (RF) regression

An RF regression is an ensemble learning method for regression based on multiple DTs that provides predictions by averaging the predictions from individual trees. This technique was first proposed by Ho (1995), who found that a combination of tree predictors splitting with hyperplanes yields better predictive power as they expand without suffering from overtraining. Since the concept for the ensemble method was proposed, various extensions have been developed (Amit & Geman, 1997; Breiman, 2001).

In RF, several DTs are constructed using a different bootstrapped sample of the data during the training time, and the trees in the ensemble grow independently. In a standard tree model, each node is split using the best split among all of the features (variables). However, an RF model performs its split process by randomly using a subset of features. An unpruned regression tree is grown for each bootstrapped sample. After a large number of trees are generated, predictions are averaged over the different trees. Liaw and Wiener (2002) stated that "this somewhat counterintuitive strategy turns out to perform very well compared to many other classifiers, including discriminant analysis, support vector machines and neural networks, and is robust against overfitting". Due to its strength, this method has been applied to various real estate valuation problems (Antipov & Pokryshevskaya, 2012; Čeh et al., 2018; Hong et al., 2020; Levantesi & Piscopo, 2020).

2.3.4. Gradient-boosting technique

The XGBoost, LightGBM, and CatBoost techniques fall under the category of so-called "gradient-boosting techniques". Boosting refers to an ensemble learning method that creates a more accurate learner by combining simple and weak learners (i.e., DTs) in a direction that minimizes error. While bagging techniques, such as RF, train multiple trees independently in parallel and average the results from the trees to make predictions, the boosting technique builds the trees sequentially such that each subsequent tree aims to reduce the error of the previous tree. In other words, we attempt to minimize the errors made by previous tree models in succeeding trees by adding weights to the model. This technique is based on the intuition that the next best possible model will reduce the overall prediction error when combined with previous models. The trees in an ensemble can only be interdependent as each successive tree learns from its predecessors.

The gradient-boosting algorithm constructs new base-learners to be maximally correlated with the negative gradient of the loss function, which is associated with the whole ensemble. The loss function defined in gradient boosting is a measure that indicates how good the coefficients of the model are at fitting the underlying data. The loss function can be arbitrarily applied, but in real estate valuation, it would be based on the difference between the true and predicted real estate prices. The gradient descend procedure is employed to minimize the loss when trees are added. For a more detailed description of the gradient-boosting algorithm, please refer to Friedman (2001, 2002).

2.3.5. XGBoost algorithm

XGBoost (extreme gradient boosting) is a gradient-boosting technique (ensemble) with enhanced performance and speed used in tree-based (sequential DTs) learning algorithms. It was developed by Tianqi Chen and initially maintained by the Distributed (Deep) Machine Learning Community group (Chen & Guestrin, 2016).

As XGBoost is described as a scalable ML system in Chen and Guestrin (2016), the advantage of the algorithm is its scalability. The model runs more than 10 times faster than existing popular solutions do on a single machine and scales to billions of examples in distributed or memory-limited settings. Similar to other gradient-boosting techniques, XGBoost uses the gradients of different cuts to select the next cut. However, XGBoost also computes second partial derivatives of the loss function just as Newton's method does. This provides information about the direction of the gradients to minimize the loss function. Computing the derivative imposes a slight cost, but it helps to estimate the cut that should be used. In addition, the algorithm employs advanced regularization to improve model generalization.

2.3.6. LightGBM algorithm

The LightGBM technique was developed to overcome the disadvantage that gradient-boosting decision tree (GBDT) models operate inefficiently when the feature dimension is high and the quantity of data is large (Ke et al., 2017). The existing gradient-boosting tree models investigate all data points to estimate the information gain that will yield the best split point, which leads to an increase in computational complexity. To deal with this problem, two techniques are employed in the LightGBM technique: gradient-based one-sided sampling (GOSS) and exclusive feature bundling (EFB). Thus, the GBDT algorithm with GOSS and EFB is defined as LightGBM.

The goal of GOSS is to reduce the number of data points while maintaining the accuracy of the learned DTs. The basic idea of GOSS is to use the gradient for each data instance for data sampling. After the algorithm calculates the absolute value of the gradient for each data point, an instance with a gradient of a large absolute value is preserved. The algorithm performs random sampling on the instances with a small gradient. Ke et al. (2017) proved that GOSS can efficiently reduce the number of samples while not losing any theoretical training accuracy. The EFB method deals with the sparsity of the feature space. In a sparse feature space, mutually exclusive features can be bundled, and it can reduce the training duration of the model without sacrificing accuracy.

2.3.7. CatBoost algorithm

Dorogush et al. (2018) and Prokhorenkova et al. (2017) introduce a new gradient-boosting technique called CatBoost, with which they propose to manage heterogeneous datasets that contain features with different data types. Its development was motivated by a statistical issue called "prediction shift" that arose in all existing gradient-boosting techniques. In the implementation of the gradient-boosting technique, the predictive model relies on the targets of the training samples after boosting. Prokhorenkova et al. (2017) stated that this boosting process leads to a shift in the distribution of the training dataset from the distribution of the test dataset. A similar problem occurs in preprocessing categorical features. In this paper, these problems are called "target leakage" and "prediction shift".

The CatBoost algorithm employs several methods to deal with the aforementioned problems. First, the algorithm employs a new encoding technique called "ordered target encoding". In many existing gradient-boosting algorithms, one-hot encoding is used at the preprocessing stage. In one-hot encoding, each categorical value is converted into a new categorical column, and a binary value of 1 or 0 is assigned to each column. This causes an increase in the number of features. The target encoding in CatBoost is a more scalable method as the encoded quantity is an estimation of the expected target value for each category of the feature. Next, "ordered target statistics" are used to solve the target leakage problem. They arrange observations in a training dataset according to an artificial timeline defined by a permutation of the training dataset. For an observation from a training set, target statistics are computed using its own "history". The ordered boosting technique is integrated with ordered target statistics. In practice, several permutations of the training set are then generated, and a randomly chosen permutation is used to compute the target statistics at each step of the gradient-boosting algorithm.

3. Dataset and descriptive statistics

Seoul is the capital and largest city of South Korea. With a population of 9.7 million and an area of 605.21 square kilometers (km^2), Seoul's real estate market was ranked second in the world in terms of the price per square meter to buy an apartment in the city center (downtown) in 2020 (Chris, 2020).

Seoul is composed of 25 administrative divisions called "*gu*". Each *gu* is different in size (between 10 and 47 km^2), and the population of each ranges between 140,000 and 630,000 residents (Figure 5). Each *gu* is subdivided into "*dongs*". In total, Seoul consists of 423 administrative *dongs*.

We collected our dataset on the apartments sold in 2018 from South Korea's Ministry of Land, Infrastructure, and Transport (MOLIT). After excluding data with missing values, analysis was performed on the remaining 56,897 data points. The dataset covers about 76% of

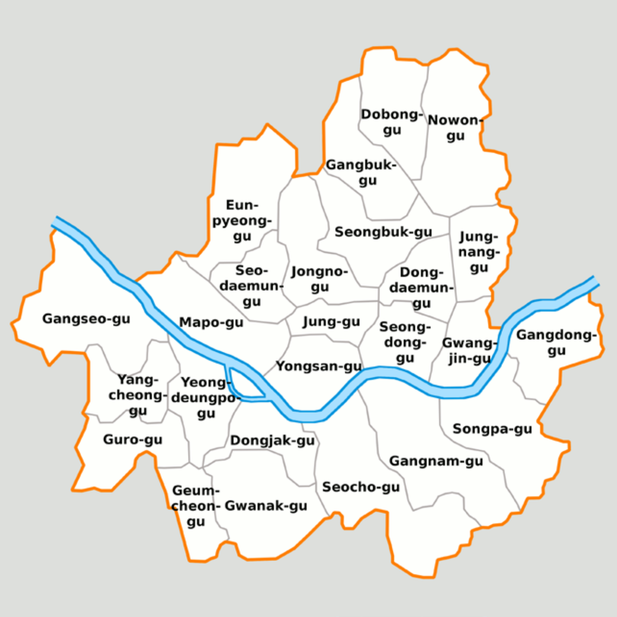


Figure 5. A map of Seoul, Korea (source: Wikimedia Commons, 2005)

all apartment transactions that occurred in Seoul during the period. Our models involve regressing the observed apartment price against apartment features that are hypothesized to contribute to the price. The target variable and features used in our analysis are provided in Table 1.

The property attributes consist of intrinsic characteristics of the property, including the number of bedrooms and bathrooms, the floor area, and the age of the property, and they are commonly used to determine the property's value. In previous studies, the number of bedrooms and the floor area of a property were found to be positively related to its price (Fletcher et al., 2000; Garrod & Willis, 1992; Li & Brown, 1980). Kain and Quigley (1970) observed that the elapsed year (i.e., the age) of a property can negatively affect its price. In this study, we employ the number of elapsed years (transaction year–construction year), area, floor level, number of bedrooms and bathrooms, and heating system.

Apartment attributes are characteristics common to all properties of an apartment. For example, two apartments may have different areas and different numbers of rooms but still have the same floor area ratio value. Thus, the characteristics of an apartment having the same value are defined as an apartment attribute. For these attributes,

Table 1. The variables used in our models

Category	Variable	Units
Target variable	Traded apartment prices	Korean won
Property attributes	Years elapsed	years
	Size (in area)	square meters (m^2)
	Floor	floor level
	Number of bedrooms	number
	Number of bathrooms	number
	Heating system	central/individual/local district (categorical)
Apartment attributes	Hallway type	stairs/corridors/combined (categorical)
	Number of households in complex	number
	Number of apartment buildings in complex	number
	Average number of parking spots per household	number of parking spots/number of units
	Floor area ratio	ratio
	Building coverage ratio	ratio
	The top floor of an apartment	floor level
	The lowest floor of an apartment	floor level
Neighborhood attributes	Dong	423 categorical variables
	Latitude	latitude of a property
	Longitude	longitude of a property
	Distance to subway station	meters
	Distance to national park	meters
	Distance to elementary school	meters
	Distance to middle school	meters
	Distance to high school	meters
	Distance to university	meters
	Distance to museum	meters
	Distance to district office	meters

we consider hallway type, number of households in the apartment complex, number of buildings in the apartment complex, parking lot availability, floor area ratio, building coverage ratio, and the highest/lowest floor of the building. To evaluate the quality of an apartment complex's parking lot, we calculate the average number of parking spots per apartment (total number of parking spots in an parking lot divided by the number of households in the complex). The floor area ratio (FAR) and building coverage ratio (BCR) are the ratio of the gross floor area and the building area divided by the land area, respectively.

Neighborhood attributes are the characteristics related to a property's geographic location. Yu (2007) posited that the characteristics include all of the externalities associated with a house's geographic location, such as accessibility, proximity to externalities, environmental amenities, and land use information. To account for the geographic location of the properties in our models, we employ an apartment's administrative division (*dong*) and its latitude and longitude for ML models. For OLS-based linear regression, accessibility (distance) to nearby facilities is also used. The facilities considered are subway stations, national parks, elementary schools, middle schools, high schools, universities, museums, and district offices. Except for information about the administrative division to which the apartment belongs (which can be found in the data provided by MOLIT), all other information was calculated as the distance from each property to the center point of the facility using the MAP Open Application Programming Interface (API).

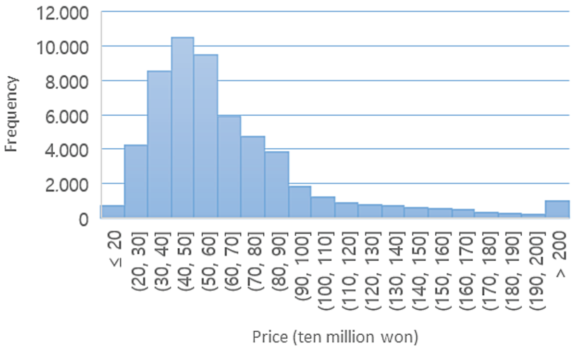


Figure 6. A histogram of the distribution of prices

The descriptive statistics are presented in Table 2. A histogram of the traded apartment prices (target variable) is presented in Figure 6.

4. Quantitative results

4.1. First-stage analysis: comparison of single-algorithm predictors

In this step, the OLS-based predictor and the single-algorithm predictors (i.e., SVR, RF, XGBoost, LightGBM, and CatBoost) are trained on the same train set, and their performances are compared. As previously mentioned, 80% of the entire dataset (in-samples, Set 1) will be used to train and evaluate single predictors, and 20% (out-samples, Set 2) will be used to evaluate the combined model. The single predictors are trained and evaluated on

Table 2. Descriptive statistics

Variable	Mean	Median	Standard deviation	Min.	Max.
Number of years elapsed	16.824	17.000	7.774	0.000	47.000
Area	81.381	84.670	28.043	12.100	273.310
Floor level	9.368	8.000	6.280	1.000	63.000
Number of rooms	3.033	3.000	0.628	1.000	7.000
Number of bathrooms	1.685	2.000	0.484	1.000	5.000
Number of households in complex	1,060.615	700.000	1,096.367	6.000	6,864.000
Number of apartment buildings in the complex	12.146	8.000	12.957	1.000	122.000
Average number of parking spots per household	1.137	1.130	0.440	0.080	11.950
Floor area ratio	271.444	253.000	95.427	72.000	1096.000
Building coverage ratio	22.264	21.000	7.203	2.000	49.000
Latitude	37.558	37.553	0.057	37.439	37.688
Longitude	126.989	127.013	0.091	126.807	127.181
Distance to a subway station	832.997	639.495	663.299	3.806	5,112.545
Distance to a national park	1,030.008	970.290	520.581	63.869	3,268.238
Distance to an elementary school	334.372	319.441	168.536	10.597	1,810.038
Distance to a middle school	469.687	435.007	250.278	2.587	2,130.155
Distance to a high school	577.090	507.220	336.620	24.616	2,771.288
Distance to a university	1,843.071	1,524.187	1,206.868	55.467	7,111.538
Distance to a museum	1,897.700	1,699.641	1,084.829	45.630	6,839.077
Distance to a district office	1,968.059	1,896.326	994.141	16.821	6,521.695

the in-samples (Set 1) based on a 5-fold cross-validation technique in the first stage analysis.

To measure the accuracy of the predictors, we employ three conventional measurements: the MAPE, R-squared value and COD. The MAPE is a straightforward measurement for determining the average percentage error from actual prices. Percentage deviations (for each sample) are averaged after taking the absolute value by ignoring the sign on the error. Because of its convenience and intuitiveness, this measurement is frequently used in mass appraisals. The formula is expressed as follows:

$$MAPE = \frac{100}{n}\sum_{i=1}^{n}\left|\frac{\hat{p}_i - p_i}{p_i}\right|, \tag{4}$$

where n, p_i and $\hat{p}_i$ are the sample size for the prediction, the actual price and predicted price of property i, respectively.

The R-squared value measures the proportion of the variance in the target variable (i.e., actual transaction price) for which the model accounts. It represents the portion of the observed price that is predictable by the model. The R-squared value is calculated as follows:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}\left(p_i - \hat{p}_i\right)^2}{\sum_{i=1}^{n}\left(p_i - \overline{p}_i\right)^2}, \tag{5}$$

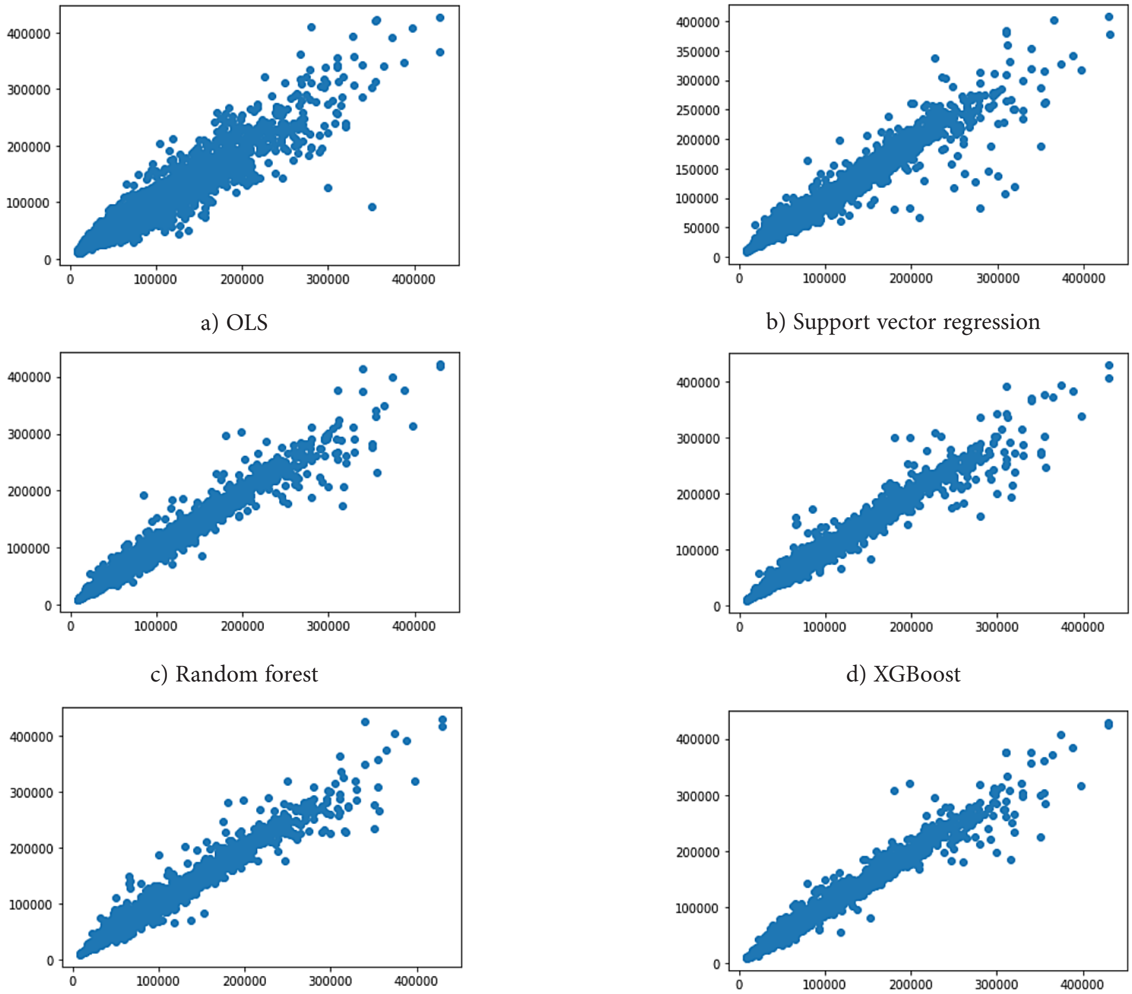
where $\overline{p}_i$ is the sample mean of the actual transaction price for property i.

The COD measures the dispersion of sales ratio, the quotient obtained by dividing the predicted price with actual transaction price, around the median sales ratio (Hong et al., 2020). The COD is a measure of uniformity and relates to the consistency of assessment levels within a group of properties. It can be expressed as:

$$COD = \frac{100}{R_m}\left(\frac{\sum_{i=1}^{n}\left|R_i - R_m\right|}{n}\right), \tag{6}$$

where R_i is the ratio between the predicted value and actual value for the apartment price i; R_m is the median ratio.

Let us observe the result for the single-algorithm predictors. Figure 7 shows the scatter plot between the actual transaction prices and the predicted values. Table 3 presents the accuracy measurements obtained from the



a) OLS

b) Support vector regression

c) Random forest

d) XGBoost

e) LightGBM

f) CatBoost

Figure 7. Scatter plots of the actual contract prices (depicted on the horizontal axis) and the predicted values (shown on the vertical axis) of the single predictors

Table 3. Accuracy measurements: first-stage analysis

		OLS	SVR	RF	XGBoost	LightGBM	CatBoost
1st	MAPE	11.871	6.629	5.069	4.855	4.959	4.481
	R^2	0.901	0.957	0.974	0.975	0.977	0.98
	COD	11.889	6.637	5.06	4.86	4.957	4.48
2nd	MAPE	11.862	6.752	5.023	4.831	4.975	4.46
	R^2	0.898	0.941	0.974	0.971	0.972	0.978
	COD	11.886	6.735	5.007	4.833	4.972	4.45
3rd	MAPE	11.846	6.901	5.087	4.862	5.09	4.511
	R^2	0.897	0.953	0.973	0.976	0.974	0.979
	COD	11.868	6.903	5.073	4.864	5.08	4.504
4th	MAPE	11.664	6.63	4.972	4.769	4.932	4.401
	R^2	0.902	0.927	0.967	0.968	0.97	0.973
	COD	11.721	6.651	4.972	4.775	4.935	4.406
5th	MAPE	12.078	6.769	5.096	4.939	5.052	4.57
	R^2	0.892	0.956	0.972	0.974	0.974	0.979
	COD	12.095	6.77	5.083	4.942	5.047	4.568
Average	MAPE	11.864	6.736	5.049	4.851	5.0016	4.485
	R^2	0.898	0.947	0.972	0.973	0.973	0.978
	COD	11.892	6.739	5.039	4.855	4.9982	4.482

single-algorithm predictors. In summary, the result tells the following. First, the ML-based predictors are more performative than the OLS-based predictor is. The values of MAPE for ML-based predictors are around 4.4 to 6.6, while the corresponding value of the OLS predictor is 11.68, more than double the ML-based predictors' values. The R-squared of ML-based predictions are also noticeably higher than the R-squared of the OLS. The R-squared of SVR, RF, XGBoost, LightGBM, and CatBoost are 0.945, 0.969, 0.971, 0.969, and 0.976, respectively, which implies that 95% to 97% of the variance of the dependent variable has been accounted for while the remaining 5% to 3% of the variability has not. Details about the hedonic model and the hyperparameters of the ML models are presented in Appendix 1 and Appendix 2, respectively.

Second, the predictions obtained from ML predictors (particularly, the boosted tree algorithms, XGBoost, LightGBM, and CatBoost) are accurate enough to be directly applied to a wide range of practical mass appraisals. The value of the MAPE for those predictors is around 5, which indicates that the percent deviation of the predictions from the actual contract price is approximately 5%, on average. If we consider that, in the transaction price, there is noise that is impossible to capture (such as a contractor's preference or imperfect information), the absolute level of accuracy obtained from the ML predictors may be close to a professional appraiser's evaluation. The reason that our model shows accurate predictive power is related to the fact that the data only include apartment transactions. Hong et al. (2020) stated that "the structural characteristics of the apartments can be sufficiently represented by a number of common and measurable features … Housing in different residential areas or in detached dwellings are usually more various in their amenities, interior decorations, and features and consequently are difficult to codify or consolidate in a dataset, which eventually undermines the accuracy of predictors". Nevertheless, this study shows that ML-based automated valuation models may successfully estimate the market price. Cannon and Cole (2011) stated that the MAPE of human appraisal is about 12%.

An interesting point is that the DT-based algorithms (i.e., RF, XGBoost, LightGBM, and CatBoost) are more performative than both OLS and SVR predictors. The MAPE values of the RF, XGBoost, LightGBM, and CatBoost algorithms are 4.96, 4.77, 5.05, and 4.43, respectively, while the corresponding SVR value is 6.67. We also found that the R-squared value of SVR is significantly less (0.945) than that of each of the tree-based predictors. This implies the DT algorithm has an advantage in capturing the complexity of the housing market.

The existence of housing submarkets can be related to the advantages of tree-based algorithms. The housing market is not a single, integrated market but is broken down into submarkets that can be distinguished according to various classifiers, such as property type, size, quality, and location. For example, for small houses, if the number of bathrooms is greater than one, the relationship between the property value and the number of bathrooms might be insignificant. Conversely, for large houses, the number of bathrooms may have more significance as the families that live in such houses are likely to be relatively larger.

The structure of the housing submarket is complex because of its multiple layers. For example, submarkets exist not only for high-priced and low-priced houses but also for different property types and locations. Therefore, to improve the predictive accuracy of a mass appraisal model, the hierarchical structure of housing submarkets should be explored.

The DT algorithm can be used to explore the hierarchical structure of conditions that classifies properties in the housing market with similar characteristics (Fan et al., 2006). This algorithm forms a tree of conditions classifying the sample in the order of variables with a high level of information. The advantage is that this model can explore different valuation structures for each branch. For example, in the upper condition node, if housing size is divided above and below a certain level, then the valuation structures for large and small houses (distinguished by the upper condition) can be established separately. This means that the ML algorithm can capture heterogeneity in housing submarkets and consequently prevent the loss of explanatory power that might be attributable to the existence of submarkets. In other words, the ML algorithm can serve as a data-driven model to determine the hierarchy of the housing submarkets.

4.2. The features of prediction errors

We investigate the feature of prediction errors obtained in the first-stage analysis. Table 4 presents the pairwise correlation coefficients between different models. The correlation coefficients extend from 0.55 to 0.84, which implies that there are considerable positive correlations among the errors.

This reminds us that errors can come not only from the incompleteness of the model but also from the stochasticity of the property price itself. The positive correlations imply that when a certain predictor makes an error, other predictions tend to deviate from the price in the same way. It might be impossible to take account of all of the complexity of the real world as models are a simplification and standardization of the real world. If an unobservable factor is affecting the transaction price, the predictions tend to deviate similarly from the observed price. This means that the prediction error cannot be perfectly eliminated even if the model is perfect.

The transaction price includes various types of noise. While some types of noise (such as fire sales and contractor's psychology or caprice) are idiosyncratic, the sizes of some types of noise (such as valuation ambiguity or information asymmetricity between seller and buyer) could be dependent on the characteristics of the property because each market participant is unique. The relationship between the error and housing characteristics would become more significant the more definitive the housing submarket is.

Table 5 shows the MAPEs of the model for the different property sizes. The prediction accuracy for small housing (<60 m^2) is higher than that for the others. Table 6 compares the MAPEs for housing with different ages. It is shown that the predictions for new housings (the elapsed years of which are less than, or equal to, five) are lower. The interesting point is that the relative performance of the models can also change with property characteristics. When each segmented market has different complexity, the model's adaptiveness to each complexity can vary. For example, in Table 5, the MAPE of SVR for the small housings has decreased from 6.7133 to 6.6619, while the MAPEs of the other models increased. Table 6 shows that the most performative model can vary with property

Table 4. Pairwise correlation coefficients for percentage error

	SVR	RF	XGBoost	LightGBM	CatBoost
SVR	–	0.586	0.558	0.601	0.617
RF	–	–	0.849	0.747	0.740
XGBoost	–	–	–	0.730	0.781
LightGBM	–	–	–	–	0.717
CATBoost	–	–	–	–	–

Note: Numbers were rounded off to three decimal places.

Table 5. Mean absolute percentage error (MAPE) values calculated according to property size

	SVR	RF	XGBoost	LightGBM	CatBoost
<60 m^2	6.7133	4.6894	4.4200	4.9134	4.2958
≥60 m^2	6.6619	5.1161	4.9667	5.1305	4.5100

Table 6. MAPE values calculated according to the number of years elapsed

	SVR	RF	XGBoost	LightGBM	CatBoost
<5 years	6.8373	5.1131	5.0173	5.433	5.0696
≥5 years	6.6692	4.957	4.7592	5.029	4.3922

characteristics. On average, the most performative single-algorithm predictor was CatBoost; however, XGBoost has the lowest MAPE for newer housing. Therefore, it would be appropriate to use XGBoost when estimating the price of real estate that is less than 5 years old, and CatBoost algorithm otherwise. This implies that the appropriate algorithm may differ depending on the property characteristics. Note that one of the combined models is a method that recommends the most suitable algorithm after learning the errors of single predictors according to features with ML techniques (ML-based voting). As such, the performance of the prediction model can be improved by learning the errors of a single predictor according to the feature and finding the predictor with the best predictive power through ML-based voting.

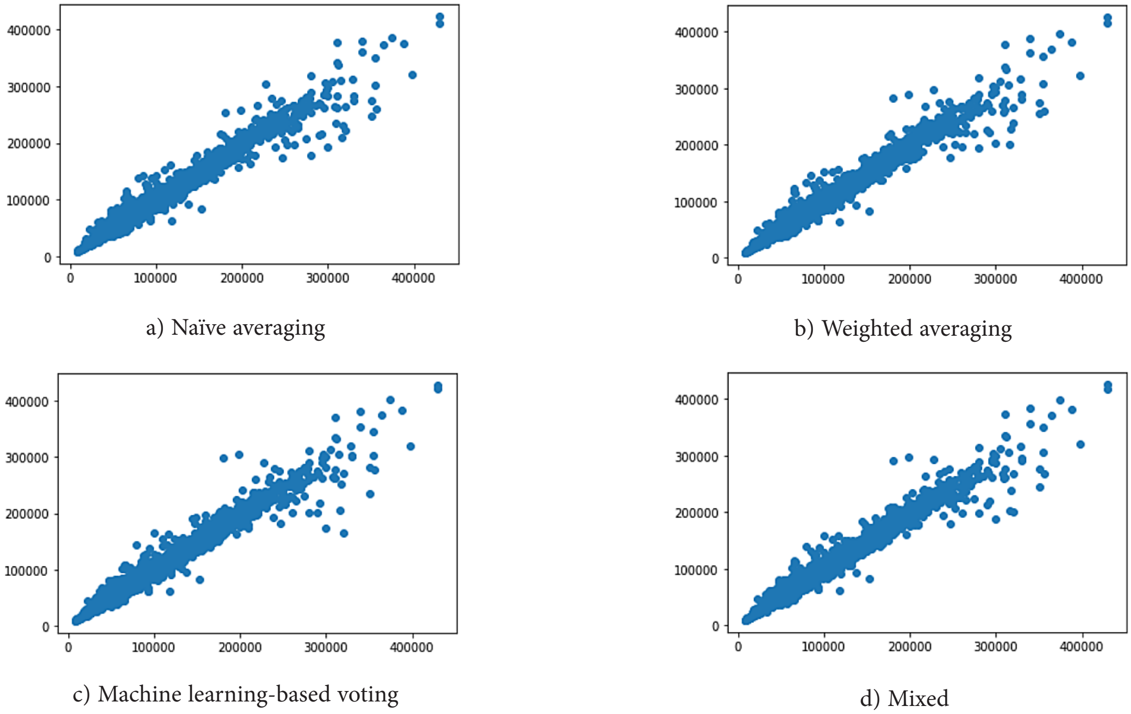
4.3. Second-stage analysis

To examine whether the combination of predictors can provide improvement, we consider the three aforementioned approaches. The first approach is the naïve averaging, which implies that the difference in a predictor's relative performance is eliminated. In the approach, the prediction value is calculated as the average of the prediction values obtained from each single predictor. The second approach is the weighted average. We set a parameter for the allowable performance gap, and the weights of the predictors in averaging can be obtained based on their relative accuracy. The third approach is the ML-based voting and averaging. The ML algorithms can also be trained for which predictor is most performative with the intermediate test samples. By training so, the predictors can recommend algorithms for each sample in the test set. The predictions from recommended predictors are averaged (soft voting).

To compare the performance of combined predictors and single predictors, evaluation must be performed on the same set. Note that in the previous evaluation procedure, the performance measures were calculated on Set 1 (5-fold cross validation). The results of calculating MAPE, R^2 and COD of the single predictors and the combined predictors on Set 2 are presented in Table 7 and Figure 8.

Table 7. Accuracy measurements: second-stage analysis

	MAPE	R^2	COD
OLS	11.7395	0.9095	11.748
SVR	6.7548	0.9497	6.7586
RF	5.0279	0.9725	5.0209
XGBoost	4.8234	0.9727	4.8331
LightGBM	5.0223	0.9740	5.0223
CatBoost	4.449	0.9777	4.4476
Naïve averaging	4.5055	0.9777	4.5041
Weighted averaging (θ = 3)	4.4086	0.9783	4.4078
ML-based voting	4.3788	0.9774	4.3894
Mixed	4.3209	0.9785	4.3201



a) Naïve averaging

b) Weighted averaging

c) Machine learning-based voting

d) Mixed

Figure 8. Scatter plots of the actual contract prices (depicted on the horizontal axis) and the predicted values (shown on the vertical axis) of the combined predictors

4.3.1. Naïve averaging

The results of the naïve averaging approach show that even when the relative performance of models is ignored, it is helpful to improve the predictive accuracy by taking the arithmetic mean of the predictors. If the naïve combination cannot provide an improvement at all by attribution, the performance of the combined predictor would be the average performance of the predictors combined. We could find that the naïve averaging predictor is better in MAPE (4.5) than the average MAPE of predictors combined (5.16). The predictive accuracy of the combined predictor is closest to the most performative algorithm (CatBoost, 4.49).

4.3.2. Weighted averaging with parameterization

In combining predictors, it is helpful to consider that predictor performance differs. Since a higher accuracy implies that the algorithm better captures the complexity of the real world, we can impose a greater weight on the predictor based on the algorithm. As suggested in the previous section, we set a parameter for the allowable performance gap, θ, and the weights of the predictors were calculated based on this gap.

Table 7 demonstrates that the weighted average combination provides an improvement in prediction accuracy. The MAPE from the weighted averaging predictor is 4.4076, an improvement on the naïve averaging predictor. The performance of the combined predictor is better than that of the most performative single-algorithm predictor. The result indicates that prediction errors due to modeling can be offset by the combination of multiple predictors, and the weighted averaging approach is an efficient way of combining single-algorithm predictors. When each algorithm fails to capture the complexity of the real market, the predictors tend to over- or under-estimate some cases. However, if those algorithms are not biased in the same way (i.e., the errors are independent), the error from one model could be neutralized by errors from another model, at least partially.

Another virtue of this combination approach is that it reduces the modeling costs incurred in the practice of mass appraisal. Sometimes a modeler is required to use various algorithms, but it is difficult for him or her to know in advance which algorithm is suitable. This problem can be solved by employing this approach as unsuitable algorithms are automatically ignored in the combined predictor. The modeler is only required to set a single parameter, θ.

4.3.3. Machine learning (ML)-based voting and averaging

Next, the features of ML-based voting are discussed. Before the combination model is examined, we investigate whether the performance of single-algorithm predictors can be trained using ML algorithms. As discussed, if some errors are related to the characteristics of the property, some variations in predictive accuracy would be expected. At first, we train the five ML algorithms (i.e., SVR, RF, XGBoost, LightGBM, and CatBoost) for the absolute percentage errors obtained from Set 1. Then, we construct the combined model, and the model will be evaluated in Set 2.

Table 8 shows the correlation coefficient between the predicted performances and the actual performances. We could find that there are weakly positive correlations (around 0.2 to 0.4), which implies that performances themselves contain predictable components and that the ML algorithms can (at least partially) capture them. If ML algorithms can predict how predictor performance changes, the predicted performance can be used in constructing a more sophisticated predictor by choosing the most efficient algorithm for a different observation.

We apply the ML algorithms to predict which predictor is the best for each sample. On Set 1, the most accurate single-algorithm model is trained. Five ML classifiers are trained for the most performative algorithm in Set 1. In the evaluation process, each classifier will recommend one algorithm for each sample in Set 2. Finally, the predictions from the recommended algorithms (there are five recommended algorithms for each sample) are combined using soft voting (an arithmetic average).

Table 8. Correlation coefficients of predicted and actual percentage error

		Actual absolute percentage error				
		SVR	RF	XGBoost	LightGBM	CatBoost
Predicted absolute percentage error	SVR	0.186	0.164	0.124	0.108	0.085
	RF	0.356	0.400	0.365	0.348	0.326
	XGBoost	0.329	0.366	0.335	0.281	0.287
	LightGBM	0.318	0.373	0.336	0.291	0.284
	CatBoost	0.335	0.384	0.323	0.324	0.289

Note: Numeric values were rounded off to four decimal places. Rows indicate the predicted model accuracy obtained by each algorithm. Columns indicate the actual accuracy obtained by each algorithm. For example, the value in the second row and third column (i.e., 0.365) indicates the correlation coefficient between the actual absolute percentage error and the predicted absolute percentage error of the XGBoost predictor, which is predicted using the RF model.

Table 9. Number of recommendations in machine learning (ML)-based voting (N = 11,380)

		Recommended				
		SVR	RF	XGBoost	LightGBM	CatBoost
Algorithm used	SVR	1,455	1,309	1,240	1,361	6,015
	RF	2,237	1,986	1,781	2,120	3,256
	XGBoost	0	0	0	0	11,380
	LightGBM	2,220	2,007	1,878	2,120	3,155
	CatBoost	2,291	1,901	1,973	1,928	3,287

Note: For example, the number in the second row and third column (i.e., 1,781) indicates the number that XGBoost predicted using RF as the most performative algorithm.

In Table 7, we can find that the predictor obtained from ML-based voting also provides a performance improvement. The MAPE of the combined predictor (4.3796) is less than not only that of the most performative single-algorithm predictor (CatBoost, 4.449) but also that of naïve averaging (4.5055) and that of weighted averaging (4.4076). This implies that the pattern in the prediction residuals of an algorithm can be further analyzed by using another algorithm. If a certain predictor could better capture the complexities of the market than other predictors in some cases, and those pattern could be detected by ML algorithms, the algorithms can also be used in the voting process. The result of voting (the combined predictor obtained) would be more powerful than the most performative single-algorithm predictor would be.

Table 9 shows the number of recommendations from each classifier. The recommendation pattern partially reflects the relative performance of the algorithms. At first, in all classifiers, CatBoost seems the most frequently recommended. This might relate to the relative performance of the model, as CatBoost is the most performative predictor in the comparison of single-algorithm predictors (see Table 7). For the 11,380 samples in the test set, CatBoost is recommended 6,015, 3,256, 11,380, 3,155, and 3,287 times by SVR, RF, XGBoost, LightGBM, and CatBoost, respectively. The second-most selected algorithm is SVR. This is interesting because the performance of the SVR predictor is the poorest among the single-algorithm predictors. As discussed in the previous section, all of the other algorithms, except for SVR, are based on the DT algorithm. This implies that the mechanism of SVR is different from that of the others, so this can compensate for capturing complexity, which is not captured well by the DT-based models.

4.3.4. Mixed predictor

Last, we examine the performance of the combination between the weighted averaging predictor and ML-based voting. The weighted averaging method and the ML recommendation method rely on different combinatorial advantages, so balancing those combination approaches might further improve the performance of mass appraisal.

For simplicity, the two predictors are averaged with the same weight (1:1). The accuracy of the mixed approach is shown in Table 7. Interestingly, we found that the predictive accuracy can be increased further from the combination of predictors.

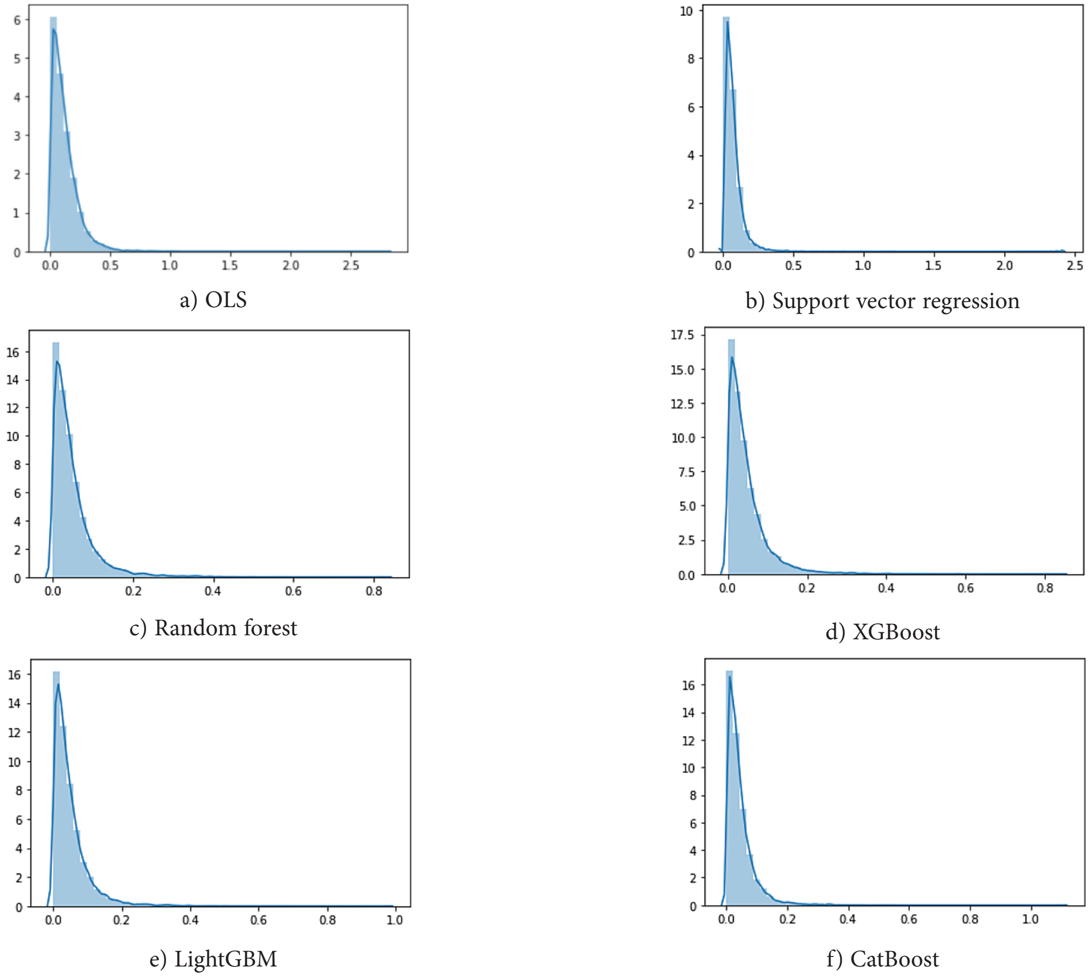
4.4. Distributions and features

The distributions of the absolute percentage errors are shown in Figures 9 and 10, and the corresponding statistics are presented in Tables 10 and 11.

The skewness of the predictors is around 3–4, except for the SVR predictor (= 9). It happens because we take the absolute value of percentage errors. In the same context, we could find that the median percentage errors are significantly lower than the means. The high level of skewness of the SVR predictor implies that it may make extraordinary errors more frequently. This is also revealed in a comparison of 75% quantile points. The 75% quantile point of the SVR predictor is 8.79, which is significantly higher than that of the other ML predictors (around 6).

We investigate the occurrence of outliers in predictions because they are particularly undesirable in the practice of mass appraisals. Tables 12 and 13 compare the frequency of outliers obtained from all of the single-algorithm predictors and the combined predictors. The percentages for which the error of prediction exceeds 25%, 50%, 75%, and 100% are presented, respectively. Under these criteria, we may find that the occurrence of outliers is markedly reduced in the DT-based ML algorithms (i.e., RF, XGBoost, LightGBM, CatBoost). If we define the outliers as deviations greater than 50% from the actual value, then about 1% of the OLS-based predictions and 0.3% of SVR predictions are revealed outliers in comparison with the approximately 0.08% of the tree-based algorithms. Moreover, the four tree-based algorithms make no errors over 100%.

We also find that the combined predictors have as high a level of stability in predictors as the tree-based algorithms do. However, the ML-based voting method creates a few more outliers than the weighted averaging method in the distribution of percentage error. Based on the 50% criteria, the proportion of percentage error of ML-based voting is double (0.061%) that of the weighted averaging method (0.035%).



a) OLS

b) Support vector regression

c) Random forest

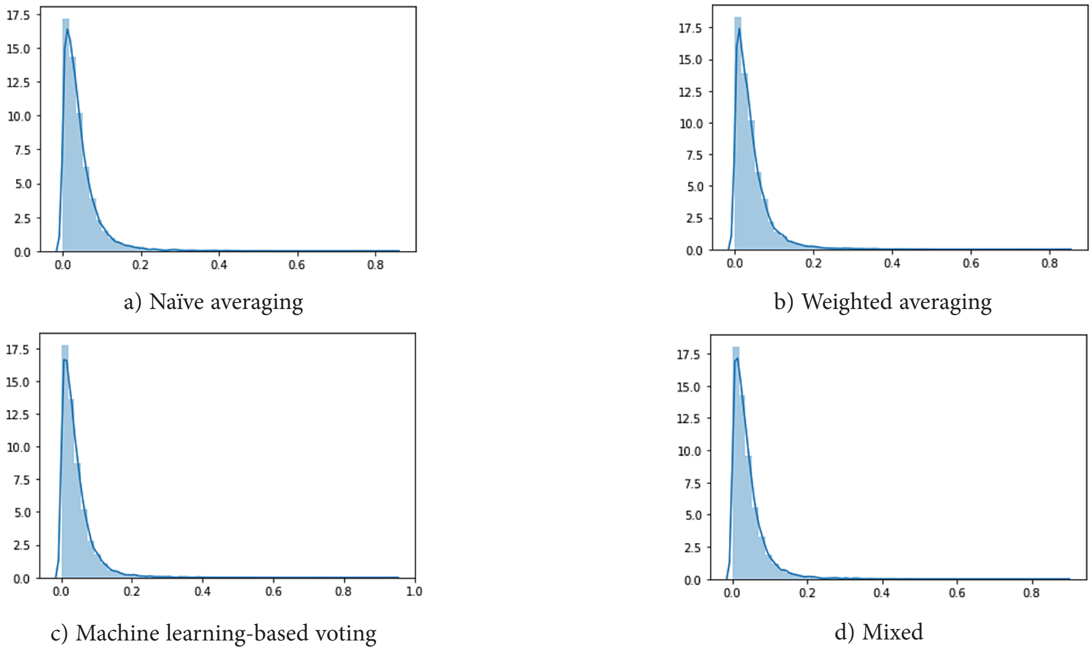
d) XGBoost

e) LightGBM

f) CatBoost

Note: The horizontal axis indicates the deviation of the ratio from the actual price. For example, a prediction error of 10% is expressed as 0.1. The vertical axis indicates the probability density function for the kernel density estimation.

Figure 9. Distribution of prediction errors (single predictors)



a) Naïve averaging

b) Weighted averaging

c) Machine learning-based voting

d) Mixed

Note: The horizontal axis indicates the deviation of the ratio from the actual price. For example, a prediction error of 10% is expressed as 0.1. The vertical axis indicates the probability density function for the kernel density estimation.

Figure 10. Distribution of prediction errors (combined models)

Table 10. Statistics for absolute percentage error: single-algorithm predictors

	OLS	SVR	RF	XGBoost	LightGBM	CatBoost
Standard deviation	11.428750	7.7838	5.5557	5.4530	5.564381	4.924121
Skewness	3.893300	3.5452	9.2260	3.6800	3.883400	4.529300
25% quantile	4.059275	2.4644	1.4995	1.4344	1.511737	1.379700
Median	8.893264	5.1735	3.3654	3.2559	3.354949	3.076825
75% quantile	16.059570	8.7968	6.2616	6.1183	6.321019	5.625845

Table 11. Statistics for absolute percentage error: combined predictors

	Naïve averaging	Weighted averaging	Machine learning (ML)-based voting	Mixed
Standard deviation	4.9184	4.8582	4.8934	4.7768
Skewness	3.8320	3.8530	4.5252	4.1148
25% quantile	1.4351	1.3459	1.3636	1.3569
Median	3.1294	3.0372	3.0209	2.9812
75% quantile	5.6871	5.5450	5.5829	5.4207

Table 12. Right-tail probability of absolute percentage error: single-algorithm predictors

	OLS	SVR	RF	XGBoost	LightGBM	CatBoost
>25%	9.525%	1.880%	1.326%	1.239%	1.3%	0.843%
>50%	0.975%	0.272%	0.079%	0.0105%	0.105%	0.087%
>75%	0.219%	0.123%	0.026%	0.026%	0.026%	0.017%
>100%	0.105%	0.079%	0%	0%	0%	0.008%

Table 13. Right-tail probability of absolute percentage error: combined predictors

	Naïve averaging	Weighted averaging	Machine learning (ML)-based voting	Mixed
>25%	0.913%	0.905%	0.843%	0.861%
>50%	0.052%	0.035%	0.061%	0.052%
>75%	0.017%	0.017%	0.026%	0.017%
>100%	0%	0%	0%	0%

Conclusions

In this paper, several ML-based prediction models are examined as automatic valuation models, and we propose that the combined models improve predictive power. As single predictors, the SVR, RF, XGBoost, LightGBM, and CatBoost algorithms were employed. To construct the combined model based on single predictors, naïve averaging, weighted averaging, and the ML-based voting method were employed. The results indicate that predictive performance can be improved by a combination of single predictors using apartment transaction data for 2018 in Seoul, the capital of and largest city in South Korea. From South Korea's MOLIT, we collected data on all apartment transactions in 2018 and used about 76% of it (i.e., 56,897 observations) after excluding data with missing values. The dataset was randomly divided into training sets (consisting of 80% of all of the transactions) and test sets (20% of the transactions) to construct the combined model. Since the performance evaluation results of single predictors must construct the combined models, we randomly selected 80% of the data in the training set to train the single predictors and used the remaining 20% to assess the predictors.

Because of the performance evaluation of single predictors, the tree-based algorithms (i.e., RF, XGBoost, LightGBM, and CatBoost) were found to be superior to other algorithms (support vector, OLS regression). The MAPEs of the RF, XGBoost, LightGBM, and CatBoost algorithms are 5.03%, 4.82%, 5.02%, and 4.45%, respectively, while that of the OLS and SVR models is 11.74% and 6.75%, respectively. In particular, the performance of the algorithms based on the gradient-boosting tree was quite high. The CatBoost algorithm showed superior predictive power to the other algorithms with a MAPE and R^2 value of 4.45% and 97.78%, respectively. On average, combinations of single predictors exhibited better performance than single predictors did. The MAPEs of the combined model based on naïve averaging, weighted averaging, and ML-based voting were 4.51%, 4.41%, and 4.38%, respectively. In the case of the naïve averaging method, the predictive accuracy was close to the most performative algorithm (CatBoost) even though the method simply takes the mean. In the case of combinations based on weighted averaging or ML-based voting, the MAPE was less than that of single predictors. Furthermore, we found that the combined predictors exhibited as high a level of stability in predictors as the tree-based algorithms did. In outlier predictions, the probability of the predictions deviating more than 25% from the actual price was found to be 0.91%, 0.91%, and 0.84% for the combined models (i.e., naïve averaging, weighted averaging, and ML-based voting), while that of the predictions by single predictors (i.e., OLS, SVR, RF, XGBoost, LightGBM, and CatBoost) were 9.53%, 1.89%, 1.33%, 1.24%, 1.3%, and 0.843%, respectively.

From a theoretical perspective, our results on single predictors demonstrate that ML-based predictors are more performative than OLS regression-based predictors as mentioned in many previous studies are. Regarding the combined models, our results indicate that prediction errors due to modeling can be eliminated by constructing a combination of multiple predictors. The performance of the combined predictor (weighted averaging and ML-based voting) was better than that of the most performative single-algorithm predictor. This means that the errors from one model may be neutralized by the errors in another model if those algorithms are not biased in the same way (i.e., the errors are independent). In addition, the superiority of the combined model based on ML-based voting implies that the pattern in the prediction residuals of one algorithm can be further analyzed by employing another algorithm. When one algorithm predicts better than another algorithm in some cases and a pattern is detected, ML-based algorithms may be used in the voting process to determine the most predictive algorithm.

From a practical perspective, our model reduces the practitioner's modeling costs incurred in mass appraisals. Practitioners have recently been brought face-to-face with the various ML techniques being developed every day but cannot be proficient in all of the computational skills and relevant real estate issues. It is difficult for practitioners to know in advance which algorithm is most suitable for analysis. Our methods suggest that the quality of automatic valuation models can be improved by combining multiple ML models based on simple parameterization or the ML-based voting scheme. In this approach, employing various algorithms is less potentially harmful as any unsuitable algorithms would be eventually ignored in the combined predictor. This enables practitioners to simply examine the various ML models as possible for the combination.

Our research can be extended in several directions. First, it is possible to employ more diverse ML techniques to construct combined models. In the present study, most of the predictors used to construct the combined model are DT-based techniques. Note that the second-most selected algorithm was SVR in the combined model based on ML-based voting. The predictive power of a combined model may increase if it is composed of predictive models based on various principles. One idea would be to use an artificial neural network in the model. Various methods of combining predictors may also be considered. A second direction would be to analyze why there is a difference between the predicted values generated through the ML technique and the actual values. To use ML-based valuation models in practice, it is necessary to determine the conditions under which ML techniques provide inaccurate prediction values. The results of this study indicate that the errors in valuation models may also be analyzed using ML techniques. This is useful from a practical point of view if we can derive the conditions for using the ML-based valuation model. Developing a reliable housing price index based on the ML technique is another idea for extending this research.

Funding

This work was supported by the Ministry of Education of the Republic of Korea and the National Research Foundation of Korea under Grant NRF-2021S1A5A2A03063960.

Author contributions

W. Kim and J. Hong conceived the study and were responsible for the design and development of the data analysis. J. Hong was responsible for data analysis and W. Kim was responsible for the interpretation of the results.

Disclosure statement

There are no conflicts of interest.

References

Adamczyk, T., & Bieda, A. (2015). The applicability of time series analysis in real estate valuation. *Geomatics and Environmental Engineering*, *9*(2), 15–25. https://doi.org/10.7494/geom.2015.9.2.15

Amit, Y., & Geman, D. (1997). Shape quantization and recognition with randomized trees. *Neural Computation*, *9*(7), 1545–1588. https://doi.org/10.1162/neco.1997.9.7.1545

Antipov, E. A., & Pokryshevskaya, E. B. (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. https://doi.org/10.1016/j.eswa.2011.08.077

Bellotti, A. (2017). Reliable region predictions for automated valuation models. *Annals of Mathematics and Artificial Intelligence*, *81*(1–2), 71–84. https://doi.org/10.1007/s10472-016-9534-6

Binoy, B. V., Naseer, M. A., Kumar, P. A., & Lazar, N. (2022). A bibliometric analysis of property valuation research. *International Journal of Housing Markets and Analysis*, *15*(1), 35–54. https://doi.org/10.1108/IJHMA-09-2020-0115

Bogin, A. N., & Shui, J. (2020). Appraisal accuracy and automated valuation models in rural areas. *Journal of Real Estate Finance and Economics*, *60*(1–2), 40–52. https://doi.org/10.1007/s11146-019-09712-0

Boser, B. E., Guyon, I. M., & Vapnik, V. N. (1992, July). A training algorithm for optimal margin classifiers. In *Proceedings of the 5th Annual Workshop on Computational Learning Theory* (pp. 144–152). Association for Computing Machinery. https://doi.org/10.1145/130385.130401

Breiman, L. (2001). Random forests. *Machine Learning*, *45*(1), 5–32. https://doi.org/10.1023/A:1010933404324

Cannon, S. E., & Cole, R. A. (2011). How accurate are commercial real estate appraisals? Evidence from 25 years of NCREIF sales data. *Journal of Portfolio Management*, *37*(5), 68–88. https://doi.org/10.3905/jpm.2011.37.5.068

Chau, K. W., & Chin, T. L. (2003). A critical review of literature on the hedonic price model. *International Journal for Housing Science and its Applications*, *27*(2), 145–165.

Chau, K., Wong, S., Yiu, C., & Leung, H. (2005). Real estate price indices in Hong Kong. *Journal of Real Estate Literature*, *13*(3), 337–356. https://doi.org/10.1080/10835547.2005.12090166

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. https://doi.org/10.3846/1648715X.2016.1259190

Chen, T., & Guestrin, C. (2016, August). Xgboost: a scalable tree boosting system. In *Proceedings of the 22nd International Conference on Knowledge Discovery and Data Mining* (pp. 785–794). Association for Computing Machinery. https://doi.org/10.1145/2939672.2939785

Chris, A. (2020, July 15). *Price rankings by city of price per square meter to buy apartment in city centre (buy apartment price)*. https://www.numbeo.com/cost-of-living/city_price_rankings?itemId=100

Čeh, M., Kilibarda, M., Lisec, A., & Bajat, B. (2018). Estimating the performance of random forest versus multiple regression for predicting prices of the apartments. *ISPRS International Journal of Geo-Information*, *7*(5), 168. https://doi.org/10.3390/ijgi7050168

Deaconu, A., Buiga, A., & Tothăzan, H. (2022). Real estate valuation models performance in price prediction. *International Journal of Strategic Property Management*, *26*(2), 86–105. https://doi.org/10.3846/ijspm.2022.15962

Dimopoulos, T., Tyralis, H., Bakas, N. P., & Hadjimitsis, D. (2018). Accuracy measurement of random forests and linear regression for mass appraisal models that estimate the prices of residential apartments in Nicosia, Cyprus. *Advances in Geosciences*, *45*, 377–382. https://doi.org/10.5194/adgeo-45-377-2018

Do, A. Q., & Grudnitski, G. (1992). A neural network approach to residential property appraisal. *Real Estate Appraiser*, *58*(3), 38–45.

Dorogush, A. V., Ershov, V., & Gulin, A. (2018). Catboost: gradient boosting with categorical features support. *arXiv preprint arXiv:1810.11363*.

Dubin, R. A., & Sung, C. H. (1990). Specification of hedonic regressions: non-nested tests on measures of neighborhood quality. *Journal of Urban Economics*, *27*(1), 97–110. https://doi.org/10.1016/0094-1190(90)90027-K

Fan, G. Z., Ong, S. E., & Koh, H. C. (2006). Determinants of house price: a decision tree approach. *Urban Studies*, *43*(12), 2301–2315. https://doi.org/10.1080/00420980600990928

Feng, S. T., Peng, C. W., Yang, C. H., & Chen, P. W. (2021). Nonlinear relationships between house size and price. *International Journal of Strategic Property Management*, *25*(3), 240–253. https://doi.org/10.3846/ijspm.2021.14607

Fletcher, M., Gallimore, P., & Mangan, J. (2000). Heteroscedasticity in hedonic house price models. *Journal of Property Research*, *17*(2), 93–108. https://doi.org/10.1080/095999100367930

Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. *Annals of Statistics*, *29*(5), 1189–1232. https://doi.org/10.1214/aos/1013203451

Friedman, J. H. (2002). Stochastic gradient boosting. *Computational Statistics and Data Analysis*, *38*(4), 367–378. https://doi.org/10.1016/S0167-9473(01)00065-2

Gabrielli, L., & French, N. (2021). Pricing to market: property valuation methods–a practical review. *Journal of Property Investment & Finance*, *39*(5), 464–480. https://doi.org/10.1108/JPIF-09-2020-0101

Garrod, G. D., & Willis, K. G. (1992). Valuing goods' characteristics: an application of the hedonic price method to environmental attributes. *Journal of Environmental Management*, *34*(1), 59–76. https://doi.org/10.1016/S0301-4797(05)80110-0

Glumac, B., & Des Rosiers, F. (2021). Practice briefing–Automated valuation models (AVMs): their role, their advantages and their limitations. *Journal of Property Investment and Finance*, *39*(5), 481–491. https://doi.org/10.1108/JPIF-07-2020-0086

Gnat, S. (2021). Property mass valuation on small markets. *Land*, *10*(4), 388. https://doi.org/10.3390/land10040388

Guo, J. Q., Chiang, S. H., Liu, M., Yang, C. C., & Guo, K. Y. (2020). Can machine learning algorithms associated with text mining from internet data improve housing price prediction performance? *International Journal of Strategic Property Management*, *24*(5), 300–312. https://doi.org/10.3846/ijspm.2020.12742

Han, X., & Clemmensen, L. (2014). On weighted support vector regression. *Quality and Reliability Engineering International*, *30*(6), 891–903. https://doi.org/10.1002/qre.1654

Hannonen, M. (2005). An analysis of land prices: a structural time-series approach. *International Journal of Strategic Property Management*, *9*(3), 145–172. https://doi.org/10.3846/1648715X.2005.9637534

Ho, T. K. (1995, August). Random decision forests. In *Proceedings of the 3rd International Conference on Document Analysis and Recognition* (Vol. 1, pp. 278–282). IEEE Publications.

Ho, W. K., Tang, B. S., & Wong, S. W. (2021). Predicting property prices with machine learning algorithms. *Journal of Property Research*, *38*(1), 48–70. https://doi.org/10.1080/09599916.2020.1832558

Hong, J., Choi, H., & Kim, W. S. (2020). A house price valuation based on the random forest approach: the mass appraisal of residential property in South Korea. *International Journal of Strategic Property Management*, *24*(3), 140–152. https://doi.org/10.3846/ijspm.2020.11544

Huh, S., & Kwak, S. J. (1997). The choice of functional form and variables in the hedonic price model in Seoul. *Urban Studies*, *34*(7), 989–998. https://doi.org/10.1080/0042098975691

Yeap, G. P., & Lean, H. H. (2020). Nonlinear relationship between housing supply and house price in Malaysia. *International Journal of Strategic Property Management*, *24*(5), 313–322. https://doi.org/10.3846/ijspm.2020.12343

Yilmazer, S., & Kocaman, S. (2020). A mass appraisal assessment study using machine learning based on multiple regression and random forest. *Land Use Policy*, *99*, 104889. https://doi.org/10.1016/j.landusepol.2020.104889

Yu, D. (2007). Modeling owner-occupied single-family house values in the city of Milwaukee: a geographically weighted regression approach. *GIScience and Remote Sensing*, *44*(3), 267–282. https://doi.org/10.2747/1548-1603.44.3.267

Kain, J. F., & Quigley, J. M. (1970). Measuring the value of housing quality. *Journal of the American Statistical Association*, *65*(330), 532–548. https://doi.org/10.1080/01621459.1970.10481102

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., & Liu, T.-Y. (2017). LightGBM: a highly efficient gradient boosting decision tree. *Advances in Neural Information Processing Systems*, *30*, 3146–3154.

Kittler, J., Hatef, M., Duin, R. P. W., & Matas, J. (1998). On combining classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *20*(3), 226–239. https://doi.org/10.1109/34.667881

Kok, N., Koponen, E. L., & Martínez-Barbosa, C. A. (2017). Big data in real estate? From manual appraisal to automated valuation. *Journal of Portfolio Management*, *43*(6), 202–211. https://doi.org/10.3905/jpm.2017.43.6.202

Kryvobokov, M., & Wilhelmsson, M. (2007). Analysing location attributes with a hedonic model for apartment prices in Donetsk, Ukraine. *International Journal of Strategic Property Management*, *11*(3), 157–178. https://doi.org/10.3846/1648715X.2007.9637567

Krogh, A., & Vedelsby, J. (1995). Neural network ensembles, cross validation, and active learning. *Advances in Neural Information Processing Systems*, *7*, 231–238.

Lancaster, K. J. (1966). A new approach to consumer theory. *Journal of Political Economy*, *74*(2), 132–157. https://doi.org/10.1086/259131

Lee, T. W., & Chen, K. (2016). *Prediction of house unit price in Taipei City using support vector regression* [Conference presentation]. Asia Pacific Industrial Engineering and Management Systems Conference, Taipei City, China.

Levantesi, S., & Piscopo, G. (2020). The importance of economic variables on London real estate market: a random forest approach. *Risks*, *8*(4), 112. https://doi.org/10.3390/risks8040112

Li, M. M., & Brown, H. J. (1980). Micro-neighborhood externalities and hedonic housing prices. *Land Economics*, *56*(2), 125–141. https://doi.org/10.2307/3145857

Liaw, A., & Wiener, M. (2002). Classification and regression by random forest. *R News*, *2*(3), 18–22.

Limsombunchai, V. (2004, June). House price prediction: hedonic price model vs. artificial neural network. In *New Zealand Agricultural and Resource Economics Society Conference* (pp. 25–26), Blenheim, New Zealand.

Lin, H., & Chen, K. (2011, July). Predicting price of Taiwan real estates by neural networks and support vector regression. In *Proceedings of the 15th WSEAS International Conference on Systems* (pp. 220–225), Corfu Island, Greece.

Liu, C. L. (2005). Classifier combination based on confidence transformation. *Pattern Recognition*, *38*(1), 11–28. https://doi.org/10.1016/j.patcog.2004.05.013

Lu, C. J., Lee, T. S., & Chiu, C. C. (2009). Financial time series forecasting using independent component analysis and support vector regression. *Decision Support Systems*, *47*(2), 115–125. https://doi.org/10.1016/j.dss.2009.02.001

Malpezzi, S. (2003). Hedonic pricing models: a selective and applied review. *Housing Economics and Public Policy*, *1*, 67–89. https://doi.org/10.1002/9780470690680.ch5

McCluskey, W. J., Deddis, W. G., Lamont, I. G., & Borst, R. A. (2000). The application of surface generated interpolation models for the prediction of residential property values. *Journal of Property Investment and Finance*, *18*(2), 162–176. https://doi.org/10.1108/14635780010324321

McCluskey, W., & Anand, S. (1999). The application of intelligent hybrid techniques for the mass appraisal of residential properties. *Journal of Property Investment and Finance*, *17*(3), 218–239. https://doi.org/10.1108/14635789910270495

McCluskey, W., Davis, P., Haran, M., McCord, M., & McIlhatton, D. (2012). The potential of artificial neural networks in mass appraisal: the case revisited. *Journal of Financial Management of Property and Construction*, *17*(3), 274–292. https://doi.org/10.1108/13664381211274371

McMillan, M. L., Reid, B. G., & Gillen, D. W. (1980). An extension of the hedonic approach for estimating the value of quiet. *Land Economics*, *56*(3), 315–328. https://doi.org/10.2307/3146034

Merz, C., & Pazzani, M. (1996). Combining neural network regression estimates with regularized linear weights. *Advances in Neural Information Processing Systems*, *9*, 564–570.

Myles, A. J., Feudale, R. N., Liu, Y., Woody, N. A., & Brown, S. D. (2004). An introduction to decision tree modeling. *Journal of Chemometrics*, *18*(6), 275–285. https://doi.org/10.1002/cem.873

Pace, R. K., & Hayunga, D. (2020). Examining the information content of residuals from hedonic and spatial models using trees and forests. *Journal of Real Estate Finance and Economics*, *60*(1–2), 170–180. https://doi.org/10.1007/s11146-019-09724-w

Pagourtzi, E., Assimakopoulos, V., Hatzichristos, T., & French, N. (2003). Real estate appraisal: a review of valuation methods. *Journal of Property Investment & Finance*, *21*(4), 383–401. https://doi.org/10.1108/14635780310483656

Pi-ying, L. (2011). Analysis of the mass appraisal model by using artificial neural network in Kaohsiung city. *Journal of Modern Accounting and Auditing*, *7*(10), 1080.

Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A. V., & Gulin, A. (2017). Catboost: unbiased boosting with categorical features. *arXiv preprint arXiv:1706.09516*.

Raymond, Y. C. (1997). An application of the ARIMA model to real-estate prices in Hong Kong. *Journal of Property Finance*, *8*(2), 152–163. https://doi.org/10.1108/09588689710167843

Rosen, S. (1974). Hedonic prices and implicit markets: product differentiation in pure competition. *Journal of Political Economy*, *82*(1), 34–55. https://doi.org/10.1086/260169

Safavian, S. R., & Landgrebe, D. (1991). A survey of decision tree classifier methodology. *IEEE Transactions on Systems, Man, and Cybernetics*, *21*(3), 660–674. https://doi.org/10.1109/21.97458

Selim, H. (2009). Determinants of house prices in Turkey: hedonic regression versus artificial neural network. *Expert Systems with Applications*, *36*(2), 2843–2852. https://doi.org/10.1016/j.eswa.2008.01.044

Sheppard, S. (1999). Chapter 41 Hedonic analysis of housing markets. *Handbook of Regional and Urban Economics*, *3*, 1595–1635. https://doi.org/10.1016/S1574-0080(99)80010-8

Sims, S., Dent, P., & Oskrochi, G. R. (2008). Modelling the impact of wind farms on house prices in the UK. *International Journal of Strategic Property Management*, *12*(4), 251–269. https://doi.org/10.3846/1648-715X.2008.12.251-269

Sing, T. F., Yang, J. J., & Yu, S. M. (2022). Boosted tree ensembles for artificial intelligence based automated valuation models (AI-AVM). *Journal of Real Estate Finance and Economics*, *65*, 649–674. https://doi.org/10.1007/s11146-021-09861-1

Smola, A. J., & Schölkopf, B. (2004). A tutorial on support vector regression. *Statistics and Computing*, *14*(3), 199–222. https://doi.org/10.1023/B:STCO.0000035301.49549.88

Song, Y. Y., & Lu, Y. (2015). Decision tree methods: applications for classification and prediction. *Shanghai Archives of Psychiatry*, *27*(2), 130–135.

Taniguchi, M., & Tresp, V. (1997) Averaging regularized estimators. *Neural Computation*, *9*(5), 1163–1178. https://doi.org/10.1162/neco.1997.9.5.1163

Torres-Pruñonosa, J., García-Estévez, P., & Prado-Román, C. (2021). Artificial neural network, quantile and semi-log regression modelling of mass appraisal in housing. *Mathematics*, *9*(7), 783. https://doi.org/10.3390/math9070783

Verikas, A., Lipnickas, A., & Malmqvist, K. (2002). Selecting neural networks for a committee decision. *International Journal of Neural Systems*, *12*(5), 351–361. https://doi.org/10.1142/S0129065702001229

Verikas, A., Lipnickas, A., Malmqvist, K., Bacauskiene, M., & Gelzinis, A. (1999). Soft combination of neural classifiers: a comparative study. *Pattern Recognition Letters*, *20*(4), 429–444. https://doi.org/10.1016/S0167-8655(99)00012-4

Wang, D., & Li, V. J. (2019). Mass appraisal models of real estate in the 21st century: a systematic literature review. *Sustainability*, *11*(24), 7006. https://doi.org/10.3390/su11247006

Wikimedia Commons. (2005). *Districts of Seoul* [Digital image]. https://commons.wikimedia.org/wiki/File:Map_Seoul_districts_de.png

Zhou, G., Ji, Y., Chen, X., & Zhang, F. (2018). Artificial neural networks and the mass appraisal of real estate. *International Journal of Online Engineering*, *14*(3), 180–187. https://doi.org/10.3991/ijoe.v14i03.8420

Zurada, J., Levitan, A., & Guan, J. (2011). A comparison of regression and artificial intelligence methods in a mass appraisal context. *Journal of Real Estate Research*, *33*(3), 349–388. https://doi.org/10.1080/10835547.2011.12091311

Appendix 1. Hedonic pricing model

The formula of the hedonic model is expressed as follows:

$$\ln P_i = \beta_0 + \sum_{s=1}^{n_s}\beta_s x_{s,i} + \sum_{l=1}^{n_l}\beta_l x_{l,i} + \sum_{t=1}^{n_t}\sum_{d=1}^{n_d}\beta_{d,t} I_d I_t + \varepsilon_i,$$

where $\ln P_i$ is the logarithm of housing prices for the i th observation. x_s indicates the structural attributes including years elapsed, property size (the natural logarithm of m^2), floor level, number of bedrooms, number of bathrooms, heating system (dummy), hallway type (dummy), number of households in the complex, average number of parking spots in the complex, FAR, BCR, and the highest and lowest floors in the complex. x_l is the accessibility measurement. It contains the Euclidean distance from the closest subway station, national park, elementary school, middle school, high school, university, museum, and government office.

We also considered the spatiotemporal dummies as discussed in Pace and Hayunga (2019). I_d indicates the location dummy based on the administrative district, "Dong". Dong is the most granular level of administrative districts, and functions similarly to a zip code dummy. I_d contains 264 variables. I_t is the temporal dummy variable that indicates the month of the contract data. Because we examine data for 2018, it contains 12 variables. The spatiotemporal dummy is the interaction between the spatial and temporal dummies. It contains 2,678 variables (264 × 12 = 3,168); however, 490 variables are omitted due to a lack of observations. Lastly, ε_i is an error term that independently follows a normal distribution.

Appendix 2. Hyperparameters

The hyperparameters of the optimized machine learning models are as follows.

SVR	Random forest	XGBoost	LightGBM	CatBoost
C: 50 degree: 3 epsilon: 0.1 gamma: 0.05 kernel: 'rbf'	max_depth: default max_features: default max_leaf_nodes: default max_samples: default min_samples_leaf: 1 min_samples_split: 2 min_weight_fraction_leaf: 0 n_estimators: 100	max_depth: 20 colsample_bylevel: default colsample_bynode: default colsample_bytree: default learning_rate: 0.1 min_child_weight: default reg_alpha: default reg_lambda: 2 n_estimators: 100	boosting_type: 'gbdt' class_weight: default colsample_bytree: 1.0 learning_rate: 0.1 max_depth: 200 min_child_samples: 20 min_child_weight: 0.001 reg_alpha: 0 reg_lambda: 0 n_estimator: 3000	Iterations: 4000 learning_rate: 0.1 depth: 10 l2_leaf_reg: 1 model_size_reg: 0.5 loss_function: 'RMSE'