© Statistical Research and Training Center

مجله‌ی پژوهش‌های آماری ایران
پژوهشکده‌ی آمار
دوره‌ی ۱۷، شماره‌ی ۱، بهار و تابستان ۱۳۹۹، صص ۱۹۱–۲۱۴

J. Statist. Res. Iran **17** (2020): 191–214
DOI: 10.52547/jsri.17.1.191
DOR: 20.1001.1.25385771.2020.17.1.10.1

Machine Learning Models for Housing Prices Forecasting using Registration Data

Mehdi Farahzadi[†], Rahman Farnoosh[‡,*] and Mohammad Hassan Behzadi[†]

[†] Islamic Azad University
[‡] Iran University of Science and Technoligy

Received: 2022/25/05 Approved: 2022/22/02

Abstract. This article has been compiled to identify the best model of housing price forecasting using machine learning methods with maximum accuracy and minimum error. Five important machine learning algorithms are used to predict housing prices, including Nearest Neighbor Regression Algorithm (KNNR), Support Vector Regression Algorithm (SVR), Random Forest Regression Algorithm (RFR), Extreme Gradient Boosting Regression Algorithm (XGBR), and the Long Short-Term Memory Neural Network Algorithm (LSTM). This research has been done using the data of the Statistics Center of Iran, which contains information on the purchase and sale of residential units in Tehran in the years 2014 to 2020 and includes 998299 transactions and 11 features. Loss of data, batch data conversion, normalization, etc. are performed on the housing data set to obtain the final and error-free data set. To divide the data set into training and test data sets, the important and practical method of cross-validation or K-Fold has been used because of its simplicity and effectiveness and as a universally valid method. Various evaluation criteria such as MSE, RMSE, MAE,ME and R^2 were used to compare the models and identify the best model. Comparison of models in terms of all evaluation criteria in all K-fold subsets proves the stability and superiority of the Extreme Gradient Boosting Regression model.

[*] Corresponding author

Keywords. Housing price forecasting, nearest neighbor regression, random forest regression, support vector regression, long short-Term memory neural network, and extreme gradient boosting regression.

MSC 2010: 62P20.

1 Introduction

Buying a house is without a doubt one of the most important decisions a person can make in their life. Housing price forecasting is vital not just for buyers but also for planners, policymakers, real estate firms, and mass builders.

Housing demand increases every year, which indirectly increases housing prices every year. Therefore, planners, buyers, builders of houses, and the real estate industry want to know the most important factors influencing housing prices.

Housing price forecasting is a complex nonlinear issue that is affected by several direct and indirect features (such as year of construction, area, etc.). Sophisticated price forecasting approaches focus on the nonlinear relationship between input and output data. (Wang et al. (2019)). Classic representatives of this group are regression methods, such as linear regression, support vector regression (SVR), nearest neighbor regression (KNNR), and random forest regression (RFR), which have been widely used to Forecast real estate prices and provide satisfactory results in the past. They have shown such as (Madhuri et al. (2019), Manasa et al. (2020), Lou et al. (2017), Ho et al. (2021), and Rafiei and Adeli (2016)).

Artificial neural network (ANN) methods have shown promising results when Forecasting housing prices. (Varma et al. (2018), Peter et al. (2020)). The ability to make nonlinear relationships between input and output variables is their main advantage. Therefore, they are used for nonlinear forecasts of housing prices. Recent studies have suggested that machine learning models, including ANNs, are effective for forecasting property prices. such as (El Hamzaoui and Perez (2011), Morano et al. (2015), Punia et al. (2020), Kang et al. (2020), Kim et al. (2020), and Abidoye and Chan (2017 and 2018)).

To employ machine learning algorithms, a large dataset is necessary, We had to check to see if we could acquire access to an important and accepted registration data source for forecasting housing prices in Tehran, which had

© 2020, SRTC Iran

never been done previously using a trusted source with a lot of data and machine learning algorithms. Therefore, we are looking for a way to forecast housing prices in the city of Tehran that has the advantages of important machine learning methods while its error is low.

Tehran's city data was taken from Iran's real estate transaction management information system from 1393 to 1399, encompassing 998299 transactions and 11 features. There are various machine learning algorithms for house price forecasting. In this study, five machine learning models were tested, including Nearest Neighbor Regression (KNNR), Support Vector Machine Regression (SVR), Random Forest Regression (RFR), and Extreme Gradient Boosting Regression (XGBR), and Long Short-Term Memory neural network (LSTM).

A series of initial preprocessing steps such as data wiping, missing data imputation, batch data conversion, normalization, etc. are performed on the housing dataset to achieve the final dataset without error.

Various criteria have been introduced to evaluate housing forecasting price models. In this paper, the main criteria were used for evaluating models, such as mean square error (MSE), root mean square error (RMSE), mean absolute error (MAE), maximum error (ME), and coefficient of determination (R^2).

2 Methodology

2.1 Nearest Neighbor Regression (KNNR)

The nearest neighbor regression algorithm is a statistical method used to estimate the value of an unknown point based on its nearest neighbors. The nearest neighbors are usually defined as the points that have the shortest distance from the vicinity of an unknown point to the unknown point itself. Several techniques are used to measure the distance between neighbors, the simplest of which is the Euclidean distance function. K Neighbor points that have the shortest distance to the unknown point are used to estimate its value utilizing Equation (1).

$$\hat{y}_i = \sum_{i=1}^{n} w_i y_i \qquad (1)$$

In Equation (1), is the weight of each point of a single neighbor to the desired point y. The nearest neighborhood regression algorithm defined in

Equation (1) is the weighted neighborhood mean. The simplest regression model of the nearest neighborhood is the mean proximity obtained for uniform weights in which all neighboring points have the same effect on estimating the desired value $\left(w_i = \frac{1}{n}\right)$. Conversely, different weights can be applied when it is assumed that neighboring points have a different effect on the estimated point. The simplest Equation (2) shows the weight function (Hamed et al. (2020)).

$$w_i = \frac{d_i}{\sum_{i=1}^{n} d_i} \qquad (2)$$

In Equation Equation (2), is the distance between the unknown point and its neighbor. The weight function must reach its maximum value at a distance of zero from the interpolated point, and as the distance increases, the function must decrease proportionally (Hamed et al. (2019)). The regression of the nearest neighborhood presented in Equation (2) depends only on the neighboring points. As a result, the estimating method ignores the whole dataset procedure.

2.2 Random Forest Regression (RFR) Algorithm

The random forest creates hundreds or even thousands of decision trees that operate as regression functions in and of themselves, with the average output of all these decision trees serving as the ultimate result of the random forest regression process (Li et al. (2018)). The tree grows during the learning process according to the complexity of the data entered. Each decision tree is made up of decision nodes and leaf nodes. The decision nodes evaluate each input sample with an experimental function and send it to different branches based on the characteristics of the sample. Suppose X represents the input vector containing m attribute such that $X = (X_1, Y_1, ..., X_m)$ and Y is the output scalar and is a training set containing n observations, which can be expressed as follows (Li et al., 2018):

$$S_n = \{x_n, y_n\}, X \varepsilon R^M, Y \varepsilon R \qquad (3)$$

During the training, the input data in each node is split by the algorithm, so that the parameters of the split functions are optimized to fit the set. In the first step, the decision tree must divide all variables in the best way possible. This process starts at the root, and each node applies its split function to the new X input. This is repeated in reverse until we reach a

© 2020, SRTC Iran

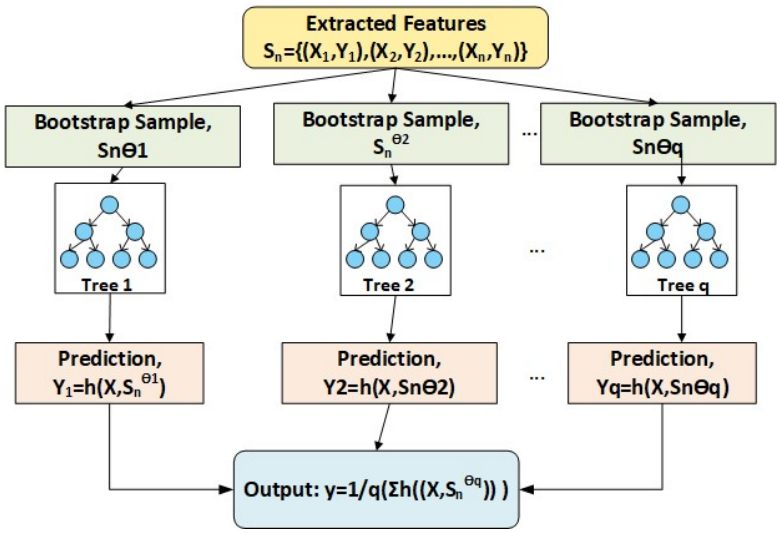

Figure 1. Random forest regression flowchart

final node (also called a tree leaf). When the maximum number of levels is reached or when a node with fewer observations than the predefined number is reached, the tree typically stops. At the end of this training process, a prediction function $h\left(X, S_n\right)$ is constructed on S_n.

The random forest regression algorithm's training step is to create many uncorrelated decision trees. A random forest regression technique creates each tree using a random selection of predictors and is therefore called a "random forest." The random forest regression algorithm uses a base classifier with a tree structure $h\left(X, \Theta_k\right)$ where k=1,...,L and Θ_k is a family of independent randomly distributed vectors and X is an input vector (Breiman (2001)). This is a group method that combines all the decision trees generated using an algorithm called begging. The process of collecting a random sample is called a bootstrap. The begging algorithm selects several bootstrap instances $\left(S_n^{\Theta_1}, ..., S_n^{\Theta_g}\right)$ and the algorithm applies the previous decision tree to these instances to set a prediction $h\left(X, S_n^{\Theta_1}\right), ..., h\left(X, S_n^{\Theta_g}\right)\Big)$. For each tree, this set generates the relevant outputs,$\hat{y_1} = h\left(X, S_n^{\Theta_1}\right), \hat{y_2} = h\left(X, S_n^{\Theta_2}\right), ..., \hat{y_g} = h\left(X, S_n^{\Theta_g}\right)\Big)$ Then the aggregation is done by averaging the outputs of all trees. As a result, the estimate Y of the output can be

obtained as follows (Rodriguez-Galiano et al. (2015)).

$$\hat{Y} = \frac{1}{g}\sum_{l=1}^{g}\hat{Y}_l = \frac{1}{g}\sum_{l=1}^{g}\hat{h}\left(X, S_n^{\Theta_l})\right) \qquad (4)$$

2.3 Support Vector Machine Regression (SVR)

Support vector machines are one of the most widely used computational intelligence techniques for forecasting and classification. The support vector machine regression algorithm is a version of the support vector machine algorithm that has been used successfully in many regression problems. The main purpose of this algorithm is to accurately fit a regression function, $y - f(x)$ in a model, to accurately $\{y_i\}$ the goals of a set of input samples $\{X_i\}$ with certain training data as $D = \{(X_1, y_1), (X_2, y_2), ..., ((X_N, y_N))\}$ forecast that $x_i \varepsilon R_N$ is the vector of the input variables and $y_i \varepsilon R_N$ is the corresponding scalar output value. For real world nonlinear problems in which the input data cannot be linearly correlated with the required output, the following linear model can be constructed in the property of higher dimensions using the nonlinear mapping function $g_i(x)$, which concerning Equation (5) has been shown (Ahmad et al. (2020)):

$$f(x, w) = \sum_{i=1}^{n} w_i g_i(x) + b \qquad (5)$$

In Equation (5), $g_i(x)$ is a function called feature, $w_i g_i(x)$ is the product of a point multiplied by the feature space F, and b is biased. This relationship is based on the principle of structural risk minimization. Suppose the normalized input variables contain the vectors X_i and X_i (i represents the i-th point of the dataset). In this case, a set of data points can be defined as $\{(X^i, Y_i)\}_{i=1}^{N}$, where N represents the total number of samples. A Support vector machine regression can be used to estimate the function using Equation (2.3) (Ahmad et al. (2018)).

$$Y = f(X) = W - \phi_{(X)} + b \qquad (6)$$

In Equation (6), represents the space with higher dimensions. A regular risk function presented in Equation (7) is utilized to estimate the coefficients W and b.

© 2020, SRTC Iran

$$Minimise : \frac{1}{2}\|w\|^2 + c\frac{1}{N}\sum_{i=1}^{N} L_\varepsilon\left(Y_i, f\left(x_i\right)\right) \qquad (7)$$

In Equation (7), $\|w\|^2$ is known as the regulatory expression, and C is the penalty parameter to determine the flexibility of the model.

2.4 Extreme Gradient Boosting Regression (XGBoostR)

The Extreme Gradient Boosting algorithm was proposed by (Chen and Guestrin (2016)). The Extreme Gradient Boosting algorithm is a regression tree whose decision rules are the same as the decision tree rules and support both regression and classification. This algorithm is an efficient and scalable version of the gradient boosting machine that has been widely used in computer vision, data mining, and other fields (Ma et al. (2020)). The Extreme Gradient Boosting technique is part of an advanced gradient boosting framework that is commonly utilized in machine learning because of its great efficiency and versatility. Another regularization term is added to the loss function of the Extreme Gradient Boosting regression technique to assist smooth the final learned weights and minimize overfitting (Chen and Guestrin (2016)). To optimize the loss function, this approach employs first- and second-order gradients. In addition to adding a regularization term to minimize overfitting, the Extreme Gradient Boosting regression approach also enables row and column sampling to overcome this problem, making the model faster to explore since parallel and distributed computations ensure faster learning (Huang et al. (2020)).

2.5 Long Short-Term Memory neural network (LSTM)

A Long Short-Term Memory neural network is a type of recurrent neural network (Bai et al. (2021)) that consists of three components or layers, including the input layer, the hidden layer, and the output layer. The network nodes in the hidden layer have connections, and the hidden layer inputs are at the same time the outputs of the input layer.

The number of input layer neurons is equal to the number of input variables and the number of output layer neurons is equal to the number of output variables, and the hidden layers of Long Short-Term Memory neural networks are composed of a memory cell. LSTM are superior to standard recurrent neural networks due to the presence of this memory cell, which helps

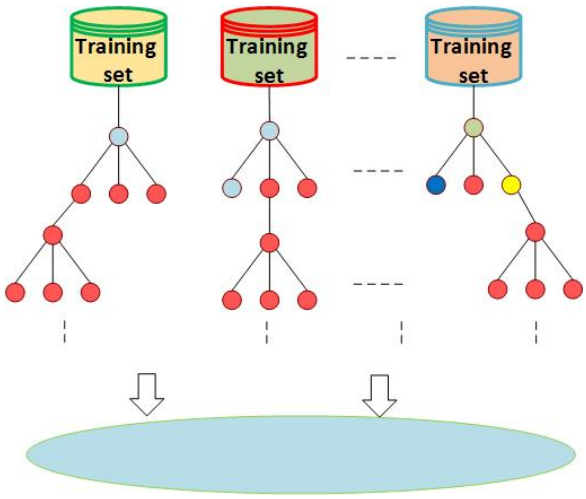

Figure 2. Extreme Gradient Boosting Regression Flowchart

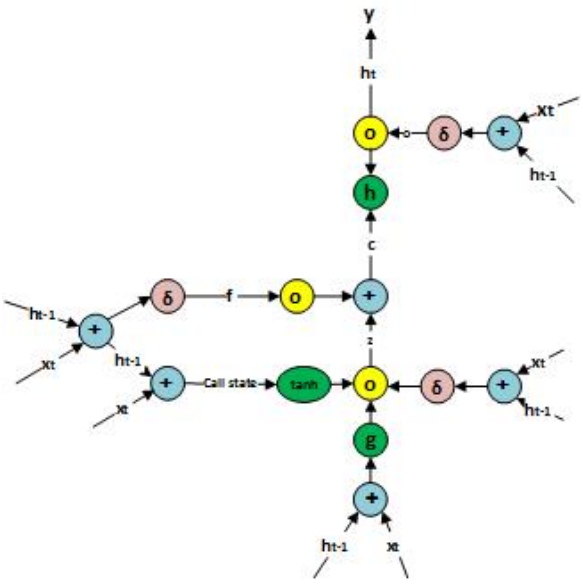

Figure 3. General structure of a Long Short-Term Memory neural network

© 2020, SRTC Iran

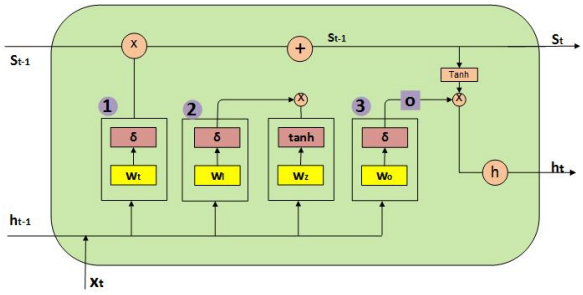

Figure 4. General structure of the memory cell architecture in the neural network of Long Short-Term Memory

retain information over time (Punia et al. (2020)). The Long Short-Term Memory neural network overcomes the Gradient fading problem for modelling long-term dependencies. Compared to other versions of the recursive neural network, this method calculates memory units using different activation functions. The main advantage of using a Long Short-Term Memory network, unlike the traditional neurons used in the recurrent neural network, is that their cellular state accumulates those activities over time. Because derivatives are distributed on the result of additions of sets, the derivatives do not disappear quickly and go back in time. In this way, the Long Short-Term Memory neural network can perform tasks across long sequences and discover long-term features (Abdel-Nasser and Mahmoud (2017)).

2.6 Dataset

The Statistical Center of Iran provided data for this article, which was edited from the Iranian Real Estate Management Information System. This system was created and used to organize housing matters by clarifying real estate transaction information. There were 998299 residential unit buying and sales transactions in Tehran from 2014 to 2020, with 11 features.

2.6.1 Preprocessing

This part of a series of initial preprocessing such as data imputation, batch data conversion, normalization, etc. will be done on the housing dataset to achieve the final dataset without error, which is described in each of the separate steps.

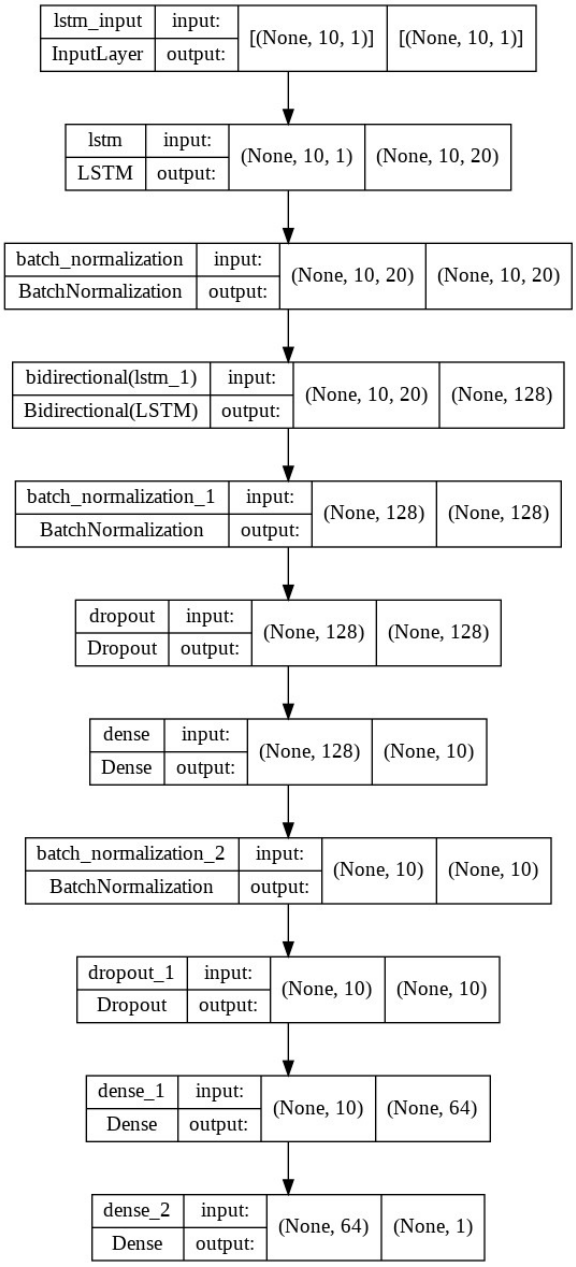

Figure 5. Complete Long Short-Term Memory neural network architecture

© 2020, SRTC Iran

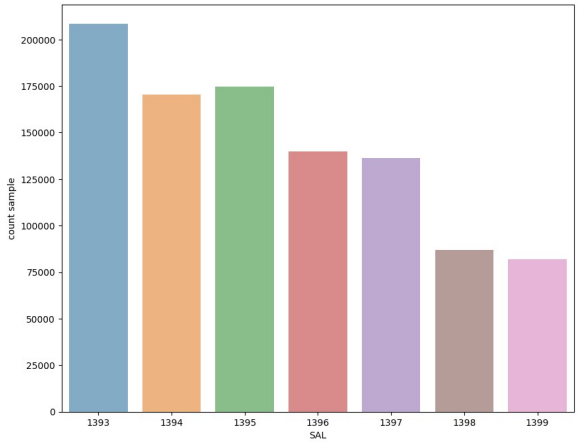

Figure 6. Housing dataset based on the number of samples available each year

Table 1. Housing dataset and its characteristics.

Feature name	Type	Range	Description
ID	Numeric	-	This identifier is used to identify the contract in the real estate transaction management information system
year	Numeric	1399-1393	The information collected is the real estate transaction management information system from 1393 to 1399.
Season	Numeric	1-4	This information includes the buying or selling seasons, including spring, summer, autumn, and winter.
Unit number of the municipality	Numeric	1-22	A residential unit's municipal area ranges from 1 to 22 districts.
postal code of a residential unit	Numeric	-	6-digit postal code of a residential unit.
Contract date	date	1399-1393	This feature includes the exact date of purchase or sale of the residential unit.
Type of Residential unit	Batch	1-4	The type of residential unit is considered in four forms: (1) apartments, (2) buildings, (3) villas, and (4) others.
Residential unit area	Numeric	1-15000	This feature shows the area of a residential unit based on square meters.
Contract cost	Numeric	10000-3000000000	Indicates the amount of the contract for the purchase or sale of a residential unit in thousand of Rials.
Skeleton type	Batch	1-4	The type of skeleton is considered in four forms: (1) metal, (2) concrete, (3) metal and concrete, and (4) others.
Building life	Numeric	1-100	This feature shows the lifespan of a residential unit by year.

Data Cleansing

In this section, a series of initial preprocessions such as missing data imputation, batch data conversion, normalization, etc. are performed on the housing dataset to achieve the final and error-free dataset. Each of these steps is described below.

Range of properties: At this stage, all the features are examined in terms of upper and lower limits. For example, the number of seasons should not be more than 4. All features will be considered in this step.

Feature extraction: In this step, the date yyyy/mm/dd is converted to year, month, and day, and it is also checked that the year data is not empty. Correction of letters: In the housing dataset, there are some features such as the type of skeleton in repetitive states such as,

(«بتونی»،«بتوني»)، («فلزی»،«فلزي»)

etc., all of which are standardized at this stage so that the type is not mistaken.

Sample deletion: Some data may not have all of the key features, such as the contract amount. At this point, these data samples will be removed from the dataset.

The Missing Data Imputation

There are two methods for missing data imputation; the middle one for numerical features and the maximum repetition method for batch features. In the dataset, there are two types of data, including the type of residential unit and the type of skeleton, in each of which the most repetitive feature replaces empty or lost data. For other numeric features, the median obtained for each feature is used to replace empty or missing data.

Convert Batch Features

In this step, a conversion will be performed on the housing dataset, which aims to convert the batch data type, including the housing unit type and the skeleton type, into numerical data for use in the models. In the real estate management information system of Iran, the type of residential unit is considered in four forms: apartment, building, villa, and others. Also in this system, the type of skeleton is considered in four forms: metal, concrete, metal and concrete, and others. In this paper, the Label Encoder method is used for this conversion. In this method, the feature of the type of residential

© 2020, SRTC Iran

unit after coding the features uniquely will be [apartment (0), building (1), villa (2), other (3)]. Finally, the characteristics of the type of residential unit become [0,1,2,3]. This procedure is also applied to the skeleton type attribute, converting all of the dataset's features to numeric types.

Normalization

One of the important steps is to use normalisation methods. Normalisation is performed to scale data values over a specific range of values. Most machine learning and deep learning algorithms do better price forecasts by normalising the data. The Min-Max normalisation method is one of the scaling methods that is very popular and causes the data to be in the range between [0 and 1].

2.7 Evaluation Criteria

Various criteria have been introduced to evaluate housing price forecasting models. In this paper, several main criteria such as mean square error (MSE), root mean square error (RMSE), mean absolute error (MAE), maximum error (ME), and coefficient of detection (R^2) are employed as key prediction criteria.

3 Findings

3.1 Simulation Environment

Housing price forecasting models are coded in the Python programming language and run in the Google Columbine environment. The Colab environment was created by Google to implement machine learning methods. Tensorflow, Keras, Sklearn, Scikit learn, and Xgboost libraries are used.

3.2 Optimal Adjustment of Parameters

The optimal parameters of housing price forecast models are given in Table 2. Of course, only some important parameters are listed here along with their values, and the default and low-impact parameters are avoided.

Table 2. Optimal adjustment of the parameters of the implemented algorithm.

Algorithm	Abbreviation	Parameter name	Amount
Regression of Nearest Neighbor	KNNR	metric	Minkowski
		n_neighbors	1000
		weights	uniform
Support Vector Machine Regression	SVR	max_iter	2000
		tol	1e-5
		C	1
Random Forest Regression	RFR	n_estimators	100
		max_features	sqrt
		max_depth	20
Extreme Gradient Boosting regression	XGBR	learning_rate	0.1
		n_estimators	100
		Min_child_weight	sqrt
		max_depth	1
Long Short-Term Memory Neural Network	LSTM	activation	relu
		optimizer	adam
		learning_rate	0.01
		epochs	100
		batch_size	3000

3.3 Training and Test Dataset

Because of its simplicity and efficacy as a universally valid approach for dividing the primary housing dataset into training and test datasets, the crucial and practical method of cross-validation, or K-Fold, was adopted. This model makes it possible to select samples for training and test datasets by reusing and constructively using the existing datasets in the main housing datasets, which provides better results in splitting test and training datasets than the base and random models. In the K-Fold method, the samples in the main housing data set called D are randomly divided into reciprocal K subsets of similar size, which can be represented as follows:

$$D = D_1 \cup D_2 \cup ... \cup D_k, D_i \cap D_j = 0\,(i \neq j) \qquad (8)$$

Each D_i subset maintains the compatibility of the data distribution as much as possible. All D_i subsets except one are utilised as training sets in this method, while the remaining subset is used as a test dataset.

In this paper, the total number of samples of the housing dataset is equal to 998299. In the best case, 25% of it can be considered each time for testing

© 2020, SRTC Iran

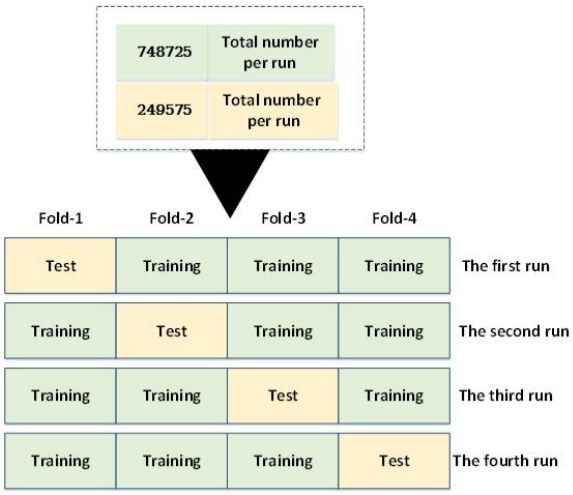

Figure 7. Flowchart of the quadruple cross-validation method on the housing dataset

and the remaining 75% as training, in which case k = 4 should be set. Thus, in this paper, the dataset is divided into four subsets, the number of samples in each training is equal to 748725, and the number of samples in the test data is equal to 249575, which is shown graphically in Figure 13 for better understanding.

3.4 Statistical Analysis of Housing Dataset Features

In this section, we will take a look at some of the features of the Housing Database. The main features such as the type of skeleton of the residential unit, the area of purchase of the residential unit, the building life of the residential unit, months, and purchasing seasons in terms of density and distribution from 1393 to 1399 have been studied. For a better understanding of each feature, it is shown graphically by year.

The degree of correlation between the characteristics of the main housing dataset in Figure 13 shows that the area of the residential unit has the highest correlation with the price of the residential unit with a value of 0.52, and on the other hand, the postal code of the residential unit has the highest correlation with the price of 0.14.

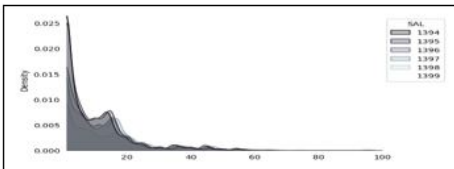

Figure 8: Density and distribution of building a life in the housing dataset separately each year

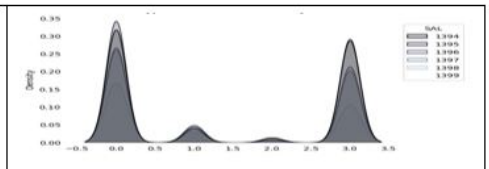

Figure 9: Density and distribution of skeletal type in the housing dataset separately each year

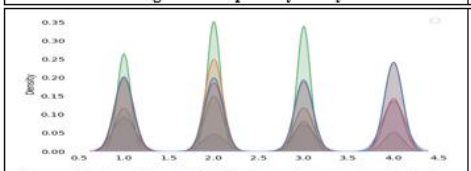

Figure 10: Density and distribution of quarterly sales in the housing dataset separately each year

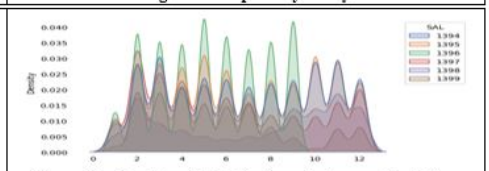

Figure 11: Density and distribution of sales months in the housing dataset separately each year

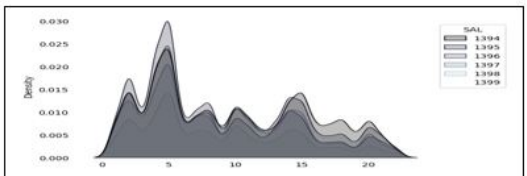

Figure 12: Density and distribution of the residential area of the municipality in the housing dataset separately each year

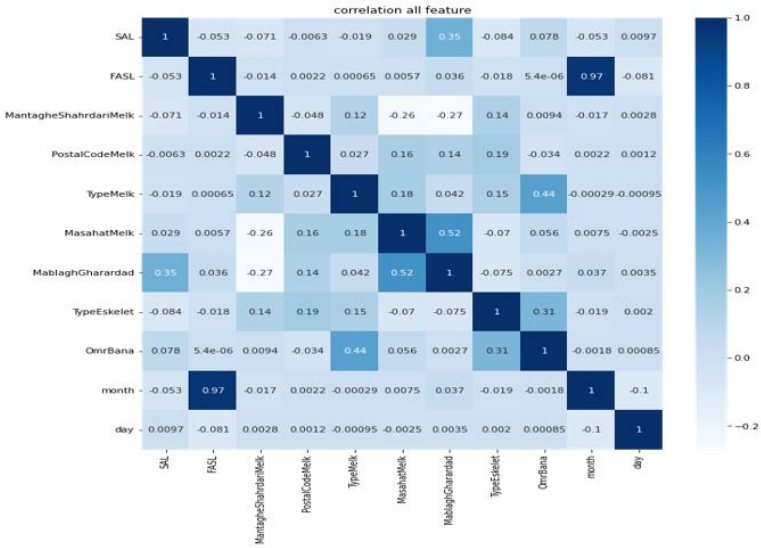

Figure 13. Correlation rate of features of the main housing dataset

© 2020, SRTC Iran

Table 3. Comparison of models on the Fold-1 test dataset.

Algorithm	MSE	RMSE	MAE	ME	R^2
KNNR	0.0000100	0.00316	0.00102	0.39011	0.52
SVR	0.0000147	0.00384	0.00148	0.38921	0.2
RFR	0.0000034	0.00186	0.00058	0.25757	0.83
XGBR	0.0000032	0.0018	0.00059	0.24232	0.84
LSTM	0.0000180	0.00424	0.00167	0.37243	0.15

Table 4. Comparison of models on the Fold-2 test dataset.

Algorithm	MSE	RMSE	MAE	ME	R^2
KNNR	0.0000104	0.0032	0.001	0.335	0.52
SVR	0.0000142	0.0038	0.0017	0.3261	0.34
RFR	0.0000034	0.0018	0.0006	0.2164	0.84
XGB R	0.0000031	00.0018	0.0006	0.1434	0.85
LSTM	0.0000067	0.0026	0.0014	0.2126	0.67

3.5 Evaluation of Models

To evaluate housing price forecasting models, MSE, RMSE, MAE, ME, and R2 are used. The models will be implemented based on the parameters in Table 1. However, according to the test dataset that used K-Fold, the results in each subset of data were evaluated.

3.5.1 Evaluation of Models on Fold-1

The models are compared in terms of MSE, RMSE, MAE, ME, and R2 criteria on Fold-1, the results of which are given in Table 3.

The XGBR model has the lowest error and best performance in all criteria except MAE on the Fold-1 test dataset. In the MAE criteria, RFR performs better with a slight difference.

3.5.2 Evaluation of Models on Fold-2

The models are compared in terms of MSE, RMSE, MAE, ME, and R^2 criteria on the Kfold-2 test dataset, the results of which are given in Table 4.

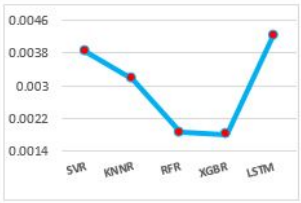

Figure 14: Comparison of models in terms of RMSE criteria on the Fold-1 test dataset

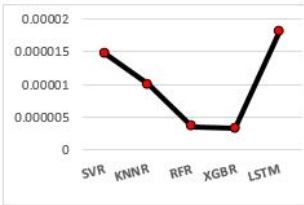

Figure 15: Comparison of models in terms of MSE criteria on the Fold-1 test dataset

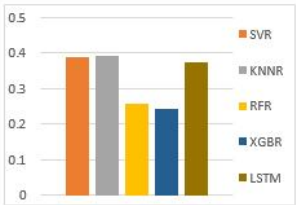

Figure 16: Comparison of models in terms of ME criteria on the Fold-1 test dataset

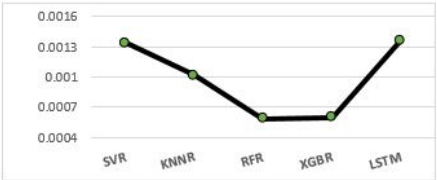

Figure 17: Comparison of models in terms of MAE criteria on the Fold-1 test dataset

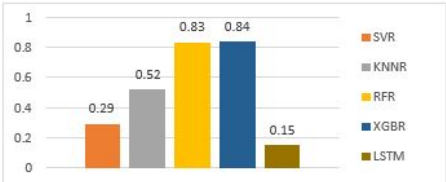

Figure 18: Comparison of models in terms of R^2 criteria on the Fold-1 test dataset

Table 5. Comparison of models on the Fold-3 test dataset.

Algorithm	MSE	RMSE	MAE	ME	R^2
KNNR	0.0000100	0.00316	0.00102	0.39	0.52
SVR	0.0000147	0.00384	0.00148	0.39	0.29
RFR	0.0000034	0.00186	0.00058	0.26	0.83
XGB R	0.0000032	0.0018	0.00059	0.242	0.84
LSTM	0.0000180	0.00424	0.00167	0.372	0.15

The XGBR model has the lowest error and best performance on all Fold-2 test databases by all criteria.

3.5.3 Evaluation of Models on Fold-3

The models in terms of MSE, RMSE, MAE, ME, and R^2 criteria are shown on the Fold-3 test dataset in Table 5.

The XGBR model has the lowest error and best performance in the MAE criterion in terms of all criteria except MAE on the Fold-3 test dataset. With a slight difference, RFR performs better.

© 2020, SRTC Iran

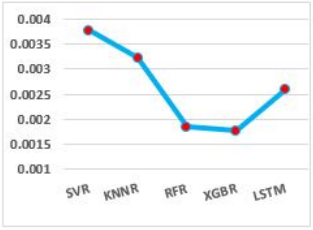

Figure 19: Comparison of models in terms of RMSE criteria on the Fold-2 test dataset

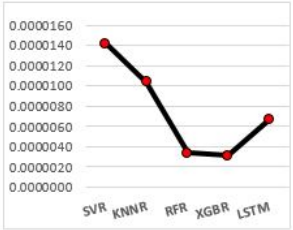

Figure 20: Comparison of models in terms of MSE criteria on the Fold-2 test dataset

Figure 21: Comparison of models in terms of ME criteria on the Fold-2 test dataset

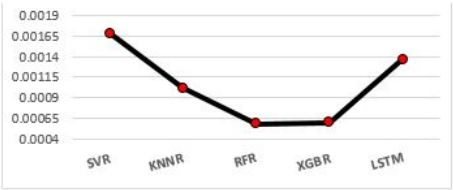

Figure 22: Comparison of models in terms of MAE criteria on the Fold-2 test dataset

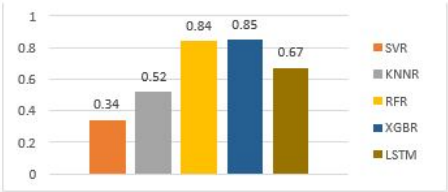

Figure 23: Comparison of models in terms of R^2 criteria on the Fold-2 test dataset

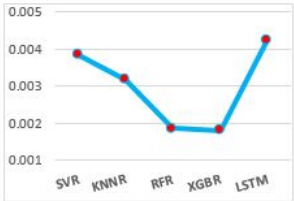

Figure 24: Comparison of models in terms of RMSE criteria on the Fold-3 test dataset

Figure 25: Comparison of models in terms of MSE criteria on the Fold-3 test dataset

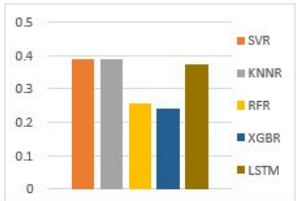

Figure 26: Comparison of models in terms of ME criteria on the Fold-3 test dataset

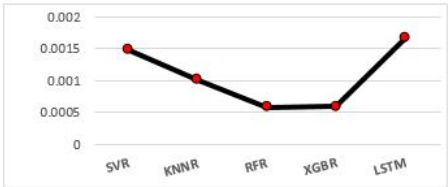

Figure 27: Comparison of models in terms of MAE criteria on the Fold-3 test dataset

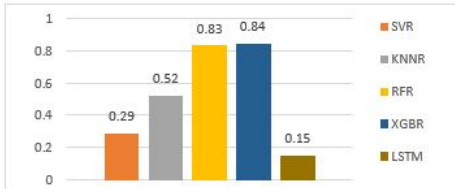

Figure 28: Comparison of models in terms of R^2 criterion on Fold-3 test dataset

Table 6. Comparison of models on the Fold-4 test dataset.

Algorithm	MSE	RMSE	MAE	ME	R^2
KNNR	0.0000146	0.00383	0.00102	0.97	0.43
SVR	0.0000176	0.0042	0.00134	0.89	0.31
RFR	0.0000058	0.00242	0.00058	0.79	0.77
XGB R	0.0000061	0.00246	0.00060	0.808	0.76
LSTM	0.0000074	0.00272	0.00121	0.709	0.71

Figure 29: Comparison of models in terms of RMSE criteria on the Fold-4 test dataset

Figure 30: Comparison of models in terms of MSE criteria on the Fold-4 test dataset

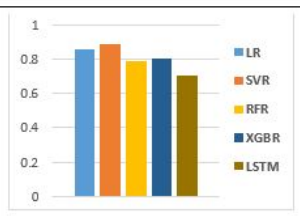

Figure 31: Comparison of models in terms of ME criteria on the Fold-4 test dataset

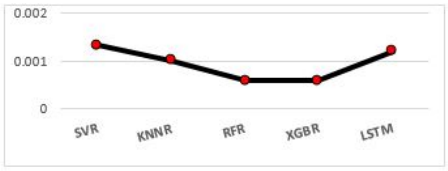

Figure 32: Comparison of models in terms of MAE criteria on the Fold-4 test dataset

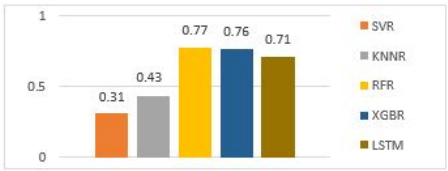

Figure 33: Comparison of models in terms of R^2 criterion on Fold-4 test dataset

3.5.4 Evaluation of Models on Fold-4

The models in terms of MSE, RMSE, MAE, ME, and R^2 criteria are shown on the Fold4 test dataset in Table 6.

The RFR model has the lowest errors and the best performance in all criteria on the Fold-4 test dataset.

3.5.5 Summary of Evaluation of Models

In the previous sections, housing price forecasting models were implemented on the entire KFold test dataset. The results of each Fold were evaluated separately. The results showed that the XGBR model was more successful than all the models in three of the test data subsets, and the RFR model

© 2020, SRTC Iran

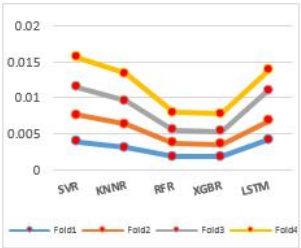

Figure 34: Comparison of models in terms of RMSE criteria on all K-Fold subsets

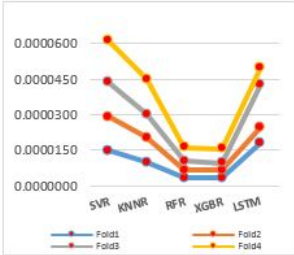

Figure 35: Comparison of models in terms of MSE criteria on all K-Fold subsets

Figure 36: Comparison of models in terms of ME criteria on all K-Fold subsets

Figure 37: Comparison of models in terms of MAE criteria on all K-Fold subsets

Figure 38: Comparison of models in terms of R^2 criteria on all K-Fold subsets

was more successful in KFold-4. Overall, the results showed that in a close competition between the severe gradient amplification regression model and the random forest regression, the Extreme Gradient Boosting regression was more successful in almost all test data subsets.

4 Discussion and Conclusion

For house price prediction in Tehran, we offered five machine learning models. The best model was determined based on the results of the criteria, which included mean square error (MSE), root mean square error (RMSE), mean absolute error (MAE), maximum error (ME), and detection coefficient (R^2). To estimate house prices, we examined Nearest Neighborhood Regression (KNNR), Support Vector Machine Regression (SVR), Random Forest Regression (RFR), Extreme Gradient Boosting Regression (XGBR), and Long Short-Term Memory Neural Network Memory Algorithm (LSTM). The findings indicate that the Extreme Gradient Boosting Regression forecasts house prices better than all other models across all assessment criteria.

Applying the findings of our research to a large number of people can help to find house price patterns. Furthermore, two ideas that come to mind are: first, determining other effective factors that influence house prices and advising that these variables be added to the Iranian real estate management information system; and second, the overall findings of this study may be used to forecast housing prices in other cities in Iran or other countries.

References

Abdel-Nasser, M., and Mahmoud, K. (2019). Accurate photovoltaic power forecasting models using deep LSTM-RNN. *Neural Computing and Applications*, **31**, 2727-2740.

Abidoye, R.B., and Chan, A.P. (2017). Modelling property values in Nigeria using artificial neural network. *Journal of Property Research*, **34**, 36-53.

Abidoye, R.B., and Chan, A.P. (2018). Improving property valuation accuracy: A comparison of hedonic pricing model and artificial neural network. *Pacific Rim Property Research Journal*, **24**, 71-83.

Ahmad, M.W., Reynolds, J., and Rezgui, Y. (2018). Predictive modelling for solar thermal energy systems: A comparison of support vector regression, random forest, extra trees and regression trees. *Journal of cleaner production*, **203**, 810-821.

Ahmad, M.S., Adnan, S.M., Zaidi, S., and Bhargava, P. (2020). A novel support vector regression (SVR) model for the prediction of splice strength of the unconfined beam specimens. *Construction and building materials*, **248**, 118475.

Bai, Y., Xie, J., Liu, C., Tao, Y., Zeng, B., and Li, C. (2021). Regression modeling for enterprise electricity consumption: A comparison of recurrent neural network and its variants. *International Journal of Electrical Power & Energy Systems*, **126**, 106612.

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

Chen, T., and Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining, (pp. 785-794).

El Hamzaoui, Y., and Perez, J.A.H. (2011, November). Application of artificial neural networks to predict the selling price in the real estate valuation process. In 2011 10th Mexican International Conference on Artificial Intelligence (pp. 175-181). IEEE.

Hamed, Y., Shafie, A.F., Mustaffa, Z., and Rusma, N. (2019). Error-reduction approach for corrosion measurements of pipeline inline inspection tools. *Measurement and Control*, **52**, 28-36.

© 2020, SRTC Iran

Hamed, Y., Alzahrani, A.I., Mustaffa, Z., Ismail, M.C., and Eng, K.K. (2020). Two steps hybrid calibration algorithm of support vector regression and K-nearest neighbors. *Alexandria Engineering Journal*, **59**, 1181-1190.

Ho, W.K., Tang, B.S., and Wong, S.W. (2021). Predicting property prices with machine learning algorithms. *Journal of Property Research*, **38**, 48-70.

Huang, G., Wu, L., Ma, X., Zhang, W., Fan, J., Yu, X., and Zhou, H. (2019). Evaluation of CatBoost method for prediction of reference evapotranspiration in humid regions. *Journal of Hydrology*, **574**, 1029-1041.

Kang, J., Lee, H.J., Jeong, S.H., Lee, H.S., and Oh, K.J. (2020). Developing a forecasting model for real estate auction prices using artificial intelligence. *Sustainability*, **12**, 2899.

Kim, M.K., Kim, Y.S., and Srebric, J. (2020). Predictions of electricity consumption in a campus building using occupant rates and weather elements with sensitivity analysis: Artificial neural network vs. linear regression. *Sustainable Cities and Society*, **62**, 102385.

Li, Y., Zou, C., Berecibar, M., Nanini-Maury, E., Chan, J.C.W., Van den Bossche, P., and Omar, N. (2018). Random forest regression for online capacity estimation of lithium-ion batteries. *Applied energy*, **232**, 197-210.

Lu, S., Li, Z., Qin, Z., Yang, X., and Goh, R. S. M. (2017). A hybrid regression technique for house prices prediction. In 2017 IEEE international conference on industrial engineering and engineering management (IEEM) (pp. 319-323). IEEE.

Ma, B., Meng, F., Yan, G., Yan, H., Chai, B., and Song, F. (2020). Diagnostic classification of cancers using extreme gradient boosting algorithm and multi-omics data. *Computers in biology and medicine*, **121**, 103761.

Madhuri, C.R., Anuradha, G., and Pujitha, M.V. (2019). House price prediction using regression techniques: a comparative study. In 2019 International conference on smart structures and systems (ICSSS) (pp. 1-5). IEEE.

Manasa, J., Gupta, R., and Narahari, N.S. (2020, March). Machine learning based predicting house prices using regression techniques. In 2020 2nd International conference on innovative mechanisms for industry applications (ICIMIA) (pp. 624-630). IEEE.

Morano, P., Tajani, F., and Carmelo, M.T. (2015). Artificial Intelligence in Property Valuation. Advances in Environmental Science and Energy Planning.

Peter, N.J., Okagbue, H.I., Obasi, E.C., and Akinola, A.O. (2020). Review on the application of artificial neural networks in real estate valuation. *International Journal of Advanced Trends in Computer Science and Engineering*, **9**, 2918 – 2925.

Punia, S., Nikolopoulos, K., Singh, S.P., Madaan, J.K., and Litsiou, K. (2020). Deep learning with long short-term memory networks and random forests for demand forecasting in multi-channel retail. *International journal of production research*, **58**, 4964-4979.

Rafiei, M.H., and Adeli, H. (2016). A novel machine learning model for estimation of sale prices of real estate units. *Journal of Construction Engineering and Management*, **142**, 04015066.

Rodriguez-Galiano, V., Sanchez-Castillo, M., Chica-Olmo, M., and Chica-Rivas, M.J.O.G.R. (2015). Machine learning predictive models for mineral prospectivity: An evaluation of neural networks, random forest, regression trees and support vector machines. *Ore Geology Reviews*, **71**, 804-818.

Varma, A., Sarma, A., Doshi, S., and Nair, R. (2018). House price prediction using machine learning and neural networks. In 2018 second international conference on inventive communication and computational technologies (ICICCT) (pp. 1936-1939). IEEE.

Wang, W.C., Chang, Y.J., and Wang, H.C. (2019). An application of the spatial autocorrelation method on the change of real estate prices in Taitung City. *ISPRS International Journal of Geo-Information*, **8(6)**,249.

Mehdi Farahzadi
Department of Statistics,
Science and Research Branch,
Islamic Azad University,
Tehran, Iran.
email: *mfarahzadi@gmail.com*

Rahman Farnoosh
Department of Mathematics,
Iran University of Science and Technology,
Tehren, Iran.
email: *rfarnoosh@iust.ac.ir*

Mohammadhasan Behzadi
Department of Statistics,
Science and Research Branch,
Islamic Azad University,
Tehran, Iran.
email: *behzadi@srbiau.ac.ir*

© 2020, SRTC Iran