Predicting House Prices using Machine Learning and LightGBM.

Asst. Prof. Linda John
Department of Information Technology,
St. John College of Engineering and Management,
Palghar, India
lindaj@sjcem.edu.in

Rakshit Shinde
Department of Information Technology,
St. John College of Engineering and Management,
Palghar, India
rakshits@sjcem.edu.in

Shoaib Shaikh
Department of Information Technology,
St. John College of Engineering and Management,
Palghar, India
shoaibs@sjcem.edu.in

Devanshu Ashar
Department of Information Technology,
St. John College of Engineering and Management,
Palghar, India
devanshua@sjcem.edu.in

Abstract:
Purchasing a home in today's environment is a time-consuming process. It is necessary to approach a real estate agent and obtain information, which is not safe and can result in theft. We are attempting to create an online efficient means of determining house prices for a certain region based on customer criteria using Machine Learning in this research project. To get the best results, we're using the LightGBM framework to train the dataset. For predicting property values, the most optimal and precise result will be chosen. LightGBM (Light Gradient Boosting Machine) is a free and open source distributed gradient boosting system for Machine Learning that was created by Microsoft. It is used for ranking, classification, and other machine learning applications and is based on decision tree algorithms. Performance and scalability are at the forefront of the development process. We also employed the Naive Bayes method to evaluate the likelihood of finding a house in a specific location where the costumer is interested in purchasing a home. The accuracy obtained utilising the methods demonstrated that this strategy offers the least error and highest accuracy when compared to other Gradient Boosting Machine frameworks.

Machine Learning, LightGBM, Naive Bayes, Gradient Boosting Machine, and Decision Tree are some of the terms used in this paper.

I) Introduction

Now-a-days buying a house is not an easy thing to achieve, mostly in metropolitan cities. Different things need to be taken into consideration. Doing this on own is certainly impossible. Real Estate individuals play a very crucial role in this process. But there is a theft of getting fooled by them, in order to avoid this Machine learning Algorithms can be used. This Algorithms can provide better prediction, more efficient than Real Estate people. Here, user can provide his flat requirements and depending on the user input output will be provided. User may give invalid input which may generate invalid output, thus in order to avoid this we have made used of Naïve Bayes algorithm. With the help of Naïve Bayes, we will determine if the input given by the user is valid or not. Depending on which respective output is displayed.

Data is at the heart of technical innovations, achieving any result is now possible using predictive models. Machine learning is extensively used in this approach. Machine learning means providing valid dataset and further on predictions are based on that, the machine itself learns how much importance a particular event may have on the entire system based on its pre-loaded data and accordingly predicts the result. Various modern applications of this technique include predicting stock prices, predicting the possibility of an earthquake, predicting company sales and the list has endless possibilities.

For our research project, we have considered Bangalore as our primary location and are predicting house prices for various localities in and around Bangalore. We have used parameters like 'square feet area', ‘garage square feet area’, 'number of Bedrooms', 'number of Bathrooms', number of balconies.'. We have taken into account a verified dataset with diversity so as give accurate results for all conditions. We have used various algorithms explained below in various combinations and the weight for each algorithm is given based on the accuracy percentage.

II) Related Work

The real estate industry has become a competitive and non-transparent industry. The data mining process in such an industry provides an advantage to the developers by processing those data, forecasting future trends and thus assisting them to make favourable knowledge-driven decisions. Manasa J, Narahari N S and Radha Gupta [1] used five regression techniques which included Linear Regression, Lasso and Ridge regression, support vector regression, and boosting algorithm which included Extreme Gradient Boost Regression (XGBoost). Out off this regression techniques, XGBoost showed best possible results. But the paper focused on using regression techniques for machine learning hackathons.

Hackathons generally use dataset up to certain limit thus restricting the algorithms for industrial use.

Ayush Verma, Abhijit Sarma, Sagar Doshi and Rohini Nir [2] also made use of regression techniques such as Linear Regression, Random Forest Regression and Boosted Regression. Instead of comparing individual techniques they combined the results generated by each technique and passed it as an input to their Neural Network model. The Neural Network Model compared the best result and thus improved the accuracy of the prediction. The RMSE value of Neural Network Model was relatively lower. Thus, they concluded that using Neural Network Model highly increases the accuracy of prediction. B.Vijay Kumar, B.Ashritha, CH.Teja and M.Vineeth [3] made use of Gradient Boosting Regression Model to predict house prices. In this model they created additive model in a forward stage wise fashion. The basic idea of using boosting was to identify whether a weak learner can be modified to become a better learner. The outcome was positive as the weak learner were converted into good learners which improved the accuracy of model for prediction.

P. Durganjali and M. Vani Pujitha [4] uses different classification algorithms like Logistic Regression, Decision tree, Naïve Bayes and Random forest. At last, they made use of AdaBoost algorithm for boosting up the weak learners to strong learners. The most accurate algorithm was determined as Decision tree on applying AdaBoost the accuracy highly increased. Yun Zhao, Girija Chetty and Dat Tran [5] made a hybrid model which combined basic attributes of property, such as information about bedrooms, bathrooms, as well as the visual features extracted from the property images for the price estimation task. The experimental evaluation showed that deep learning combining with XGBoost can result in better price estimation performance.

Our main focus here is to develop a model which predicts the property cost for a customer according to his\her interests. Our model analyses a set of parameters selected by the customer so as to find an ideal price according to their requirements and interest. In most of the research paper we observed that XGBoost was most commonly used for prediction purpose. It is very fast as compared to other algorithms such as Linear Regression, Random Forest, etc.

But we also observed that it has some drawbacks. The main drawback of using XGBoost is that its performance reduces as the size of dataset increases. Since, dataset used for house price prediction usually consists of huge data comprising of 'square foot area', 'location', 'number of bedrooms', 'number of bathrooms', 'garage square foot area', etc. it will lead to slow down of XGBoost. Thus, in order to overcome this drawback, we came up with another Gradient Boosting technique.

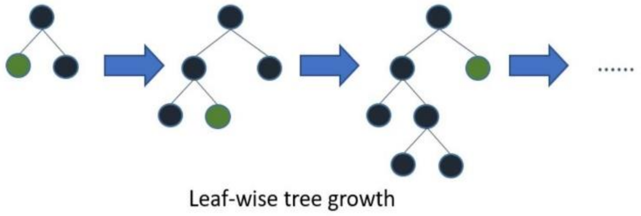


Figure 1: Growth of LightGBM tree

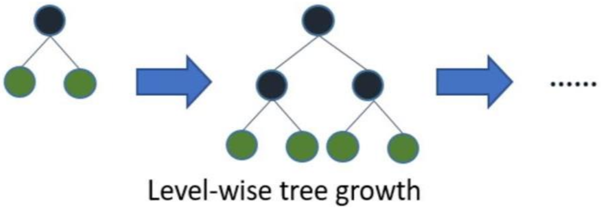


Figure 2: Growth of other boosting algorithms tree

In our project it is LightGBM, which stands for Light Gradient Boosting Machine. It is one of the fastest and highly accurate Gradient Boosting Framework. It can accurately handle huge amount of data without consuming more time. Thus, making it fast and accurate. It is observed that it is 4 times faster than the XGBoost. Which motivated us to implement this project with LightGBM Framework.

III) Proposed System

Our dataset contains a number of critical factors, and data mining is at the heart of our system. We first cleaned up the full dataset and trimmed the numbers that were outliers. Our data was made up of range values, therefore we took the mean of the housing prices in the database to convert them into a clearer format. Additionally, LightGBM does not accept object or string values. As a result, we had to assign each site a distinct address. Furthermore, we weighed each parameter based on its importance in determining the system's pricing, resulting in an increase in the value that each parameter retains in the system. One of the most important components of working with the LightGBM Framework is parameter adjustment.

1. Specifications of Dataset

TABLE I. SPECIFICATION OF DATASET

Field name	Value
Number of Rows	13320
Number of columns	8
Number of Numeric variables	7
Number of categorical variables	1
Type of problem	Regression
Missing Value	NIL
Choice of metric	Accuracy

The precision is entirely dependent on the parameters provided. We tried to figure out the best parameters we could to get the most accuracy out of it. 'Learning rate,''max depth,' 'num leaves,''max bin,' 'num iterations,' and so on were some of the most essential parameters employed. LightGBM uses the aforementioned parameters to shape the tree and conduct various operations on the data. We also utilised the Nave Bayes technique to determine whether the value entered by the user was legitimate or not. It also confirms the availability of a residence that meets the specified criteria.

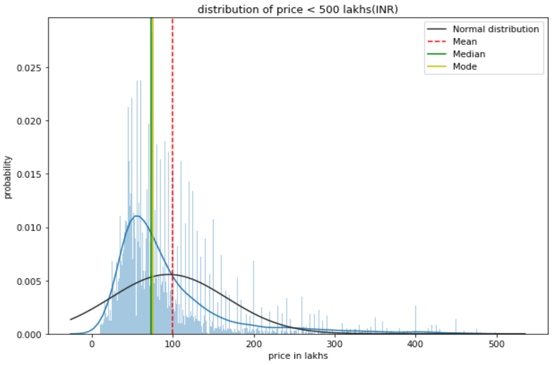


Figure 3: Visualizing distribution of price

The above figure describes the price distribution of houses in our dataset. It specifically focuses on the houses with price less than 500 lakhs because the maximum number of houses lie in this data range.

1. Algorithms Used:

- **Linear Regression**

One of the simplest and most widely used Machine Learning techniques is linear regression [6]. It's a statistical technique for performing predictive analysis. Sales, salary, age, product price, and other continuous/real or numeric variables are predicted using linear regression. A linear relationship exists between a dependent (y) and one or more independent (x) variables when using the linear regression algorithm.

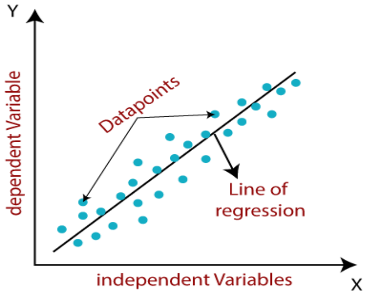


Figure 4: Representation of Linear Regression

In our project, the house price data from the dataset served as the dependent variable, while the remaining columns served as the independent variable. The dataset was separated into two halves with a 7:3 ratio. Seventy percent of the dataset was used to train the dataset, while the remaining thirty percent was used to test it. We attained an accuracy of 83.88 percent using Linear Regression on the dataset.

- **Random Forest Regressor**

A random forest [7] is a machine learning technique for solving classification and regression problems. It makes use of ensemble learning, which is a technique for solving complicated problems by combining several classifiers. Many decision trees make up a random forest algorithm. Bagging or bootstrap aggregation are used to train the 'forest' formed by the random forest method. Bagging is a meta-algorithm that increases the accuracy of machine learning methods by grouping them together.

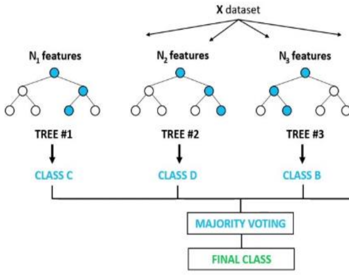


Figure 5: Representation of Radom Forest Regressor

Random forest training algorithm applies the technique of bootstrap aggregating, or bagging, to tree learners [7]. Given a training set *X* = *x1*, ..., *xn* with responses *Y* = *y1*, ..., *yn*, bagging repeatedly (*B* times) selects a random sample with replacement of the training set and fits trees to these samples:

For *b* = 1, ..., *B*:

1. Sample, with replacement, *n* training examples from *X*, *Y*; call these *Xb*, *Yb*.

2. Train a classification or regression tree *fb* on *Xb*, *Yb*.

After training, predictions for unseen samples *a'* can be made by averaging the predictions from all the individual regression trees on *a'*.

$$\hat{f} = \frac{1}{B}\sum_{b=1}^{B} f_b(x')$$

Figure 6: Formula Representation

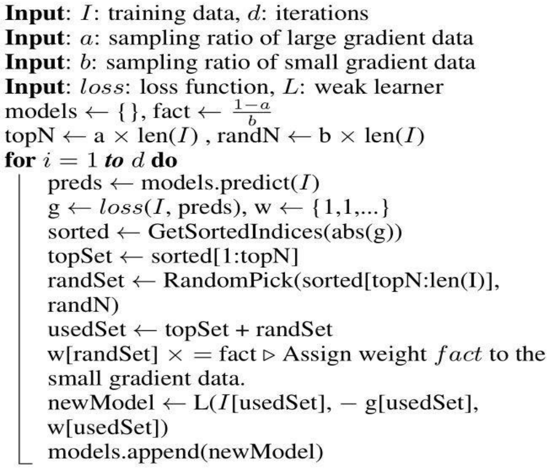
- **LightGBM**

LightGBM has been thoroughly examined and tested by Ke, Guolin et al. [8]. They offer a new Gradient boosting decision tree technique in this paper. Gradient-based OneSide Sampling and Exclusive Feature Bundling are two novel strategies used in this new algorithm. These strategies are capable of dealing with data characteristics and instances of enormous size. On these two new methodologies, these writers have conducted both experimental and theoretical research. The experiment results are consistent with the theory.

These results demonstrate that LightGBM can outperform classic algorithms like SGB and XGBoost in terms of memory consumption and computation speed using Gradient-based OneSide Sampling (GOSS) [8] and Exclusive Feature Bundling (EFB).

All of the data examples, as well as noteworthy gradients, are kept in GOSS. On samples with small gradients, this approach uses random sampling. The GBDT, on the other hand, maps every sample with a little gradient, has a small training error, and is well-trained. If we delete those useless data instance, we can improve accuracy.

The learnt model's accuracy may suffer as a result of this distribution change. The GOSS approach is used to solve this problem.



```
Input: $I$: training data, $d$: iterations
Input: $a$: sampling ratio of large gradient data
Input: $b$: sampling ratio of small gradient data
Input: $loss$: loss function, $L$: weak learner
models ← {}, fact ← $\frac{1-a}{b}$
topN ← a × len($I$) , randN ← b × len($I$)
for $i = 1$ to $d$ do
    preds ← models.predict($I$)
    g ← $loss$($I$, preds), w ← {1,1,...}
    sorted ← GetSortedIndices(abs(g))
    topSet ← sorted[1:topN]
    randSet ← RandomPick(sorted[topN:len(I)],
    randN)
    usedSet ← topSet + randSet
    w[randSet] × = fact ▷ Assign weight $fact$ to the
    small gradient data.
    newModel ← L($I$[usedSet], − g[usedSet],
    w[usedSet])
    models.append(newModel)
```

Figure 7: Parameters used in LightGBM

Because it is a flexible and effective technology, we chose the Light Gradient Boosting Machine. LightGBM is a framework for gradient boosting. A tree-based learning technique is used in this framework. As we all know, the amount of data collected is growing by the day, making traditional data science approaches increasingly difficult to deliver faster and more accurate outcomes. As its name implies, light is a fast-moving object. It has the ability to process massive amounts of data. It emphasises the output's efficiency as well as the speed of the process. The GPU learning function of LightGBM sets it apart from the competition.

This model has a number of parameters that we've provided. Some of the qualities are as follows:

The boosting type ='gbdt' is a gradient boosting decision tree. The goal 'binary' denotes that this technique will be used for binary classification. "n_estimators" this parameter determines the amount of change in the estimates. 'feature_fraction' the size of the change in the estimations is determined by this parameter. 'bagging_fraction' this parameter determines the size of the change in the estimations. learning rate = '0.07': this parameter controls the magnitude of the change in the estimations. The entire number of leaves in the tree is represented by num_leaves = 128.

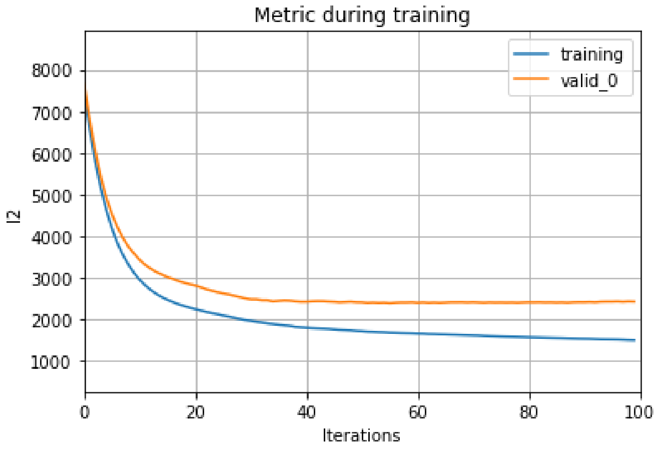


Figure 8: plot-metric in LightGBM

We have used plot-metric to plot learning curves in lightgbm. We trained our model after specifying all of the parameters. Finally, we've presented the testing set, which produced a range of 0 to 1 result. The model's accuracy was assessed here, and it was found to be 90.96 percent.

2. Working of System

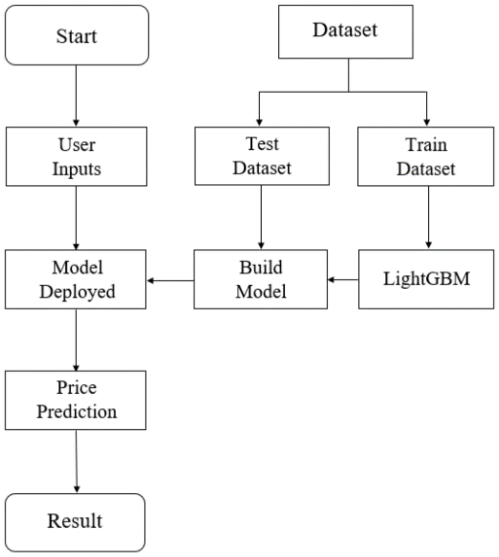


Figure 9: Representation of System

The operation of the system is represented in the diagram above. It includes both creation of model as well as prediction of house prices.

Model Generation:

Model generation step includes data cleaning, pre-processing and data splitting. Further the divided data is provided to the algorithm and thus the final build is generated. This generated model is created using a well-known python library known as Pickle. This library produces a pickle file with an extension ".pkl". This file is used for prediction. This file can be stored in the backend server database.

Price Prediction:

The prediction process includes the user providing their requirements through a form. These requirements are further passed to the backend using xmlhttprequest. The requirements are further provided as an input to the prediction file by creating a Data frame using python's pandas [9] library. It generates outcomes based on the user's input, which includes a projected house price. This result is then provided to the user again with the help of xmlhttprequest. A notification noting that the user's input is invalid is presented if the input is irrelevant or house with user's requirements is not available.

2. Result

All of the models have been implemented, and the following are the results:

TABLE II. ACCURACY OF THE MODELS.

Model Implemented	Accuracy
Logistic Regression	83.88%
Random Forest Regressor	72.4%
LightGBM	90.96%

TABLE III. RMSE values of different models.

Models Implemented	**RMSE**
Logistic Regression	37.64403
Random Forest Regressor	50.34817
LightGBM	43.998

We concluded that LightGBM is the optimal model for implementing our project based on the above results. Its accuracy was the highest of all the others, and its round mean square error was the lower than Random Forest Regressor.

IV) Conclusion

Implementing the LightGBM algorithm for the dataset provided. For the given dataset, we determined the algorithm's accuracy. The accuracy we obtained was around 90.96 percent. Along with the accuracy, the root mean square error was calculated, and it was found to be around 0.43. A round mean square error is generated by every machine learning algorithm. This inaccuracy has a significant impact on the outcome or prediction accuracy. LightGBM has the smallest error, making it the best algorithm for predicting housing prices. The user can either sell or buy a house after deciding the price. As a result, the user can estimate an actual price for his or her home based on his or her needs.

For each data tuple, LightGBM returns the likelihood. As a result, we can add more classes to the dataset to classify it according to need. We can forecast which houses are most and least likely to be purchased since we can categorise data tuples as most or least likely. Other industries, such as sales and marketing, can benefit from this concept. This approach can also be used to sort potential buyers who have asked about a specific product. The addition of additional cities to the database could be a key future improvement, allowing users to look at more residences, gain greater accuracy, and so make better decisions.

V) Future Work

The system's accuracy can be enhanced. If the system's size and processing capacity grow, it will be possible to add several more cities. A new constraint that focuses on the security of a specific site can be added. Which can be done with the help of the crime API. In addition, a learning system can be developed that collects user feedback and history so that the system can present the most appropriate results to the user based on his preferences.

VI) Acknowledgment

We would like to show our gratitude to Mrs. Linda John, Faculty, St. John College of Engineering and Management, for sharing their pearls of wisdom with us during this research work. We thank St. John College of Engineering and Management, Palghar, for providing the necessary information regarding the research.

VII) References

[1] J. Manasa, R. Gupta and N. S. Narahari, "Machine Learning based Predicting House Prices using Regression Techniques," 2020 2nd International Conference on Innovative Mechanisms for Industry Applications (ICIMIA), 2020, pp. 624-630.

[2] A. Varma, A. Sarma, S. Doshi and R. Nair, "House Price Prediction Using Machine Learning and Neural Networks," 2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT), 2018, pp. 1936-1939.

[3] B.Vijay Kumar, B.Ashritha, CH.Teja, M.Vineeth, "House Price Prediction Using Gradient Boost Regression Model", 2020 International Journal of Research and Analytical Reviews (IJRAR), March 2020, Volume 7, issue 1

[4] P. Durganjali and M. V. Pujitha, "House Resale Price Prediction Using Classification Algorithms," 2019 International Conference on Smart Structures and Systems (ICSSS), 2019, pp. 1-4.

[5] Y. Zhao, G. Chetty and D. Tran, "Deep Learning with XGBoost for Real Estate Appraisal," 2019 IEEE Symposium Series on Computational Intelligence (SSCI), 2019, pp. 1396-1401.

[6] N. N. Ghosalkar and S. N. Dhage, "Real Estate Value Prediction Using Linear Regression," 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA), 2018, pp. 1-5.

[7] The elements of statistical learning, Trevor Hastie - Random Forest Generation.

[8] G. e. a. Ke, "Lightgbm: A highly efficient gradient boosting decision tree," in 31st International Conference on Neural Information Processing Systems (ICNIPS), 2017, pp. 3146-3154.

[9] M. W, Python for data analysis: Data wrangling with Pandas, NumPy, and IPython., O'Reilly Media, Inc., 2012 Oct 8.

[10] D. Banerjee and S. Dutta, "Predicting the housing price direction using machine learning techniques," 2017 IEEE International Conference on Power, Control, Signals and Instrumentation Engineering (ICPCSI), 2017, pp. 2998-3000.

[11] L. Li and K. Chu, "Prediction of real estate price variation based on economic parameters," 2017 International Conference on Applied System Innovation (ICASI), 2017, pp. 87-90.