Vol.15, Issue No 2, 2025

Enhancing House Price Prediction Based Machine Learning using Modified Gradient Boosting

[1] Miriyala Lakshmi Prasanna, [2] Gunduboyina Likhitha, [3] Donga Janayatri, [4] Jayamangala Yuva Ratna, [5] Ms. K. Siva Syamala

[1,2,3,4] Student, Dept. of CSE, DNR College of Engineering & Technology, Balusumudi, Bhimavaram, India.

[5] Assistant Professor, Dept. of CSE, DNR College of Engineering & Technology, Balusumudi, Bhimavaram, India.

ABSTRACT

Everyday speech requests and predictions have become more reliant on machine learning in recent years. Rather, it offers a more secure vehicle system and greater support for customers. After everything that has been shown, ML is a technology that is gaining popularity across many different sectors. The home Price Index is a common tool used to measure changes in home values (HPI). The HPI is insufficient to reliably predict a person's home price on its alone because of the strong association between property values and other factors like area, population, and geography. While some research has been able to accurately forecast home values using traditional machine learning methods, these studies almost never compare and contrast various models and completely exclude the more intricate but less well-known ones. Because of its adaptive and probabilistic model selection procedure, we suggested using Modified Extreme Gradient Boosting as our model in this research. This procedure includes creating features, training and optimizing hyper parameters, interpreting models, and finally, selecting and evaluating models. House price indices, which are often used to bolster housing policy efforts and provide cost estimates. The goal of this machine learning study is to create models that can predict future shifts in the housing market. Variegated extreme gradient boosting, square footage, location, and home price

Introduction

An important component of any thriving economy is the real estate market, of which the housing market is a part. Numerous recently hired individuals have made it their career goal to become homeowners, a goal that is highly valued in many areas of the globe due to the status symbol nature of owning one's own house. Regardless, investors seek out the real estate market for the opportunities it presents, rather than the commodities it really is [1]. A growing economy can't exist without a healthy real estate market, which in turn requires a strong housing market. Homeownership is a desire for many young professionals in many nations because of the status symbol it symbolizes. This is because having a house is a symbol of social standing. The housing market, however, is attractive to investors who see property not as a commodity but as an opportunity for profit [2]. Everyone from first-time homebuyers to seasoned investors usually gets into the real estate market hoping to cash in on future price gains. In general, the proportion of the population that owns their home decreases as house prices rise. Most studies have previously concentrated on countries with high homeownership rates, especially those whose economies are still growing [3].

Because housing costs have a major impact on the market's sustainability over the long run, it is critical that individuals have access to affordable housing that satisfies their fundamental necessities. Whether or whether buying a home is a smart long-term financial plan is heavily dependent on how affordable it is. Real estate market volatility is much lower compared to that of the stock market, interest rates, and currency exchange markets. Real estate has been one of the most lucrative investment industries in recent years, especially in the last fifteen, and price swings in homes have a major influence on this industry. More recently, among the most The logic underlying property pricing has been a hotly debated subject in the real estate industry. Residential investors, REITs, private investors, and officials from several government organizations are among those who have been asked to make predictions about the future of property prices because of this issue. For this purpose, these people have made use of a wide variety of strategies. There has been a critical shortage of housing due to the exponential growth of urban populations since the start of the Industrial Revolution. This is because urbanization is happening at a fast pace in every part of the globe right now. With the passage of time, several aspects of this issue have became clearer. While

Germany was still building its infrastructure in the 1980s, this book delves at the massive housing crises that hit the country's major cities. At this juncture, people all throughout the country began to accept the reality of the housing crisis, which was caused by economic disparity, and the need for more social housing. Cities in emerging nations with a younger urban population also face challenges associated with these qualities because of the relative immaturity of their urban infrastructure.

For instance, many studies have focused on the problem of subpar housing in India. Abhay said that, while the housing scarcity issue is stimulating the construction supply, low-quality housing is common across the city (2001-2011) when providing statistics on the number of dwellings created in Karnataka over the last decade. This was brought out by Abhay while discussing the amount of residential development projects in Delhi over the analyzed decade [8]. There is a severe lack of available housing, and many people blame the disproportionately high percentage of older homes for this. The complex interplay between a home's physical features, the surrounding area, and its location makes it difficult to provide an accurate valuation. Specifically, compared to more traditional methods for solving this issue, the house price projection model has gotten much less attention in the current literature. The fact that this continues happening despite the general agreement that it is a serious issue demands immediate attention. With the advent of Big Data, machine learning has become a prominent tool for making predictions [9].This has allowed for the development of more precise property price estimates that are not dependent on past data but rather on the characteristics of the property itself. Though many studies have looked into this and discovered that machine learning works, most of them have just compared the results of various models without considering the best way to combine them [10]. The goal of this study is to anticipate home prices using an algorithm modification called extreme gradient boosting. Consequently, the ultimate goal of this study is to have a better understanding of regression approaches in machine learning. Processing the supplied datasets is essential for achieving best performance. As the worth of a property is based on its unique qualities, [11] we need to identify the most relevant ones before we can exclude the less important ones and get a good estimate. Since not all properties would be able to afford these modifications, the statistics are skewed; without them, house prices wouldn't fluctuate as much.Predicting future home prices using the Modified Extreme Gradient Boosting algorithm is the primary focus of this study. Predicting the price based on factors such as area type, location, BHK, etc. Several approaches are used to test the accuracy and performance of the Modified XGBoost.

Literature Survey

The most accurate ML models for estimating property prices were investigated by Park et al. [4]. They did this by looking at 5359 row dwellings in Virginia. The RF technique was used to handle classification problems, while the nave Bayesian algorithm was used to handle regression problems; these were the two approaches that were used. The RIPPER algorithm greatly improved price prediction, the results showed. In order to forecast whether real estate values would rise or fall in the future, Banerjee et al. [12] explored several machine learning techniques. Despite having the best accuracy rate, the RF technique was shown to have the highest overfitting. The SVM method, on the other hand, was the most reliable as it did not alter during the research. The practicality of several machine learning techniques within the framework of property evaluations was explored by Kok et al. [13]. The RF technique and the GBR method were among these approaches. In terms of overall performance, the results showed that the XGBM algorithm was the best. If Ceh et al. [14] wanted to know which method would provide better price predictions, they contrasted the hedonic pricing model with the RF algorithm. Between 2008 and 2013, the writers polled 7,407 residences in the Ljubljana region. The results showed that the RF model performed better than the others in terms of prediction, which was great news for Slovenia. 5. The accuracy of predicting the cost of renting a property in Shenzhen (China) was tested by Hu et al. [15] using supervised learning techniques. In their study, the authors used several neural network methods, including k-NN, RF, ETR, GBR, and SVR. The results showed that the ETR and RF algorithms were more consistent in their behavior. An updated methodology for assessing the geographical proximity of London and better performance estimates for property prices was developed by S. Lu et al. [16]. Neither of these geographical characteristics can be measured in a linear fashion since they are non-linear. The aim of Elham Alzain et al. [17] is to estimate future house prices in Saudi Arabia using an ANN-based model. Riyadh, Jeddah, Dammam, and Al-Khobar are four major cities in Saudi Arabia where Aqar was collected. There was a high level of agreement between the experimental and predicted values, according to the results. An accuracy of 80% is achieved by using ANN in this. P. Durganjali et al. [18] used classification algorithms to forecast how much houses would cost to resell. This research use a range of classification algorithms, such as Decision Tree, Random Forest, Leaner regression, and K-Means, to forecast the selling price of a property. Many factors, including the current economic climate, a home's location, and the home's physical characteristics, contribute to its final selling price. In this case, we use these methods, utilize RMSE as the performance matrix across several datasets, and identify the best model for accurately predicting enhanced outcomes. Sifei Lu et al. [19] developed a hybrid regression method for predicting home prices. In order to evaluate the creative feature engineering technique, this research uses a limited dataset and data characteristics. This concept was used as the basis for recent submissions in the "House Price: Advanced Regression 6 Methods" Kaggle competition. The objective of the essay is to provide readers with an idea of fair prices for those who buy from you. When utilizing hedonic analysis to predict home prices, Fletcher et al. [20] (2000) look at the pros and cons of employing aggregated vs disaggregated data. It turns out that hedonic price coefficients for a number of qualities change drastically with age, geography, and property type. Nevertheless, it is thought that this might be adequately replicated using an aggregate approach. Additionally, the hedonic pricing model has been used to quantify the individual external influences of elements such as environmental features on house prices. To quantify the impact of noise and air pollution on house values, for example, the hedonic pricing model has been used in several research.

Problem Statement

Some simple and basic houses may be predicted using the present method. Various algorithms' accuracy may also be predicted using this method. It uses only one parameter to make price predictions. In its present state, the system is unable to reap the benefits of its training phase. They have no idea what a lavish residence would look like. It can't recommend the best algorithms [21]. It is not possible to utilize certain parameters for prediction. The objective function to minimize at iteration t is the loss function with regularization:

Real value (label) known from the training data-set

$$\mathcal{L}^{(t)} = \sum_{i=1}^{n} l(y_i, \hat{y}_i^{(t-1)} + f_t(\mathbf{x}_i)) + \Omega(f_t)$$

Can be seen as f(x + Δx) where x = $\hat{y}_i^{(t-1)}$

The XGBoost objective "cannot be maximized using typical optimization methods in Euclidean space," as said, as it is clearly a function of functions. For the function f(x simplest), the following is the basic linear approximation:

$$f(x) \approx f(a) + f'(a)\boxed{(x-a)}$$

$$\Delta x = f_t(\mathbf{x}_i)$$

A function of x is the sole valid choice for representing the initial function: Only in terms of x is the initial function expressible. Applying Taylor's theorem, one may find the simplest function of x at a given point and transform it into f(x). The objective function f(x), where x stood for the total of the t CART trees, was dependent on the presently chosen tree (step t) before the Taylor approximation was applied. Here, x is the variable that has to be added in step t, the loss function is represented by the equation f(x), and the expected result from step t-1 is an. By presenting the new learner at each iteration as a simple function of the goal (loss) function, it is feasible to optimize in Euclidean space using the previously given information [22]. Because of this, optimization in t-space, which is defined by Euclidean geometry, is feasible. Step (t) requires an extra learner to eagerly lower the objective, as previously stated; this learner stands in for the forecast that was complete in stages (t-1) and (x-a). You can't eagerly lower the goal without doing this. When the second-order Taylor approximation is used:

$$f(x) \approx f(a) + f'(a)(x-a) + \frac{1}{2}f''(a)(x-a)^2$$

$$\mathcal{L}^{(t)} \simeq \sum_{i=1}^{n}[l(y_i, \hat{y}^{(t-1)}) + g_i f_t(\mathbf{x}_i) + \frac{1}{2}h_i f_t^2(\mathbf{x}_i)] + \Omega(f_t)$$

XGBoost objective using second-order Taylor approximation

Where:

$$g_i = \partial_{\hat{y}^{(t-1)}} l(y_i, \hat{y}^{(t-1)}) \text{ and } h_i = \partial^2_{\hat{y}^{(t-1)}} l(y_i, \hat{y}^{(t-1)})$$

The loss function's first and second order gradient statistics

Leaving just the goal to reduce at step t remains after removing the constant components, which is as follows:

$$\tilde{\mathcal{L}}^{(t)} = \sum_{i=1}^{n}[g_i f_t(\mathbf{x}_i) + \frac{1}{2}h_i f_t^2(\mathbf{x}_i)] + \Omega(f_t)$$

XGBoost simplified objective

Next, we want to find a learner that minimises the loss function at iteration t. This is because the loss function is just a collection of simple quadratic functions of one variable, and we know how to minimizes them.

$$argmin_x \; Gx + \tfrac{1}{2}Hx^2 = -\tfrac{G}{H}, \; H > 0 \qquad \min_x \; Gx + \tfrac{1}{2}Hx^2 = -\tfrac{1}{2}\tfrac{G^2}{H}$$

Minimizing a simple quadratic function

Keep in mind that the following scoring function is similar to the "basic quadratic function solution" mentioned before when using the authors' technique to "assess the quality of a tree structure q":

$$\tilde{\mathcal{L}}^{(t)}(q) = -\frac{1}{2}\sum_{j=1}^{T}\frac{\overbrace{\boxed{(\sum_{i\in I_j} g_i)^2}}^{\text{instances mapped to leaf}_j}}{\sum_{i\in I_j} h_i + \lambda} + \gamma T.$$

The tree learner structure q scoring function

$$y\ln(p) + (1-y)\ln(1-p) \text{ where } p = \frac{1}{(1+e^{-x})}$$

Assuming that both p and y are real labels with values between 0 and 1, binary classification using the Cross Entropy loss function is used [23]. An output x is produced by the model as a whole by the CART tree learners. In order to minimize the log loss objective function, we need to find the first and second derivatives of the hessian and gradient with respect to x. The formula for the hessian is (p-*) (1-p), while the formula for the gradient is (p-y), as you may have learned from this Stats Stack Exchange topic.

Proposed System

In a risk-free way, this study is used to forecast the home price. For excellent geographic locations, it can forecast home prices. For improved precision, it employs Extreme Gradient Boosting. Using linear functions, likelihood, etc., this approach makes predictions simpler and more accurate.

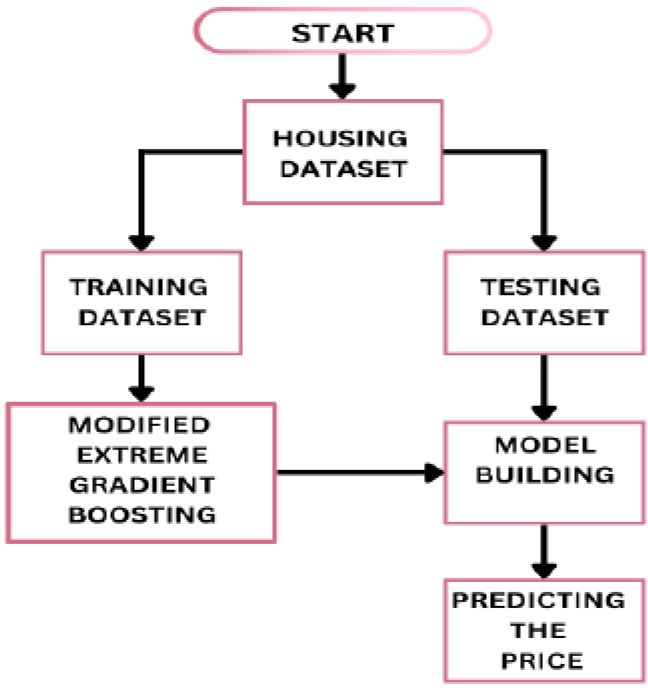

Figure 1. Flow Diagram

We pre-process the data by removing noise, detecting anomalies, and filling in missing values, among other things, when we get it from the database. Several methods are used to train the model on the pre-processed data, and then testing data is used to test the model's accuracy. Make use of a tweaked version of the extreme boosting technique to construct the model. Run the model once you've connected it to flask. Gather user input from the webpage and extract relevant keywords. After that, using the provided data, estimate the value.

System Architecture

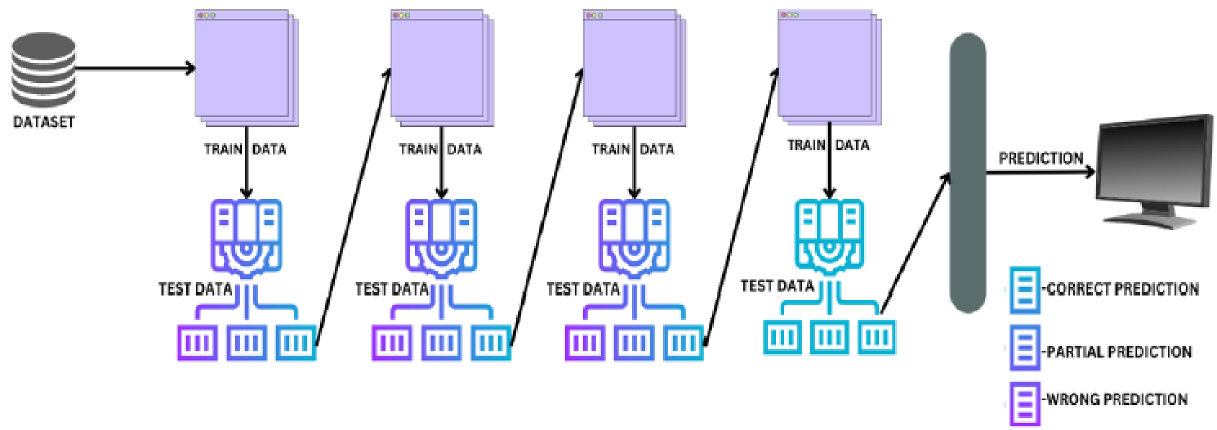

Figure 2. Proposed architecture

Algorithm 1. Modified Extreme Boosting

Initialization:

STEP 1: Given training information from the instance space S = (x1, y1),..., (m, ym) where xi EX and y; Y = -1,+1.

STEP 2: start the distribution off. $D_1(i)$ =.

Using the algorithm for t = 1,..., T, do

Utilizing the distribution Dt, train a weak learner ht: X R.

Zt is a normalisation factor chosen to create the distribution of Dt+1, and its formula is

Dt+1(i) = Dt(i)e-tyiht(xi) Zt.

f(x) = oatht(x) and H(x) = sign(f(x)) are the final scores.

Modified Extreme Boosting Algorithm A. General Parameters silent:

The value is initially set to zero. To enable quiet mode, enter 1, and to print ongoing messages, enter 0. supporter: GBTree is picked as the default. It is necessary to specify the booster to be used: both GBTree and GBLinear, which are tree-based and linear, respectively. buffer: The XGBoost algorithm sets it automatically; manually adjusting it is unnecessary. The XGBoost Algorithm automatically determines the value of the num feature; no human input is necessary. Section B. Booster Settings As a default, ETA is set to 0.3. Specifying the step size shrinkage used in an update is necessary to prevent overfitting. After each level of boosting, you may quickly see the weights of newly added features. The boosting method is made more conservative by eta by lowering the feature weights. There is a range of 0 to 1. A low eta value makes the model more overfit-resistant. GAMMA: Zero is the default value. The least loss reduction needed to establish another division must be indicated on a leaf node of the tree. As its size increases, the algorithm will exhibit a more cautious behavior. The range is from zero to. As gamma becomes greater, the algorithm becomes more cautious. The default setting for MAX DEPTH is six. Specify the maximum depth to which a tree may grow. 1 to complete the spectrum. The minimum child weight is 1 by default. It is necessary to specify the absolute minimum instance weight (hessian) that a kid must have. After the tree partitioning process, a leaf node will emerge if its combined instance weight is smaller than the minimal child weight. More divisions will be abandoned by the building procedure. is equivalent to the minimal number of instances needed for each node when using linear regression mode. As its size increases, the algorithm will exhibit a more cautious behavior. The range is from zero to. As a default, MAX DELTA STEP is set to 0. The biggest possible delta step for estimating the weight of each tree. Setting the value to 0 removes all constraints. For a more cautious update, setting it to something positive may help. This parameter might be helpful in logistic regression, although it's usually not needed. in particular when the class is grossly imbalanced. Control updates may be easier if set to a value between 1 and 10. The range is from zero to. The default choice for SAMPLE is 1. An instruction for the training instance's subsample ratio is required. On a 0.5 setting, XGBoost will have randomly selected half of the data instances. To avoid overfitting, it's important to let trees grow. The range is from zero to one. The default value for COLSAMPLE BYTREE is 1. You need to provide the subsample ratio of columns when you construct each tree. There is a range of 0 to 1.

Linear Booster Specific Parameters

We utilize these XGBoost algorithm specific linear booster parameters. Weight regularization is known as LAMBDA and Alpha. The expected value of alpha is 0, but the default value of lambda is 1. One L2 regularization term on bias is LAMBDA BIAS, which by default has the value 0. D. Specifying the Learning Tasks The following is a list of the learning task settings used by the XGBoost algorithm. This field's default value is 0.5 for BASE SCORE. It is necessary to provide the global bias and beginning prediction score for every event. The goal is: By default, the value is set to reg: linear. You need to know exactly what sort of student you're looking for. Among them are Poisson and linear regression. EVAL METRIC: It is necessary to provide the evaluation metrics for the validation data. More than that, the objective will dictate the default measure. SEED: As before, you need to provide the seed here if you want to get the same set of outputs. 3.3. Constructing Models In order to construct a machine learning model, you need two datasets: a training dataset and a testing dataset. But now there's only the one. The 80:20 rule says we should split it in half. Make sure the data frame has two columns: features and labels. Right here, we loaded the train-test split function that Sklearn provides. Next, use it to partition the dataset. In addition, the dataset is divided into two parts: the test part contains 20% of the data and the train part has 80%. The test size is 0.2. To divide the dataset, we use a random number generator that is initialized with the random state parameter. The procedure produces four data sets. Test x, test y, train x, and train y were the names given to them. Modified Extreme Boosting may be used to match data to several decision trees. Lastly, send the x and y trains to the fit method so it may train the model. It is necessary to test the model with test data after training it. Modified extreme boosting is a powerful machine learning strategy for regression problems. The random forest algorithm belongs to the class of supervised algorithms. The technique is executed in three stages: first, the independent variable is weighted; second, the forest is created from the dataset supplied; and third, predictions are made using the regressor.

Table 1. Algorithm used and its function

ID	Model	Function
1	Linear Regression	sklearn.linear_model.LinearRegression
2	Random Forest Regressor	sklearn.ensemble.RandomForestRegressor
3	Gradient Boosting Regressor	sklearn.ensemble.GradientBoostingRegressor
4	Ridge Regression	sklearn.linear_model.Ridge
5	Lasso Regression	sklearn.linear_model.Lasso
6	Ada Boosting Regression	sklearn.ensemble.AdaBoostRegressor
7	Decision Tree Regression	sklearn.tree.DecisionTreeRegressor
8	Modified Extreme Boosting	XGBRegressor()

Result and Discussion

The figure 3 shows the data and parameter present in the dataset. It also shows the columns and the rows in the csv file.

```
In [57]: housing_clean.sort_values(["price"], ascending=False)
```

Out[57]:

	area_type	availability	location	bath	balcony	price	bhk	sqft	price_per_sqft	sqft_per_bhk
12443	Plot Area	Ready To Move	Other	8	4	2600.0	4	4350.0	59770.114943	1087.5
6421	Plot Area	Soon to be Vacated	Bommenahalli	3	2	2250.0	4	2940.0	76530.612245	735.0
8398	Super built-up Area	Ready To Move	Bannerghatta Road	4	5	1400.0	5	2500.0	56000.000000	500.0
9535	Plot Area	Ready To Move	Indira Nagar	5	4	1250.0	4	2400.0	52083.333333	600.0
1299	Plot Area	Ready To Move	Chamrajpet	7	1	1200.0	9	4050.0	29629.629630	450.0
...	...	...	...	...	...	...	...	...	...	...
5410	Super built-up Area	Ready To Move	Attibele	1	1	10.0	1	400.0	2500.000000	400.0
11091	Built-up Area	Ready To Move	Attibele	1	1	10.0	1	410.0	2439.024390	410.0
7482	Super built-up Area	Ready To Move	Other	2	1	10.0	1	470.0	2127.659574	470.0
12579	Super built-up Area	Ready To Move	Chandapura	1	1	10.0	1	410.0	2439.024390	410.0
8594	Built-up Area	Ready To Move	Chandapura	1	1	9.0	1	450.0	2000.000000	450.0

12339 rows × 10 columns

Figure 3. Dataset Description

Figure 4 displays the dataset's accompanying graph. Histograms and bar graphs show the data's ratio. You can see the count of each region type in the dataset represented by the bar graph. The count and density of sqft_per_bhk (the parameter's assigned weight) are shown in the first histogram. The price per square foot is shown in the second histogram. The dataset's square footage is shown in the third histogram. The purchase price per square foot is shown in the fourth histogram. The dataset's pricing is shown in the last histogram.

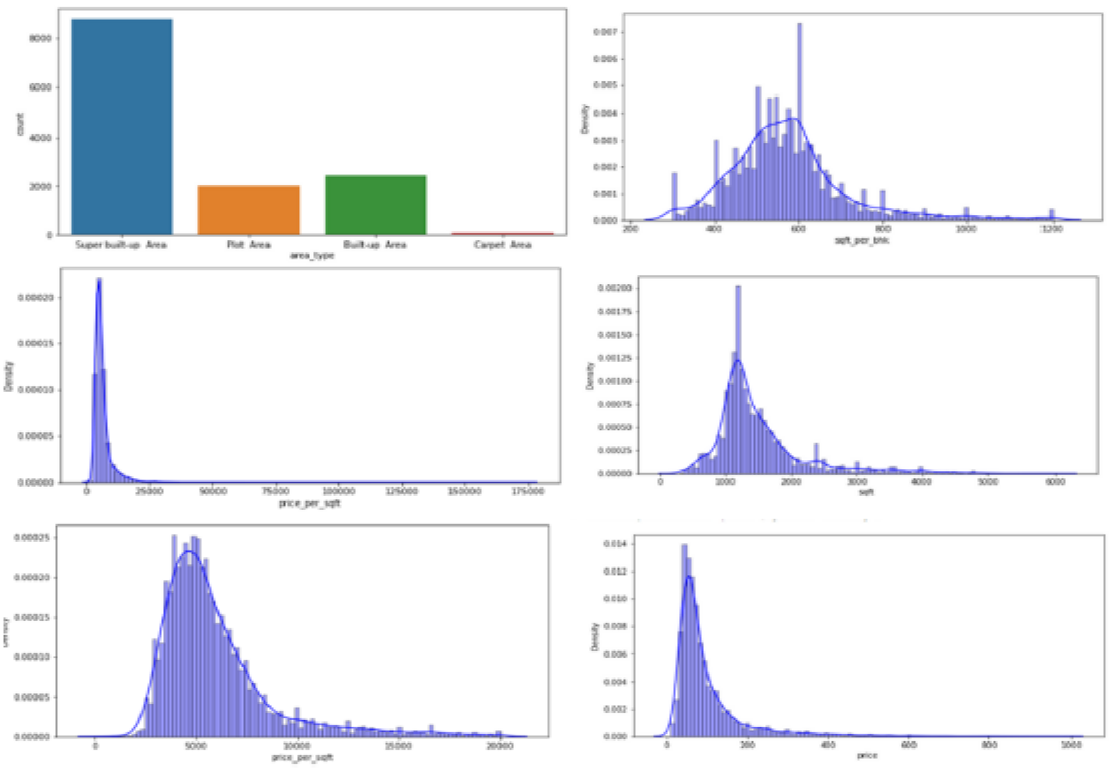

Figure 4. Data Graph

You can see the different algorithms and their respective accuracy rates in figure 5. To evaluate the efficacy of Modified XGBoost, we compare its results to those of Linear Regression, Ridge Regression, AdaBoost Regression, Decision Tree Regression, Random Forest Regression, Gradient Boosting Regression, and Modified Extreme Boosting Algorithms.

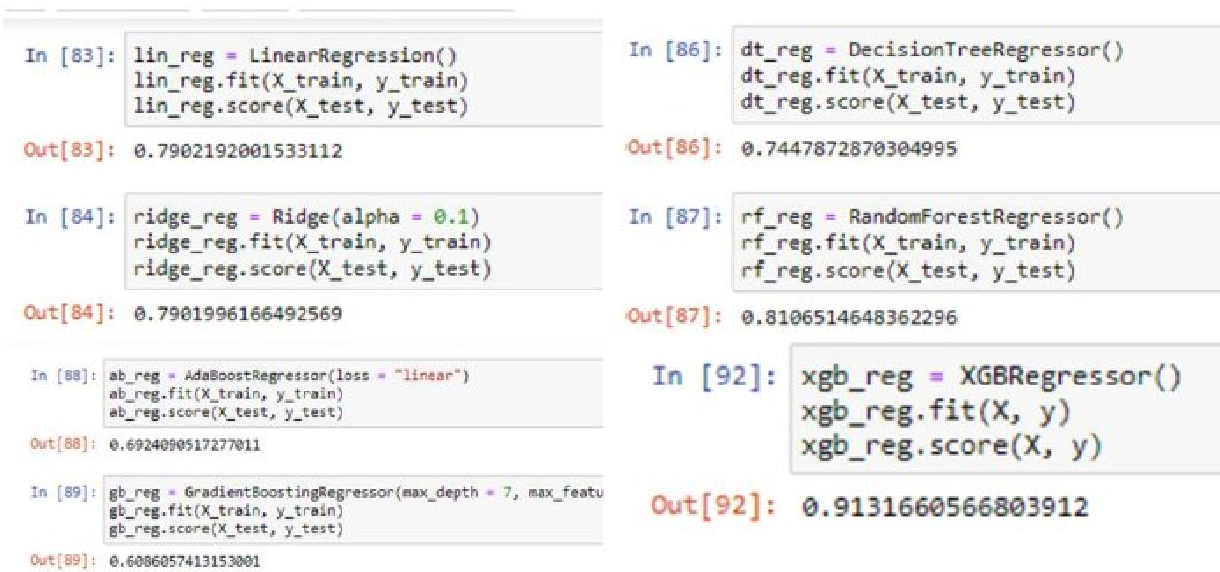

Figure 5. Accuracy

Table 2. Accuracy table

S. No.	Algorithm	Accuracy
1	Linear Regression	0.78021
2	Ridge Regression	0.79019
3	Lasso Regression	0.76216
4	Decision Tree (DT)	0.74478
5	Random Forest (RF)	0.81065
6	AdaBoost (AB)	0.69240
7	Gradient Boosting Tree (GB)	0.60806
8	Modified Extreme Boosting	0.82912

Figure 6 displays a comparison graph of the algorithms, illustrating their performance and efficiency. Among these algorithms, Modified XGBoost stands out as the most efficient.

```
In [1]: import matplotlib.pyplot as plt


        algorithms = ['Linear','Ridge','Lasso','DT','RF','AB','GB','XGBoost']
        accuracy = [0.79021, 0.79019, 0.76216, 0.74478, 0.81065, 0.69240, 0.608060, 0.91912]
        plt.plot(algorithms, accuracy)
        plt.xlabel("Algorithm")
        plt.ylabel("Accuracy")
        plt.title("Accuracy of Various Algorithms")
        plt.ylim(0, 1)
        plt.show()
```

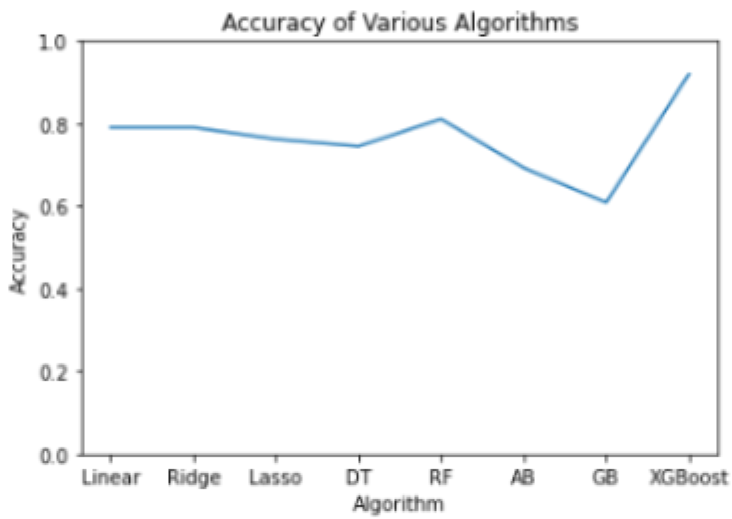

Figure 6. Accuracy graph

Figure 7 displays the flask front end of the application running on server 127.0.01.5000. There are specifics like address, neighborhood, availability, square footage, bedrooms, and baths included. It is up to the user to supply the data.

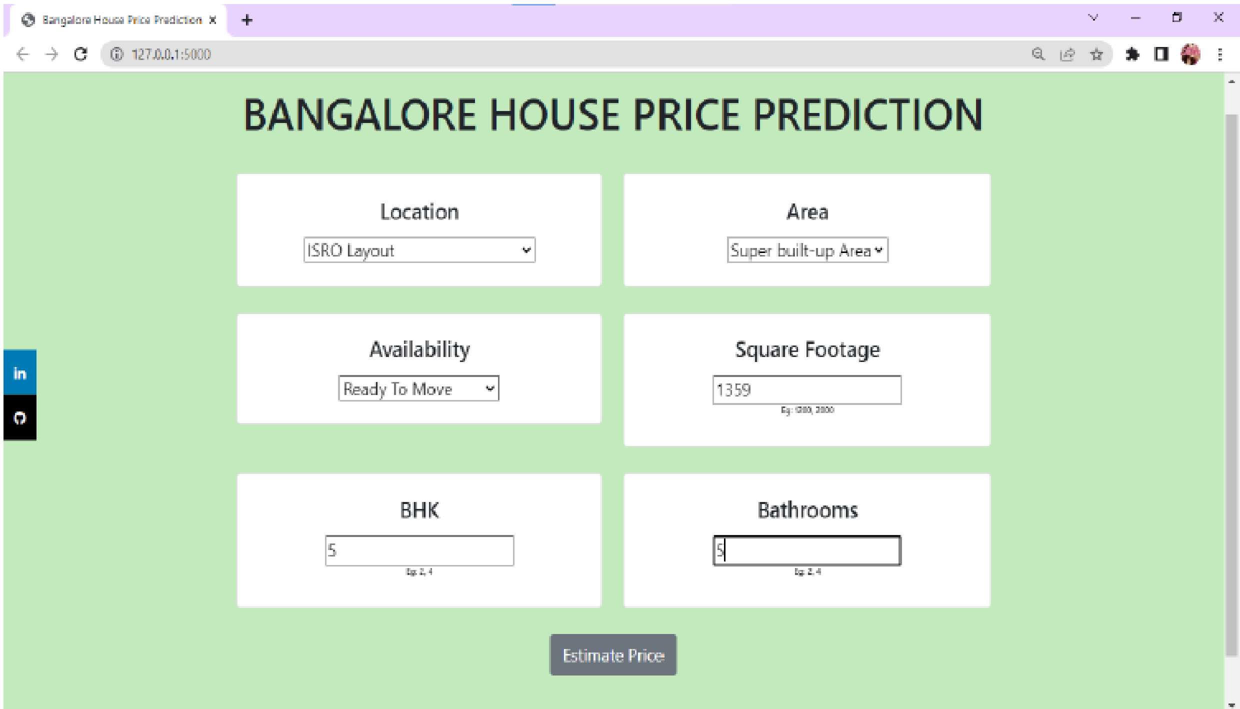

Figure 7. Running website on flask

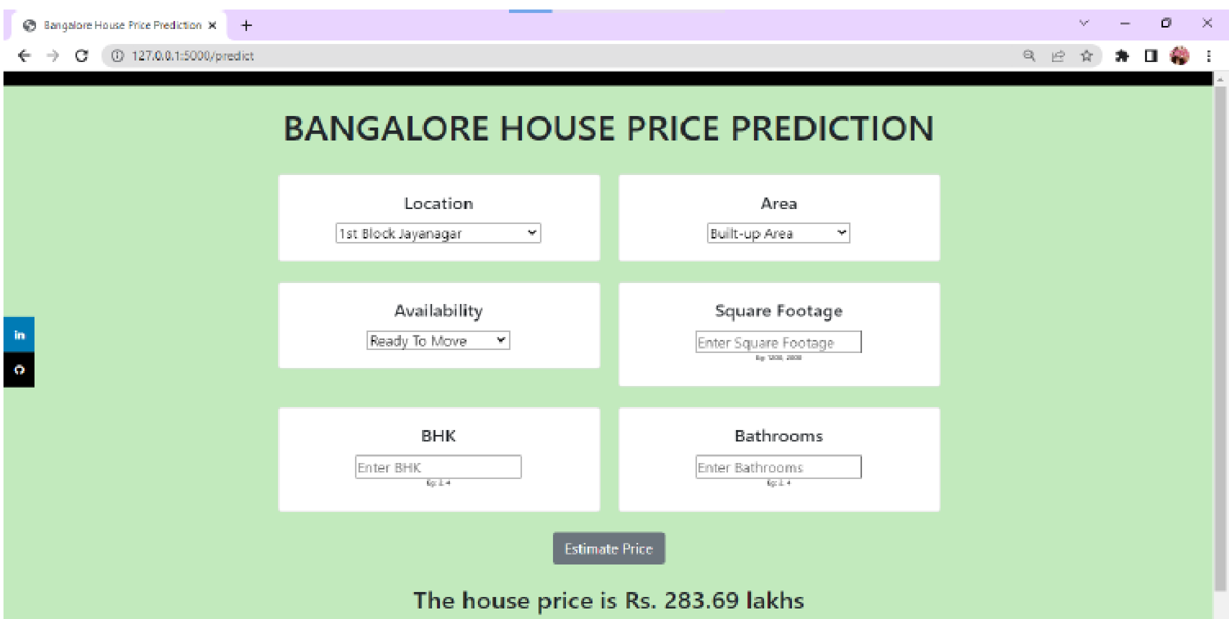

Figure 8. House price prediction

The figure 8 shows the prediction of house price for the given input.

Conclusion

Any growing economy relies on the addition of new employment, and the real estate industry might play a role in this. Owners and recipients of property are entangled in this situation. That is why accurate predictions of future property values are crucial. Real estate investors and homeowners alike pay careful attention to price movements in their properties because they serve as a proxy for the state of the economy. One useful technique for controlling property usage and budgeting is a model that can predict future housing expenses. The ability to predict the future value of real estate has several applications, including helping politicians establish fair prices and empowering owners and brokers to make educated choices. In this study, we evaluate Bayesian Regression and other popular regression algorithms for their ability to forecast Bangalore housing prices. The excellent correlation and number of characteristics in the publicly accessible data made the results encouraging. Hence, the local data need more attributes, preferably those that correlate strongly with house price. Nevertheless, XGBoost yielded the most favorable outcomes. The results show that compared to other prediction algorithms, Modified XGBoost is the best. Incorporating social media pricing data, Google Maps photos, user assessments of the property's features, and economic information might enhance ML projections in the future.

References

[1] Gallent, Nick, Phoebe Stirling, and Iqbal Hamiduddin. (2022). Pandemic mobility, second homes and housing market change in a rural amenity area during COVID-19–The Brecon Beacons National Park, Wales. Progress in Planning, 100731.

[2] Gajalakshmi, R. K., et al. (2020). An Optimized ASM based Routing Algorithm for Cognitive Radio Networks. International Conference on System, Computation, Automation and Networking (ICSCAN), IEEE.

[3] Padmini, S. V. (2022). Socio-economic conditions of slum dwellers in Karnataka-a case study in Tumkur district. Archers & Elevators Publishing House.

[4] Raj, S. Gokul, N. Srinath, and T. Ananth Kumar (2019). Real-Time Trespasser Detection Using GPS based UAV. IEEE International Conference on Innovations in Communication, Computing and Instrumentation (ICCI).

[5] Tokunaga, Robert S. (2010). Following you home from school: A critical review and synthesis of research on cyberbullying victimization. Computers in Human Behavior, 26(3): 277-287.

[6] Arunmozhiselvi, S., et al. (2022). A Systematic Approach to Agricultural Drones Using a Machine Learning Model. Machine Learning Approaches and Applications in Applied Intelligence for Healthcare Data Analytics. CRC Press, Pages 41-60.

[7] Nguyen, Quang Hung, et al. (2021). Influence of data splitting on performance of machine learning models in prediction of shear strength of soil. Mathematical Problems in Engineering, 2021: 1-15.

[8] Rajmohan, R., et al. (2022). G-Sep: A Deep Learning Algorithm for Detection of Long-Term Sepsis Using Bidirectional Gated Recurrent Unit.

[9] Ford, Elizabeth, et al. (2019). Identifying undetected dementia in UK primary care patients: a retrospective case-control study comparing machine-learning and standard epidemiological approaches. BMC Medical Informatics and Decision Making, 19(1): 1-9.

[10] Muthukumarasamy, Sugumaran, et al. (2020). Machine learning in healthcare diagnosis. Blockchain and Machine Learning for E-Healthcare Sys., 343.

[11] Mair, Carolyn, et al. (2000). An investigation of machine learning based prediction systems. Journal of systems and software, 53(1): 23-29.

[12] Saravanan, V., et al. (2023). Design of deep learning model for radio resource allocation in 5G for massive IoT device. Sustainable Energy Technologies and Assessments, 56: 103054.

[13] Čeh, Marjan, et al. (2018). Estimating the performance of random forest versus multiple regression for predicting prices of the apartments. ISPRS International Journal of Geo-Information, 7(5): 168.

[14] Hong, Yao, Jianhong Kang, and Ceji Fu. (2022). Rapid prediction of mine tunnel fire smoke movement with machine learning and supercomputing techniques. Fire Safety Journal, 127: 103492.

[15] Lu, Binbin, et al. (2014). Geographically weighted regression with a non-Euclidean distance metric: a case study using hedonic house price data. International Journal of Geographical Information Science, 28(4): 660-681.

[16] Alzain, Elham, et al. (2022). Application of Artificial Intelligence for Predicting Real Estate Prices: The Case of Saudi Arabia. Electronics, 11(21): 3448.

[17] Durganjali, P., and M. Vani Pujitha. (2019). House resale price prediction using classification algorithms. International Conference on Smart Structures and Systems (ICSSS), IEEE.

[18] Lu, Sifei, et al. (2017). A hybrid regression technique for house prices prediction. International Conference on Industrial Engineering and Engineering Management (IEEM), IEEE.

[19] Prasanalakshmi, B. (2022). Deep Regression hybridized Neural Network in human stress detection. International Conference on Smart Technologies and Systems for Next Generation Computing (ICSTSN), IEEE.

[20] Fletcher, Mike, Jean Mangan, and Emily Raeburn. (2004). Comparing hedonic models for estimating and forecasting house prices. Property Management, (3): 189-200.

[21] Hallegatte, Stéphane, et al. (2012). Investment decision making under deep uncertainty-application to climate change. World Bank Policy Research Working Paper – 6193.

[22] Sun, Shiliang, et al. (2019). A survey of optimization methods from a machine learning perspective. IEEE Transactions on Cybernetics, 50(8): 3668-3681.

[23] Coenen, Lize, et al. (2022). Machine learning methods for short-term probability of default: A comparison of classification, regression and ranking methods. Journal of the Operational Research Society, 73(1): 191-206.