Journal of Computing and Communication Vol.3 , No.1 , PP. 104-115 , 2024

HousePrice_ML: An Efficient Framework for House Price Prediction Using Soft Computing

Maged Farouk [a], Nashwa S Ragab[a], Diaa Salama*[a,b], Omnia Elrashidy [a], Mostafa Mahmoud [a], Omar Mandour[a], Hussien Walid[a], Ibrahim Hossam [a], Ali Mohamed [a], Abdelrahman Ehab [a], Reda Elazab[a]

[a] *Department of Accounting and Information Systems, Faculty of Business, Alamein International University, Alamein, Egypt*

[b] *Faculty of Computers Science, Misr International University, Cairo, Egypt*

[c] *Faculty of Computers and Artificial Intelligence, Benha University, Benha, Egypt*

*Corresponding Author: Diaa Salama [***diaa.salama@miuegypt.edu.eg***]

ARTICLE DATA

Article history:
Received 08 Jan 2024
Revised 14 Jan 2024
Accepted 02 Feb 2024
Available online

Keywords:
House price Prediction
Machine Learning
Artificial Intellgience
Linear Regression

ABSTRACT

Predicting housing prices is important to many people, such as home buyers, real estate agents, and investors. By harnessing the power of machine learning models, this paper aims to develop a highly efficient system to calculate reliable housing price forecasts. The results of this research can facilitate decision-making processes, enable more informed investments, and improve the overall buying and selling experience in the real estate market. The relationship between house prices and the economy is an important motivating factor for predicting house prices. This paper focuses on how to predict housing prices using machine learning techniques. This paper proposes an efficient framework For prediction houses using six machine learning algorithms (SVM, Tree, Neural Network, KNN, Linear Regression, Gradient Boosting). In best model 1, the number of fields equals Gradient Boosting; in best model 2, the number of fields equals Linear Regression; in Best model 3 number of fields equals Gradient Boosting. The best all model equal model 2 equal Linear Regression.

1. Introduction

Predicting house prices is a task for many individuals and organizations. Knowing the price makes home buyers think about a better plan to buy the right home at the right price. For home sellers, it is helpful to understand the home's asking price to price it in the market [1]. Organizations can also benefit from a home price forecasting model. For example, companies can use the expected value of a home's price to adjust their policies to help people facing economic hardship achieve their homeownership goal.

Predicting house prices is a difficult problem because real estate valuations depend on the physical characteristics of the building, its location, and people's perceptions of these factors. As is the case in any market, the price is driven by the willingness to pay of the buyer, which makes determining the price for a residential property difficult. Because of these new circumstances, traditional data processing and analytical tools may be unable to capture, process, and analyze highly complex information in the social and economic worlds. New techniques have been developed to treat the colossal amount of available data [2].

Machine learning (ML) has emerged, making it easier for everyone to predict house prices. Using advanced statistical techniques, we analyze data to accurately predict home prices and reveal complex information that humans can use manually, thus enabling real estate professionals to make informed decisions.

This article begins a comprehensive analysis of predicting house prices using machine learning. From the finer details of data collection and processing to the complex differences in model training and deployment [3].

Machine learning models have become highly beneficial for predicting house prices aiding buyers and sellers in the real estate market. However, these models are not always perfect and can present various.

The problem is to build a prediction model to estimate the sales price of a house based on a dataset containing information about the house, such as number of bedrooms, area, location and amenities. Typically, the dataset is divided into training and test datasets, where the model is trained on the training set, and its predictive performance is evaluated on the test set [4-5].

The main contribution of this research is that predicting house prices helps us choose what we need and facilitates knowing the prediction for each house at the appropriate price. Machine learning is also one of the most important branches of artificial intelligence, and despite its importance, it is incorrect in most matters.

The rest of the paper can be organized as follows: in Section 2, we will talk about Related Works; in Section 3, we will talk about the Methodology; and in Section 4, we will talk about the Result and Analysis.

2. Related Work

Predicting real estate prices is a challenging task that many researchers have tackled. Since accurate house prices allow better information to all parties in the housing market and improve housing policy and property valuation, a comprehensive overview of strategies for predicting house prices is valuable to both research and society.

In [6], Authors Applying machine learning methods to predict house prices, the results show that the best model 1 is Gradient boosting: Many banks and lending institutions are required to assess the value of residential properties in their mortgage portfolios. Therefore, recently, machine learning methods such as random forests and boosted trees have been increasingly used. We investigate whether introducing a loss function closer to the actual loss measure of the AVM can improve the performance of machine learning methods (especially the Gradient boosting tree method).

In [7] Authors Applying machine learning methods to predict house prices, the results show that the best model 2 is Linear Regression. In particular, linear regression is a useful tool for predicting quantitative responses. It has been around for a long time and is the subject of countless textbooks.

In [8], Authors Applying machine learning methods to predict house prices show that the best model 1 is Neural networks, composed of basic units analogous to neurons. These units are linked by connections whose strength is adjustable due to a learning process or algorithm. Each of these units integrates independently (in parallel) the information provided by its synapses to evaluate its activation state.

In [9] Authors describe when we tried all models, it became clear that Gradient boosting, Linear Regression, and Neural network are the best three models for predicting house prices. Researchers face two main challenges: The biggest challenge is the optimal number of features to predict direction accurately. Identify several features that help accurately predict the trend of house prices. Determine the optimal number of features that help accurately predict the trend of house prices and productivity growth in various housing construction sectors. It is a good indicator that productivity growth in various residential construction sectors will positively impact house prices. Impact on house price growth: Shows how apparent prices affect house prices. Home prices look fashionable at first glance, and then over a long period, they will collapse and experience a long-term decline.

We rely on attention mechanisms to determine the main factors that affect housing prices. Attention mechanisms are widely used. In many fields, due to its ability to distinguish characteristics,

Such as automatic translations, image titles and voice. Knowing these mechanisms can improve important parts of a system that assigns greater weights to the most influential and can screen out irrelevant parts by assigning lower weights to less relevant features of the model.

Feature engineering plays an important role in improving model performance. The temporal aspects of the housing market are considered using time series analysis to account for trends, seasonality, and economic fluctuations that affect home prices over time.

Some studies focus on incorporating geospatial information, considering the impact of a property's location, neighborhood characteristics, and proximity to amenities on its value.
Transfer learning methods that improve pre-trained models in tasks relevant to home price forecasting have been studied for their ability to leverage knowledge from diverse datasets. Handling missing data, dealing with outliers, and normalizing or scaling features are important preprocessing steps investigated in related research.

Researchers often use measures such as mean squared error (MSE) and mean absolute error (MAE), which were not common, and R-squared and MAPE (which were the best means) to evaluate the performance of house price forecasting models.

3. Methodology

Numerous algorithms were used, and each algorithm was studied. A study was done on each algorithm before training the model with them on the data set. Used in the data set. The following figure shows the steps the dataset took to obtain the results. Methods to achieve these goals and objectives are discussed. The reader is familiar with the specific techniques used in this study and is supported by the literature discussed in the previous articles [10].

A-Dataset Description

The first dataset consists of 6 features. The dataset was divided into 80% for training and 20% for testing. Detailed descriptions of Features can be found below.
Housing price prediction with this synthetic dataset. Perfect for data science enthusiasts, machine learning practitioners, and this dataset offers a diverse collection of features, including square footage, bedrooms, bathrooms, neighborhood types, and the year of construction. Immerse yourself in the challenge of predicting house prices and enhance your skills in regression analysis. SquareFeet of the house. Amt of bedrooms. Amt of bathrooms/restroom/washroom. Area neighborhood where the house is. Which year it was built. The price of the home.

TABLE 1
FEATURE OF Housing Price Prediction Data 1

Feature	Type	Values
SquareFoot	Numerical	From 1000 To 2999
Bedrooms	Numerical	From 2 To 5
Bathrooms	Numerical	From 1 To 3
Neighborhood	Classification	Rural or Suburb
YearBuilt	Numerical	From 1950 To 2021
Price	Numerical	From -36.6k To 492k

The Second dataset consists of 3 features. The dataset was divided into 80% for training and 20% for testing. Detailed descriptions of Features can be found below. The dataset is about predicting the house price based on area and no rooms. Price: The price of the house. Area: The total area of the house in square feet.

TABLE 2
FEATURE OF Simple House Price Prediction 2

Feature	Type	Values
Area	Numerical	From 852 To 4478
Rooms	Numerical	From 1 To 5
Price	Numerical	From 170k To 700k

The Third and final dataset consists of 10 features. The dataset was divided into 80% for training and 20% for testing. Detailed descriptions of Features can be found below. This dataset provides key features for predicting house prices, including area, bedrooms, bathrooms, stories, amenities like air conditioning and parking, and information on furnishing status. It enables analysis and modeling to understand the factors impacting house prices and develop accurate predictions in real estate markets. Price: The price of the house. Area: The total area of the house in square feet. Bedrooms: The number of bedrooms in the house. Bathrooms: The number of bathrooms in the house. Stories: The number of stories in the house. Mainroad: Whether the house is connected to the main road (Yes/No). Guestroom: Whether the house has a guest room (Yes/No). Basement: Whether the house has a basement (Yes/No). Hot water heating: Whether the house has a hot water heating system (Yes/No). Airconditioning: Whether the house has an air conditioning system (Yes/No). Parking: The number of parking spaces available within the house. Prefarea: Whether the house is in a preferred area (Yes/No). Furnishing Status: The furnishing status of the house (Fully Furnished, Semi-Furnished, Unfurnished).

TABLE 3
FEATURE OF Housing Price Prediction 3

Feature	Type	Values
Price	Numerical	From 1750000 to 1330000
bedroom	Numerical	From 1 to4
bathroom	Numerical	From 1 to4
area	Numerical	From 1700 to 16200
stories	Numerical	From 1to4
main road	Classification	Yes or no
guestroom	Classification	Yes or no
basement	Classification	Yes or no
Hot water heating	Classification	Yes or no
Air conditioning	Classification	Yes or no

B- Used Algorithms

The above datasets were passed into nine different machine-learning algorithms. The machine learning algorithms passed to the machine learning algorithms were Gradient Boosting, K Nearest Neighbor (kNN), Random Forest, and Decision Tree. For each algorithm, statistics were generated. They are accuracy, recall, precision, and specificity. The results were then plotted and compared. Results, graphs, and discussion of results are discussed [11].

1) Gradient Boosting

Machines are a powerful family of machine-learning techniques successfully used in a wide range of practical applications. They are highly customizable to the application's specific needs, such as learning with respect to different loss functions [12].

$$F(x) = \sum_{m=1}^{M} Ym\ hm\ (x)$$

F(x) is the final reinforcement model, M is the number of weak learners (trees), Ym is the contribution of the mth weak learner, and hm (x) is the prediction of the mth weak↓ learner according to input.

2) Decision Tree

A decision tree is a tree whose internal nodes can be considered tests (for input data patterns) and whose leaf nodes can be considered categories (of these patterns). These tests are filtered through the Tree to obtain the correct output for the input patterns. Decision tree algorithms are applied and used in various fields [13].

$$R(t) = \mathrm{var}(D_t) - \Sigma_{i=1}^{2} = \frac{|\ Dt, i\ |}{|\ Dt\ |} \cdot Var(Dt, i)$$

Dt is the dataset of node t, |Dt| is the number of samples of Dt, and i is the subset Dt of the i-th child node. Partition rule Feature < Threshold, 'Feature' - the feature on which the decision is made, 'Threshold' - the value used to split the data based on the selected feature.

3) Random Forest

Is algorithm This method combines multiple random decision trees and aggregates their predictions by averaging them, showing excellent performance in environments where the number of variables is much larger than the number of observations. In addition, it is general enough to be applied to large problems, can be easily adapted to a variety of ad hoc learning tasks, and provides a variety of important metrics [14].

$$Random\ Forest\ (x) = \frac{1}{N} \sum_{i=1}^{N} Tree_i\ (x)$$

RandomForest(x) - final prediction of the random forest for input x, N - number of trees in the forest, Tree; (x) - prediction of the i-th decision tree for input x.

4) Linear Regression

Linear regression is a modeling technique used to analyze data to make predictions. Simple regression creates a bivariate model that predicts a response variable (Ll_1D466) based on an explanatory variable (Ll_1D466) 1. Multiple regression extends the model to include multiple explanatory variables (x1, x2, ..., xp). , XP), thereby creating a multivariate model [15].

$$y = mx + b$$

y is the dependent variable (what you are trying to predict), x is the independent variable (the raw data),m-slope of the line (rate of change), and b-y-intercept (the value of yy when xx is 0).

5) SVM

(Support Vector Machines) have become an increasingly popular tool in machine learning tasks, including classification, regression, and novelty detection. In particular, SVMs have shown good generalization performance for realistic problems, and Theory appropriately motivates their approach [16].

$$f(x) = Sign(W.X + b)$$

f(x) - decision function, w - vector of weights, x - vector of input features, b - bias term.

6) Stochastic Gradient Descent

For optimizing expected value performance measures in discrete event systems. The algorithm increases accuracy insuccessive iterations and moves toward the generalized Gradient of the computed sample performance function [17].

$$\theta_{t+1} = \theta_t - \eta \nabla J\left(\theta_t; x^{(i)}, y^{(i)}\right)$$

θ_t is the vector of parameters at iteration t, and η is the learning rate,

$\nabla J\left(\theta_t; x^{(i)}, y^{(i)}\right)$ Is the Gradient of the cost function J for the parameters? θ_t the i-th training.

7) KNN

The algorithm is a widely applied method for classification in machine learning and pattern recognition. However, the KNN algorithm is computationally expensive and cannot perform satisfactorily in many applications [18].

$$\hat{y}(x) = majority\ vote\ \left(\left\{y^{(j)} | x^{(j)} is\ oe\ of\ the\ k\ nearest\ neighbors\ of\ x\right\}\right)$$

$\hat{y}\ (x)$- prediction class for input x, k - number of nearest neighbors considered. $x^{(j)}$ and $y^{(j)}$- features and classes in training.

8) AdaBoost

It is one of the best-boosting algorithms available. With a solid theoretical foundation and great success in practical applications, AdaBoost can boost a weak learning algorithm with slightly better accuracy than a random guess to a strong learning algorithm of arbitrary accuracy, bringing new methods, new design, brought a new philosophy to the design of learning algorithms [19].

$$z_t = \sum_{i=1}^{N} w_i^{(t)}.\exp\left(-\alpha_t \cdot y^i \cdot h^t\left(x^i\right)\right)$$

x^{i}- vector of features of the i-th training instance.
y^{i} - true label of the i-th training instance (+1 or -1 for binary classification).
$h^t\left(x^{\mathrm{i}}\right)$ - prediction of the weak classifier at iteration t for the i-th instance.
α_t - the weight assigned to the weak classifier at iteration t.
$w_i^{(t)}$ - is the weight for the i-th instance at iteration t.
z_t - is the normalization factor so that the sum of the weights to 1.

9) Constant

A "constant algorithm" may refer to a process that is not a specific algorithm but has a constant time complexity. In algorithmic analysis, time complexity measures the amount of time an algorithm takes to complete as a function of the magnitude of its inputs.

$$Pi\ (\boldsymbol{\pi}):$$

$\boldsymbol{\pi} \approx \mathbf{3.14159}9$

Euler^'s number e:

$$\boldsymbol{e} \approx \mathbf{2.}71828e$$

If you are looking for a general expression for a mathematical constant, one of the most famous constants is Pi, the ratio of the circumference of a circle to its diameter, an irrational number approximately equal to 3.14159.
The mathematical constant e (Euler's number) is another important constant equal to approximately 2.71828. It is the base of the natural logarithm and is often found in various mathematical and scientific contexts.

4. Results and Analysis

Table 4
Statistics For Algorithm That Splits Data Number Of Folds: 10

Model	MAPE	R2
Tree	0.346	0.246
SVM	0.429	-0.176
Stochastic Gradient Descent	0.283	0.554
Random Forest	0.299	0.494
Neural Network	0.288	0.558
Linear Regression	0.282	0.570
KNN	0.300	0.478
Gradient Boosting	0.281	0.569
Constant	0.472	-0.000
AdaBoost	0.309	0.462

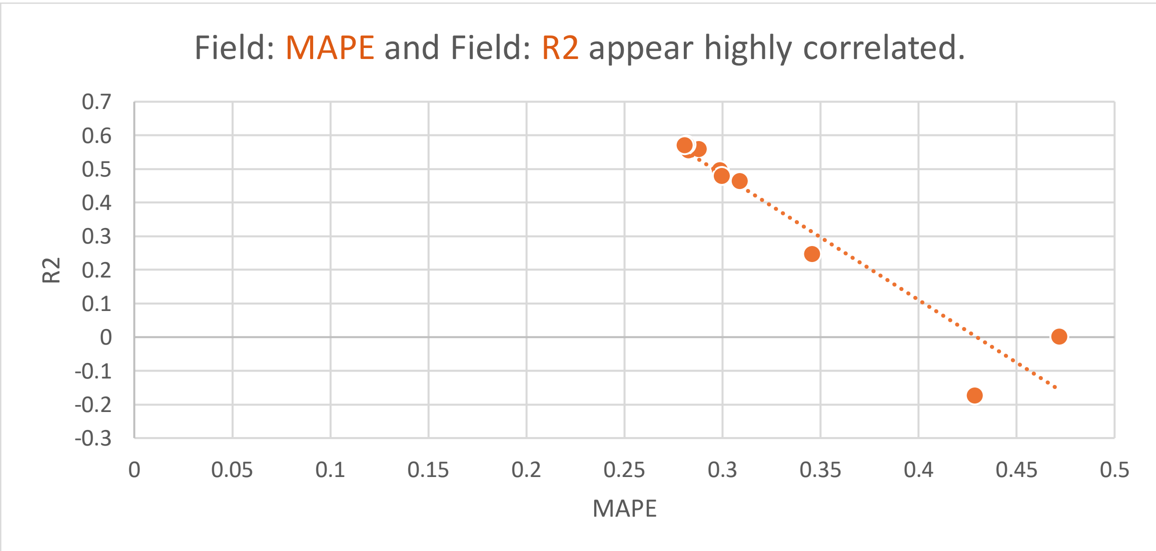

Fig 1 . Performance graph of the first dataset with 10 folds repetition

Svm and Constant are dominant

Constant had the highest MAPE with an accuracy of 0.472. At the same time, SVM was very close with an accuracy of 0.429 while in R2 with an accuracy of -0.176 and next to it was Constant with the lowest accuracy of -0.000. In contrast, the lowest MAPE was Linear Regression with an accuracy of 0.282, and the highest accuracy was Linear Regression in R2 with an accuracy of 0.570.

Table 5
Statistics For Algorithm That Splits 80/20.

Model	MAPE	R2
Tree	0.333	0.251
SVM	0.404	-0.191
Stochastic Gradient Descent	0.270	0.555
Random Forest	0.283	0.498
Neural Network	0.260	0.541
Linear Regression	0.268	0.573
KNN	0.286	0.483
Gradient Boosting	0.275	0.569
Constant	0.445	-0.000
AdaBoost	0.288	0.469

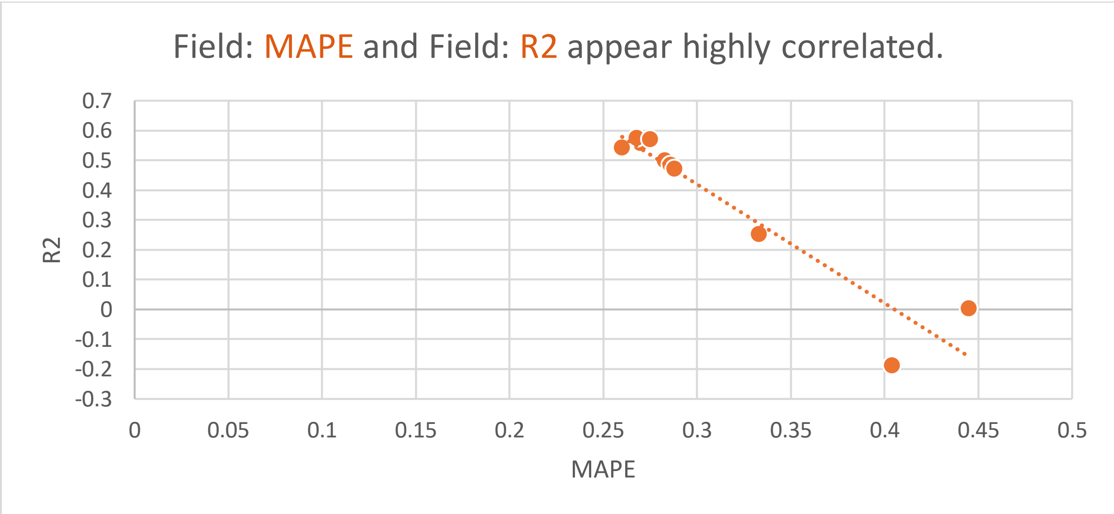

Fig 2.Performance of the graph for the first split dataset

Svm and Constant are dominant.
Constant had the highest MAPE with an accuracy of 0.445. At the same time, SVM was very close with an accuracy of 0.404, while in R2, it had an accuracy of -0.191 next to it; Constant had the lowest accuracy of -0.000 while the lowest MAPE was the neural network. With an accuracy of 0.260, the highest accuracy is the linear regression in R2 with an accuracy of 0.569

Table 6
Statistics For Algorithm That Splits Data Number Of Folds: 10

Model	MAPE	R2
KNN	0.196	0.592
Linear Regression	0.170	0.684
Neural Network	1.000	-7.572
Random Forest	0.206	0.551
SVM	0.260	-0.108

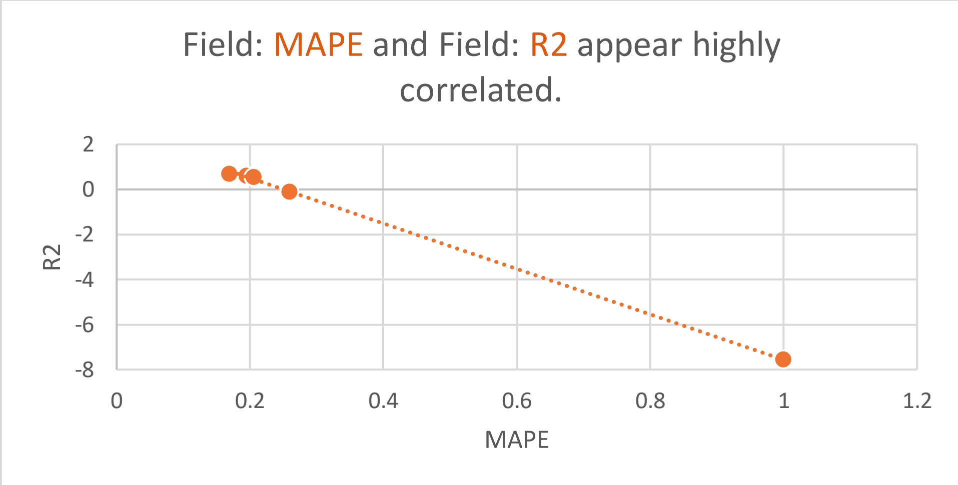

Fig 3. Performance graph of the second dataset with ten folds repetition

Svm and neural networks are dominant.
The neural network had the highest MAPE with an accuracy of 1.000, while Svm was far behind with an accuracy of 0.260. In contrast, in R2, it had an accuracy of -0.108; next to it, Linear Regression had the highest accuracy of 0.684 while it had the lowest accuracy. MAPE has an accuracy of 0.196, KNN has the accuracy of MAPE with an accuracy of 0.592, and the lowest accuracy in the R2 Neural network with an accuracy of -7.572.

Table 7
Statistics For Algorithm That Splits 80/20.

Model	MAPE	R2
KNN	0.192	0.550
Linear Regression	0.176	0.655
Neural Network	1.000	-7.464
Random Forest	0.187	0.573
SVM	0.256	-0.137

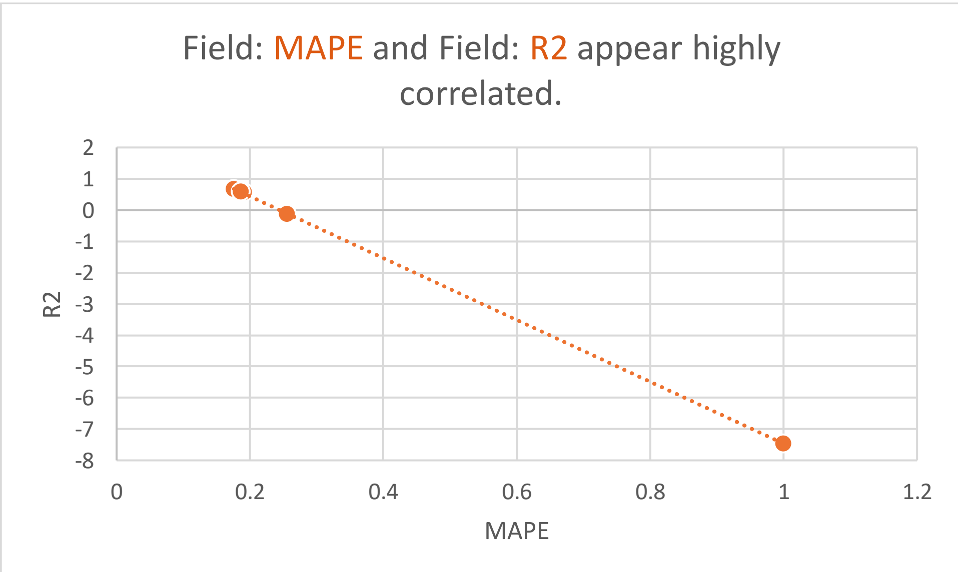

Fig 4. Performance of the graph for the second split dataset

Svm and neural networks are dominant.
The neural network had the highest MAPE with an accuracy of 1.000, while Svm was far behind with an accuracy of 0.256, while R2 had an accuracy of -0.137. Next, Linear Regression had the highest accuracy of 0.655 and the lowest accuracy. MAPE has an accuracy of 0.176, KNN has the accuracy of MAPE with an accuracy of 0.569, and the lowest accuracy in the R2 Neural network with an accuracy of -7.464.

Table 7
Statistics For Algorithm That Splits Data Number Of Folds: 10

Model	Mape	R2
Constant	0.341	-0.009
KNN	0.249	0.327
Tree	0.239	0.298
Random forest	0.192	0.573
Gradient boosting	0.174	0.652
SVM	0.313	-0.030
Linear regression	0.177	0.662
Adaboost	0.178	0.593
Neural network	1.000	-6.506
Stochastic gradient descent	0.191	0.606

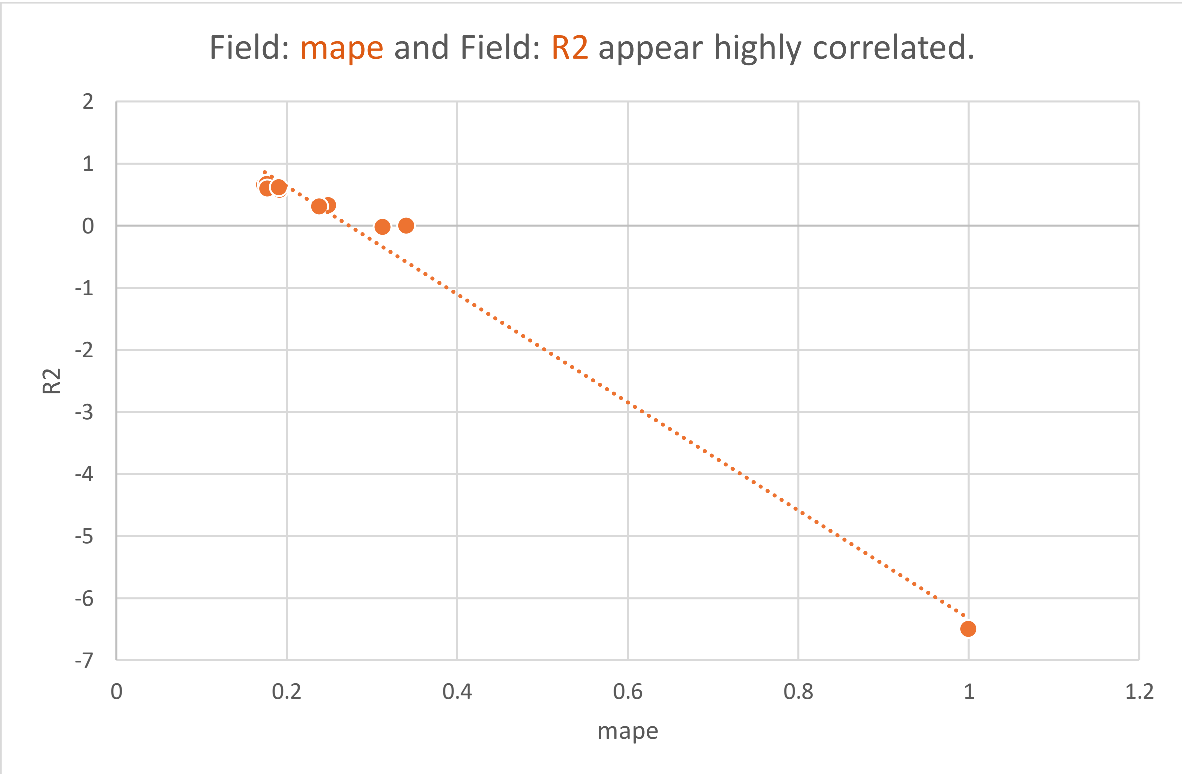

Fig 5. Performance graph of the third dataset with ten folds repetition

Constant and neural networks are dominant.
The neural network had the highest MAPE with an accuracy of 1.000 while Constant was far behind with an accuracy of 0.341 while in R2, it had an accuracy of -0.009. Next to it, Linear Regression had the highest accuracy of 0.662 while it had the lowest accuracy. MAPE has an accuracy of 0.177, SVM has the

accuracy of in MAPE with an accuracy of 0.313, and the lowest accuracy in the R2 Neural network with an accuracy of -6.506 .

Table 8
Statistics For Algorithm That Splits 80 \ 20.

Model	Mape	R2
Constant	0.343	-0.005
KNN	0.252	0.324
Tree	0.218	0.441
Random forest	0.183	0.600
Gradient boosting	0.175	0.636
SVM	0.311	-0.039
Linear regression	0.175	0.667
Adaboost	0.176	0.588
Neural network	1.000	-6.203
Stochastic gradient descent	0.185	0.643

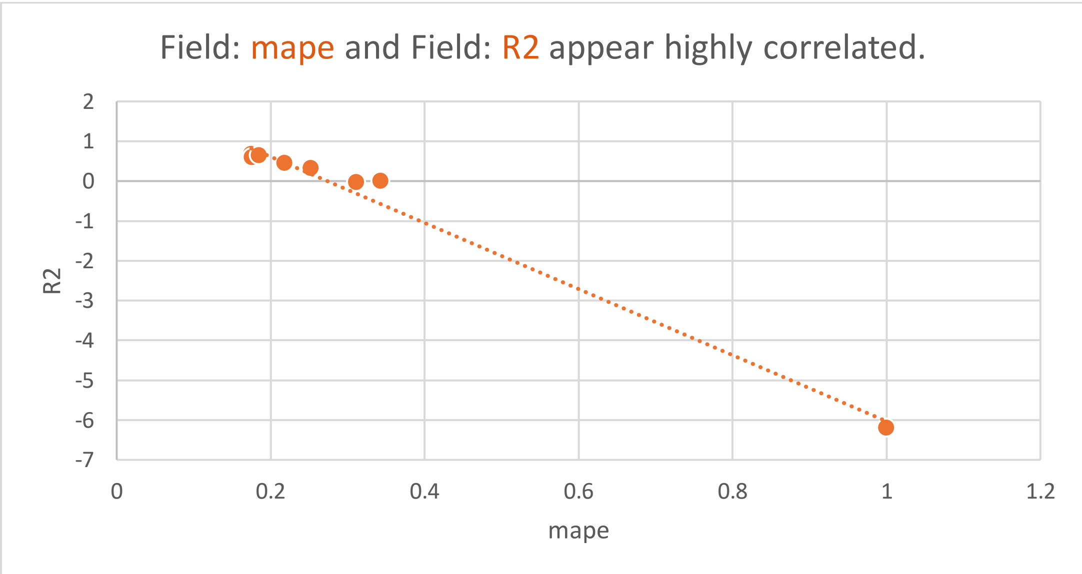

Fig 6. Performance of the graph for the third split dataset

Constant and neural networks are dominant.
The neural network had the highest MAPE with an accuracy of 1.000, while Constant was far behind with an accuracy of 0.343. In contrast, in R2, it had an accuracy of -0.005; next to it, Linear Regression had the highest accuracy of 0.667 while it had the lowest accuracy. MAPE has an accuracy of 0.175, SVM has the accuracy of MAPE with an accuracy of 0.311, and the lowest accuracy in the R2 Neural network with an accuracy of -6.203

5. Conclusion

Since house price forecasting involves various factors and can be influenced by economic, social, and local market conditions, machine learning models using factors such as location, size, number of bedrooms, and many other property-related characteristics are widely used to predict house prices. In short, house price forecasting is a challenging task requiring considering many factors carefully. The success of a forecasting model depends on the quality of the data, the relevance of the features, the selection of appropriate algorithms, and the ongoing maintenance of the model to adapt to changing market conditions. Continuous improvement and validation of the model with real data is essential to ensure its accuracy and validity over time.

References

[1] Zhang, H., Li, Y., & Branco, P. (2023). Describe the house and I will tell you the price: House price prediction with textual description data. Natural Language Engineering, 1-35.

[2] Geerts, M., & De Weerdt, J. (2023). A Survey of Methods and Input Data Types for House Price Prediction. ISPRS International Journal of Geo-Information, 12(5), 200

[3] Murgia, A., Concas, G., Marchesi, M., & Tonelli, R. (2010, September). A machine learning approach for text categorization of fixing-issue commits on CVS. In Proceedings of the 2010 ACM-IEEE International Symposium on Empirical Software Engineering and Measurement (pp. 1-10)

[4] Tang, W., Li, Z., & Cassar, N. (2019). Machine learning estimates of global marine nitrogen fixation. Journal of Geophysical Research: Biogeosciences, 124(3), 717-730.

[5] Dutta, S., Shi, A., & Misailovic, S. (2021, August). Flex: fixing flaky tests in machine learning projects by updating assertion bounds. In Proceedings of the 29th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering (pp. 603-614).

[6] Hjort, A., Pensar, J., Scheel, I., & Sommervoll, D. E. (2022). House price prediction with Gradient boosted trees under different loss functions. Journal of Property Research, 39(4), 338-364.

[7] Tranmer, M., & Elliot, M. (2008). Multiple linear regression. The Cathie Marsh Centre for Census and Survey Research (CCSR), 5(5), 1-5.

[8] Abdi, H. (1994). A neural network primer. Journal of Biological Systems, 2(03), 247-281.

[9] Tellis, W. (1997). Application of a case study methodology. The qualitative report, 3(3), 1-19.

[10] Moschovakis, Y. N. (2001). What is an algorithm?. Mathematics unlimited—2001 and beyond, 919-936.

[11] Natekin, A., & Knoll, A. (2013). Gradient boosting machines, a tutorial. Frontiers in neurorobotics, 7, 21.

[12] Navada, A., Ansari, A. N., Patil, S., & Sonkamble, B. A. (2011, June). Overview of use of decision tree algorithms in machine learning. In 2011 IEEE control and system graduate research colloquium (pp. 37-42). IEEE.

[13] Li, X. (2013). Using" random forest" for classification and regression. Chinese Journal of Applied Entomology, 50(4), 1190-1197.

[14] Krämer, W., & Sonnberger, H. (2012). The linear regression model under test. Springer Science & Business Media.

[15] Jakkula, V. (2006). Tutorial on support vector machine (svm). School of EECS, Washington State University, 37(2.5), 3.

[16] Konečný, J., & Richtárik, P. (2017). Semi-stochastic gradient descent methods. Frontiers in Applied Mathematics and Statistics, 3, 9.

[17] Kuang, Q., & Zhao, L. (2009). A practical GPU based kNN algorithm. In Proceedings. The 2009 International Symposium on Computer Science and Computational Technology (ISCSCI 2009) (p. 151). Academy Publisher.

[18] Ying, C., Qi-Guang, M., Jia-Chen, L., & Lin, G. (2013). Advance and prospects of AdaBoost algorithm. Acta Automatica Sinica, 39(6), 745-758.

[19] Manasse, M., McGeoch, L., & Sleator, D. (1988, January). Competitive algorithms for online problems. In Proceedings of the twentieth annual ACM symposium on Theory of computing (pp. 322-333).

[20] Banerjee, D., & Dutta, S. (2017, September). Predicting the housing price direction using machine learning techniques. In 2017 IEEE international conference on power, control, signals and instrumentation engineering (ICPCSI) (pp. 2998-3000). IEEE.

[21] Wang, P. Y., Chen, C. T., Su, J. W., Wang, T. Y., & Huang, S. H. (2021). Deep learning model for house price prediction using heterogeneous data analysis along with joint self-attention mechanism. IEEE Access, 9, 55244-55259.