International Journal of Advances in Engineering and Management (IJAEM)
Volume 6, Issue 07 July 2024, pp: 195-200 www.ijaem.net ISSN: 2395-5252

House Price Pridiction UsingXG-BOOST

Swarnit Soni, Vipin Janghela
Computer Science, Bachelor of Technology, Jabalpur, India.
Computer Science, Bachelor of Technology, Jabalpur,India.

Date of Submission: 01-07-2024 Date of Acceptance: 10-07-2024

ABSTRACT—A vital prerequisite for many industries, including real estate and mortgage finance, is the ability to accurately forecast house prices. It is common knowledge that a property's value is greatly impacted by its surrounding neighborhood in addition to its physical characteristics. Real estate developers' first priority is to balance budgetary limits with the different housing needs of individuals.

Keywords—feature engineering; feature importance; house price prediction; hyperparameter tuning; machine learning; regression modeling; XGBoost.

I. INTRODUCTION

One of the essential needs of humans is housing. Because the real estate industry contributes significantly to the global economy, house price forecast is crucial for real estate and mortgage lending companies. This procedure helps reduce risks and close the gap between supply and demand, which benefits both firms and consumers [1] .

Regression approaches, which use multiple variables to form models, are frequently used to estimate property prices. For many different stakeholders, an effective and user-friendly housing price prediction model offers several advantages. The model can be used by real estate companies to evaluate risks and choose wisely when making investments.It can be used by mortgage lending companies to assess loan applications and choose suitable interest rates.

The methodology helps buyers determine how affordable a property is and helps them make well-informed judgments about what to buy. Above all, the volatility of home prices in recent times has increased the demand for prediction models.

II. RELATED WORK

Forecasting home values aids in the creation of successful policies for long-term housing markets, directs investment choices, and offers insights into economic trends. In order to make well-informed investment decisions, portfolio managers and real estate investors frequently rely on house price forecasts, as highlighted by the study by [2].

There isa direct correlation between the precision of these forecasts and enhanced portfolio optimization, as evidenced by current market movements.Expecting price swings in real estate allows investors to takeadvantage ofnewpossibilities, proactively adjust their portfolios, and effectivelymanage risks, all of which contribute to more stable and successful investment results.

Additionally, the writers oftalked about how people might learn more about real estate to help them make better financial and investment decisions for themselves. In a similar vein, Ref. highlighted that policymakers and financial institutions view movements in home prices as an economic indicator because they can have an impact on borrowing, consumer spending, and the state of the economy as a whole. According to the study by, the amount that people could borrow from financial institutions drove the demand for homes, which led to an intuitive theoretical model of house prices.

III. METHODOLOGY

The techniques used to predict house prices are presented in this section. In order to determine which regression model performed best in terms of interpretation, we tested a number of models, including support vector regressor (SVR), multilayer perceptron (MLP), random forest regression (RF), linear regression (LR), and extreme gradient boosting (XGBoost).

The housing data utilized in this study is available to the publicat the following link: https://www.kaggle.com/datasets/shashanknecrothapa/ames-housing-dataset (accessed on November 2, 2023).

A. Linear Regression

Linear regression (LR) is a common and easy method for predicting home prices. A statistical method called regression analysis (LR) is used to determine the connection between a dependent variable (Y) and one or more independent variables (Xi). The equation

$$Y = \beta 0 + \sum k\ i\ \beta iXi + \varepsilon (1)$$

represents this relationship, with β0 serving as the intercept, βi serving as the slopes, Xi serving as the independent variables, and ε serving as the error term.

International Journal of Advances in Engineering and Management (IJAEM)
Volume 6, Issue 07 July 2024, pp: 195-200 www.ijaem.net ISSN: 2395-5252

B. Random Forest

The resilient and adaptable ensemble learning method known as random forest (RF), first presented by [27], produces accurate results for a wide range of datasets [14,15,28,29]. RF is well known for its accuracy in forecasting results across a range of datasets. It also performs exceptionally well when managing high-dimensional data, identifying intricate correlations, andreducing overfitting.

The j th tree estimate is thus formulated as:
mn x; Øj , ∂n = ∑i∈∂n(Øj) 1 Xi∈An(x;Øj ,∂n) Y1 Nn x; Øj , ∂n (2)

where ∂ ∗ n Øj is the set of selected data points before tree construction. An x; Øj , ∂n Y1 is the cell containing x and Nn x; Øj , ∂n is the number of points selected before tree construction that fall into An x; Øj , ∂n Y1 . The finite forest estimate as a result of the combination of trees is then represented as: mM,n(x; Ø1 . . . . ..Øm, ∂n) = 1 M ∑ m j=1 mn x; Øj , ∂n (3) where M can take any size but is limited to computing r.

C. Support Vector Machine

In the field of data mining and machine learning, the support vector machine (SVM) is widely recognized and highly esteemed for its capacity to manage intricate data patterns and accomplish high-dimensional classification jobs with exceptional precision. Vapnik introduced SVM in the 1990s, and it has subsequently shown to be useful in machine learning applications. SVMs are adaptable and have good performance in regression as well as classification[3].

SVM creates a hyperplane between the split marginal lines of the closest support vectors (input vectors) in a classification problem. Maximizing the marginal distance within the space yields the ideal separating hyperplane. For the best prediction, the greatest margin hyperplane is employed. The result is more generally applicable the higher the marginal distance. Using the kernel function, SVM is a linear classifier that can also handle non-linear classification tasks. Because the kernel method is used, SVM provides accurate prediction and is less prone to overfitting [14, 32]. SVMs use the kernel trick to transform the data and find the best decision boundary between possible outputs.

In the same way, a class of SVM for regression is the support vector regressor (SVR) for a regression problem. The goal of SVR is to offer a function f(x) estimation.

SVR fits the regression line in the E-insensitive zone in order to achieve this. For better out-of-sample performance, an E-insensitive loss function is introduced. SVR attempts to solve the following optimization problem (where e ∗ i denotes slack variables, which are the errors at the inbound and outbound of the E-insensitive region, respectively) in order to obtain the regression function
f(x). Reduce:

1/2 || kw2 || + C∑ n i=1 (ei + e ∗ i) (4)

Topic to:
yi – b − w ′xi ≤e + ei
b + w′xi − yi ≤e + e ∗ i
ei ≥ 0, e∗i ≥ 0

D. Multi-Layer Perceptron (MLP)

As an essential component of artificial neural networks (ANNs), the multi-layer perceptron (MLP) is a potent tool for resolving challenging issues across a range of industries. Deep belief network (DBN), self-organizing map (SOM), and MLP algorithm models are very useful when human experts are not available or cannot sufficiently explain the decisions made with their knowledge, when problem solutions grow in scope, or when solutions need to be adjusted in response to new information[4] . The authors of emphasized that MLP has the ability to learn both linear and non-linear functions. With the training and early experience data, MLP can learn how to do tasks. Minimizes the loss functionusing MLP. A stochastic program is MLP. The mathematical representation of the layers in a fully connected network is provided by Equations (5)–(7), where the input units are designated as xj, the activation functions are Ø(1) and Ø(2), the weights are represented by ω, the units in the lth hidden layer are marked as hi (l), and the output y
hi (1) =Ø(1)∑j ω (1) ij xj + b (1) i ~ (5)

hi (2) = Ø (2) , ∑j ω (2) ij hj (1) + b (2) i , (6)

yi = Ø (3) ∑j ω (3) ij hj (2) + b (3) i (7)

E. Extreme Gradient Boosting

Extreme gradient boosting, or XGBoost, has become incredibly important in the machine learning area because of its remarkable performance and adaptability. The authors of presented XGBoost as a scalable and effective gradient tree boosting implementation, allowing real-world problems to be solved with the least amount of resources possible.Because XGBoost is open-source, a collaborative community has been encouraged, which

International Journal of Advances in Engineering and Management (IJAEM)
Volume 6, Issue 07 July 2024, pp: 195-200 www.ijaem.net ISSN: 2395-5252

has resulted in ongoing advancements and improvements[5].The algorithm's popularity is demonstrated by its performance in a variety of fields, such as supply chains, healthcare, and finance. In mathematical terms, XGBoost can be expressed as an optimization problem using the following formulation for the objective function.

$$Obj = \sum_{i=1}^{n} L(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k)$$

where (f) is the regularization term that controls the model's complexity and keeps it from overfitting, L (yi, yˆi) is the loss function, and K is the number of trees.

$$\Omega(f) = \gamma T + 1/2\, \lambda \sum_{j=1}^{T} \omega^2_j \quad (9)$$

in where T is the number of leaves and ω is the leaf node's score. After the tth iteration, the expected value of the ith sample is:

$$\hat{y}_i(t) = \hat{y}_i^{(t-1)} + f_t(X_i) \quad (10)$$

Furthermore,

$$\sum_{i=1}^{t} \Omega(f_i) = \sum_{i=1}^{t-1} \Omega(f_i) + \Omega(f_i) \quad (11)$$

A constant is $\sum_{i=1}^{t} \Omega(f_i)$. Equations (9) through (11):

$$obj(t) = \sum_{j=1}^{T} \sum_{i \in I_j} L(y_i, \hat{y}_i^{(t-1)} + \omega_j) + 1\,2\, \lambda \omega 2_j + \gamma T + \text{uniform} \quad (12)$$

F. **Implementation**

The implementation of an ML strategy involves multiple stages.

Step 1: Gathering Information. A large amount of data is required to train machine learning models successfully, so it's imperative to collect enough data for the project. Having sufficient domain knowledge is crucial to guaranteeing that the data gathered fulfills the project's specifications. Because of their varied sources, the data may arrive in both organized and unstructured formats, and they may include missing, duplicate, or even erroneous values. As a result, preparing data becomes essential before using ML algorithms[6].

Step 2: Data preprocessing: To guarantee the integrity of the data, a number of data quality tests are carried out at this step, including:

- Validity: Integers, Booleans, and dates are examples of pertinent data types that should be contained in columns. Dates should also be formatted correctly and fall between the specified range. Column mandatory fields must not be left empty. It is possible to create a column with unique values to act as an identifier if one does not alreadyexist.
- Data cleaning: In order to handle missing values, observations can be dropped or appropriate replacements can be added. Remove duplicate data if necessary. It could be necessary to convert values into formats that ML algorithms can use.
- Denoising: Here, mistake removal and variable variability reduction are the main objectives.This can be accomplished by employing strategies like clustering, regression, and binning.
- Outliers: It's critical to recognize and manage outliers. Values in a column that do not belong in the typical group or cluster are called outliers. It's critical to recognize and effectively handle outliers since they can significantly affect prediction outcomes, frequently in a negative way.

Step 3: Model Instruction. The preprocessed data are organized and prepared for machine learning. Iterating through many models and documenting and assessing their outputs—known as candidate models—are all part of the machine learning process. A candidate model is an algorithm or architectural arrangement that is being considered as a potential solution to a particular issue. ML professionals frequently evaluate a large number of candidate models to see which one works best for the task at hand.

Each model is trained and assessed using pertinent data for this evaluation, and metrics such as accuracy or loss functions are used. The ultimate decision is based on a number of variables, including resource limitations, interpretability, computing efficiency, and model correctness. Hyperparameter tweaking may be used in some circumstances to improve performance even more. In the end, selecting a candidate model seeks to achieve equilibrium between the demands of the problem and the computational capacity and knowledge accessible.

Step 4: Model Adjustment. In machine learning, "model tuning" refers to determining the ideal values for a model's configuration parameters, also known as hyperparameters. To identify the best combination, it involves selecting the hyperparameters to optimize, setting up a space for their search, and applying techniques like grid search, random search, or Bayesian optimization. Selecting the optimal hyperparameters for a given task will improve the model's performance.

Step 5: Forecasting and Implementation. Important phases in the lifecycle of an ML model are prediction and deployment. Using a trained model to generate educated guesses or classifications based on newly discovered data is prediction. This could entail classifying emails as spam or not, forecasting market

International Journal of Advances in Engineering and Management (IJAEM)
Volume 6, Issue 07 July 2024, pp: 195-200 www.ijaem.net ISSN: 2395-5252

values, or making medical image-based disease diagnoses.

In contrast, deployment entails incorporating a trained model into a useful application or system. It necessitates giving serious thought to variables like security, latency, and scalability. Deployed models have a critical role to play in utilizing ML for useful decision making and automation. Examples of these applications include fraud detection, recommendation systems, autonomous cars, and a host of other real-world scenarios.

Step 6: Keeping an eye on and maintaining. The model is referred to as the "Golden model" when the system has been put into use and final tests have been carried out to guarantee optimal performance. An established, reliable version of a model that functions as a performance benchmark is referred to as a "golden model." The original, well-known good state and behavior of the model are embodied in this reference model.

The accuracy and dependability of machine learning models can be affected by drift over time as a result of shifting data distributions or other variables. Deviations or deterioration in performance can be identified by continuously comparing the performance of the current model to the predicted outcomes of the golden model.To ensure continued efficacy and dependability in practical applications, major variances indicate the need for model retraining or maintenance to put the model's performance back in line with the intended standards.

Step 7: The evaluation measures of the machine learning models used to forecast home prices are covered in detail in this section. Providing a comprehensive grasp of the measures utilized for model comparisons and the best-performing model selection is the aim. As a result, we will talk about assessment measures such root mean square error (RMSE), mean absolute error (MAE), mean square error (MSE), and adjusted R-squared.

The R-squared value is a useful metric for evaluating the model's fit and prediction accuracy on hypothetical data sets. The following formula can be used to determine the R-squared value: formula for R-squared value

$$R^2 = 1 - SSR / SSM \quad (13)$$

where SSR and SSM stand for squared sum error of the regression line and mean line, respectively.

Once again, XGBoost excels with a strong adjusted R-squared result in the adjusted R-squared metric, which also evaluates the model's goodness of fit. Equation (14) gives the updated R-squared formula:

$$\text{Adjusted } R2 = 1 - [(n - 1)/(n - k - 1) *(1 - R^2)] \quad (14)$$

n is the number of observations, and k is the number of independent variables.The squared difference between actual and anticipated values is used to determine the mean square error, or MSE. A model that performsbetter is indicated by a lower MSE.

The formula for figuring out the model's MSE is shown in the equation below.

Moreover, the root mean squared error (RMSE), which is essentially the square root of MSE, is an extension of MSE. It is an extra measure to support our evaluation.

Once more, a smaller RMSE indicates that the model's predictions and the actual values agree closely.

$$MSE = 1/n\Sigma \{y - y\hat{}\}^2 \quad (15)$$

The last statistic looked at is the mean absolute error (MAE), which measures the discrepancy between values that are predicted and those that are observed. As shown in the equation below, it is calculated as the total of the absolute errors divided by the sample size.

$$MAE(y, y\hat{}) = 1^nsamples \sum nsamples^{-1} i=0 |yi - y\hat{}i^1| \quad (16)$$

A better model is indicated by a lower MAE score.

G. **Hyperparameter Tuning**

Investigating different options is required in order to build the best possible machine learning model. To get the right model architecture and hyperparameter configuration, hyperparameter adjustment is essential. Tuning parameters is crucial for creating an effective machine learning model, especially for deep neural networks and tree-based ML models that require many of them .

A scalable and intuitive framework for hyperparameter optimization, KerasTuner was created especially for TensorFlow, a well-known ML platform. Hyperparameters like the number of input neurons, the number of convolution layers, and the ideal number of epochs may all be adjusted with KerasTuner. In contrast, GridSearchCV is a method that makes use of hidden layer neurons in order to adjust hyperparameters and determine the optimal values for a given model.

Moreover, another option is RandomSearchCV, which may be accessed through the sklearn module. A method for optimizing through

International Journal of Advances in Engineering and Management (IJAEM)
Volume 6, Issue 07 July 2024, pp: 195-200 www.ijaem.net ISSN: 2395-5252

a cross-validated search across the parameter settings is provided by the library. Unlike GridSearchCV, a defined number of parameter settings are sampled from the given distributions rather than trying out every possible value. The value of n_iter indicates how many different parameter sets are attempted.

IV. RESULT

XGBoost clearly stands out as having the greatest R-squared value, at 0.93. After that, with an astounding cross-validation score (CV) of 88.940, XGBoost likewise has the highest CV. This score indicates how well the model generalizes over the whole dataset, which makes XGBoost the best regression method for this particular dataset. Now that the model's correctness is taken into account, XGBoost not only tops the field but also exhibits remarkable performance without the need for hyperparameter tweaking.

When the ideal settings are used, its accuracy increases even further. Next, we look at the adjusted R-squared measure, which evaluates the model's goodness of fit. Here, XGBoost performs exceptionally well, achieving a strong adjusted R-squared score. A model that performs better is indicated by a lower MSE.Remarkably, with a mere 0.001, XGBoost has the lowest MSE value in this case. Furthermore, a smaller RMSE indicates that the model's predictions and the actual values agree closely.

An intriguing anomaly is that, at 0.075, linear regression has the lowest MAE score, closely followed by XGBoost at 0.084. Even though linear regression performed notably well in this case in terms of MAE, preference still goes to XGBoost because of its superior performance in terms of all other measures.

V. CONCLUSION

We compared XGBoost, random forest, support vector, multi-layer perceptrons, and linear regression in this work. Next, we showed how to use GridSearchCV to perform hyperparameter tuning on the ML algorithms in order to obtain the best possible answer. Based on our findings, XGBoost is the best model for predicting housing prices, with a 0.001 minimum mean square error. In addition, we employed the power of ensemble trees (XGBoost) to pinpoint our model's salient characteristics.

Consequently, this study is a priceless resource for further research in this field, highlighting the continuous superiority of XGBoost and highlighting the importance of critical elements such as overall quality and ground floor living area in making well-informed decisions within the dynamic housing market. In summary, the following are the main conclusions drawn from our research.

• We suggest using the XGBoost algorithm as a more comprehensible, straightforward, and accurate model for predicting home prices.
• The performance of regression models varies.
• Though not always, hyperparameter adjustment continuously enhances model performance.
• In the context of house price prediction, it is important to identify and highlight key influential features.
• GridSearchCV hyperparameter adjustment is effective for improving model performance.

REFERENCES

[1]. Aljohani, O. Developing a stable house price estimator using regression analysis. In Proceedings of the 5th International Conference on Future Networks & Distributed Systems, Dubai, United Arab Emirates, 15–16 December 2021; pp. 113–118.
[2]. Manasa, J.; Gupta, R.; Narahari, N.S. Machine learning based predicting house prices using regression techniques. In Proceedings of the 2020 2nd International Conference on Innovative Mechanisms for Industry Applications (ICIMIA), Bangalore, India, 5–7 March 2020; pp. 624–630.
[3]. Dejniak, D. The Application of Spatial Analysis Methods in the Real Estate Market in South-Eastern Poland. Acta Univ. Lodz. Folia Oeconomica 2018, 1, 25–37. [CrossRef].
[4]. Rahman, S.N.A.; Maimun, N.H.A.; Razali, M.N.M.; Ismail, S. The artificial neural network model (ANN) for Malaysian housing market analysis. Plan. Malays. 2019, 17, 1–9.
[5]. Yalpir, S.; Unel, F.B. Use of Spatial Analysis Methods in Land Appraisal; Konya Example.

In Proceedings of the 5th International Symposium on Innovative Technologies in Engineering and Science (ISITES2017), Baku, Azerbaijan, 29–30 September 2017.

[6]. Madhuri, C.R.; Anuradha, G.; Pujitha, M.V. House price prediction using regression techniques: A comparative study. In Proceedings of the 2019 International Conference on Smart Structures and Systems (ICSSS), Chennai, India, 14–15 March 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 1–5.