House Price Prediction Using Random Forest Regression

Dr. Mariappan A.K
Professor
Department of IT
Easwari Engineering College
Chennai - 89, India

Gayathri S
Department of IT
Easwari Engineering College
Chennai - 89, India

Janani B
Department of IT
Easwari Engineering College
Chennai - 89, India

Jhanani U
Department of IT
Easwari Engineering College
Chennai - 89, India

Abstract-

Owning house is always has been a minimum necessity of living. Buying a house is considered as an investment and a crucial decision in one's lifetime. There are numerous factors to be considered while buying a house. This paper demonstrates the house price prediction model on real dataset downloaded from Kaggle from Chennai created by Akash Kunwar. We have reviewed a few existing machine learning algorithms on the dataset and inferred drastic changes in the accuracy calculated.The optimal Random Forest Regression algorithm is selected for the prediction model using the aforementioned proof. Based on the interdependence of influencing factors, the model forecasts the price of a property.The model in turn provides the forecasted yearly price graph, predicted price per square feet, the forecasted price, etc. To make it convenient a user friendly web application is created for executing this model. This helps the buyer as well as seller to have some insights on property value and model prevents getting eluded by third party brokers.

Keywords- Machine Learning, Data Science, House price prediction, Feature Analysis, Data Extraction

I. INTRODUCTION

Research teams are increasingly using deep learning or machine learning models to carry out studies relating to the popular topic of house price forecasting.However, the results of some research do not always provide adequately accurate predictions since they do not take into account all relevant information that influences home values. So, for housing prediction, we suggest an end to end joint model. We present a machine learning-based model in this study to handle the complexity, taking into consideration the possibility that several factors may have unexpectedly intertwined effects on home prices. As previously stated, we think there are a variety of complex elements that influence property prices, and distinct traits may even carry different weights at different points in time. Diverse aspects of a home may be of interest to various buying groups. For their children, nuclear families, for instance, could choose parks and neighbouring schools. As a result, a residence can be recommended to a potential buyer if its key attributes, as determined by a mechanism, meet their needs. Finally, using tests, we show that the Random Forest model outperforms other conventional models.

Hence, our ultimate study objective is to enhance forecast accuracy and look into aspects impacting pricing outcomes. These data and the new methods enhance the accuracy of home price predictions while also exposing market influences. We not only anticipate house values accurately, but we also pinpoint the characteristics that influence them. As a result, we create an attention-based model that can learn how elements interact with one another and condense key characteristics for home buyers. In order to enhance model performance, we also develop fresh techniques for efficiently extracting features from diverse data. Due to population expansion and individuals relocating to other cities

for employment opportunities, there is a constant increase in the demand for housing on the market. For individuals who don't want to take any chances while purchasing their home, predicting the property prize on the open market for an extended period of time is essential.

II. LITERATURE REVIEW

A. P. Singh [1] research paper will assist clients in understanding the true cost of a home as well as builders in determining the selling price that will best meet client requirements using Random forest. The first round of data scraping will be necessary for the model we have created. A dataset of all the useful data can be created as a text file by employing the notion of data scraping, which makes it simple for users to discover or extract data from many sources.

In this study, predictions are made using a variety of regression techniques, such as ridge, LASSO, elastic net, gradient boosting, multiple linear, elastic net and ada boost regression *C. R. Madhuri* [2]. The effectiveness of each of the aforementioned methods has been tested on a data set in order to predict property prices. The goal of this essay is to help readers pinpoint the precise time frame for home purchasing and to help sellers estimate the selling price of a home precisely. Physical conditions, concept, location, and other associated aspects that affect cost were also taken into account.

Our study's main goal is to forecast housing values using actual factors. Here, we make an effort to base our evaluations on every fundamental aspect taken into account when determining pricing. We apply regression techniques in this strategy. No single strategy alone determines our outcomes., but rather by the weighted mean of many techniques, which provides the results that are the most accurate. The outcomes demonstrated that this strategy produces least error and highest accuracy when compared to applying individual algorithms. *A. Varma* [3] also suggest using Google Maps to achieve precise real-world valuations by leveraging real-time neighbourhood information.

There is a lot of interest in the possibility of using the synaptic memristor in artificial neural networks (ANNs). There aren't many reported real-world memristor-based network applications, though. *J. J. Wang et al* [4] develops a memristor-based ANN that trains a multi-variable regression model using the backward propagation method. Memristor-based weight unit circuit that may be programmed as either an excitatory or an inhibitory synapses is shown. The memristor's conductance determines the electronic synapse's weight, and the charge-dependent relationship governs the synapse's current.

A house's price is influenced by three things: its physical state, its design, and its location. Without taking into account market prices or cost growth, the existing framework covers assessing the cost of homes. *N. N. Ghosalkar*[5] goal is to forecast home pricing for clients while taking their requirements and financial objectives into account using Linear Regression. Future costs will be predicted by dissecting past market trends, price ranges, and upcoming technological improvements.

The available home properties on the machine hackathon platform were used to illustrate an analytical study proposed by *J. Manasa*[6]. Lasso and Ridge models, support vector regression, multiple linear regression (Least Squares), and boosting algorithms like Extreme Gradient Boost Regression are some of the regression techniques used in modelling explorations (XG Boost). By comparing the prediction errors produced by several models, these models are utilised to create a predictive model and select the best-performing model.

One of the concerns that has the people the most worried is the cost of housing. Along with negatively affecting the standard of living, excessive price growth in the housing market will also have an effect on the dynamics of the business cycle. Yet, many of the conventional house price prediction methods suggested by Y. Piao have a declining level of accuracy due to the complexity of the factors affecting residential real estate values and the ambiguity in the selection of advantageous attributes.[7]. A novel CNN-based prediction model is recommended in light of this for forecasting housing values as well as the feature selection process.

We compare the explanatory power, location information capacity, and forecast precision of these techniques with the traditional Hedonic Pricing Model. The *Y. Feng* [8] suggest that MLM has strong explanatory power and good predictive accuracy, particularly when neighbourhood impacts are investigated at various spatial scales.

D. Banerjee[9] researched the phenomenon of growing or declining housing values. Regression analysis has been utilised extensively in prior studies to answer the question of how housing prices vary. In order to categorise the issue of fluctuating home prices as a classification problem and predict whether they would rise or decline, this study uses machine learning.

A. J. Bency [10] presented a Convolutional Neural Network (CNN) framework in this study to automatically identify spatial connections and

model geographical data, specifically housing prices. We demonstrate how the needed spatial smoothing can be achieved by utilising neighbourhood data that is embedded in satellite imagery. For the cities of London, Birmingham, and Liverpool, we demonstrate a significant improvement of 57% over the SAR baseline by applying features from deep neural networks.

III. PROPOSED SYSTEM

The proposed framework is built using a variety of model-building techniques, including the highly accurate Random Forest, Linear Regressor, and XG Boost regressor SVM algorithms. The initial step is to collect the dataset from Kaggle website. Then the preprocessing work of the dataset occurs to make sure the model can be trained without any issues.To guarantee that every row and column in the dataset is appropriately prepared and preprocessed, the proposed system does extensive exploratory data analysis. We divided the model into two sections to test and train the dataset once the dataset had been cleaned and preprocessed. They are split in the ratio 7:3, the split is used for training , where the latter split is used for the testing of the dataset. After the splitting the model is trained using the desired Machine learning algorithms .The accuracy of these algorithms are used to choose the best models. This ensure that maximum accuracy is obtained. The suggested system outperforms the current approaches in terms of precision and efficiency.
The workflow of the entire project is diagrammatically represented in the Fig 1.1.

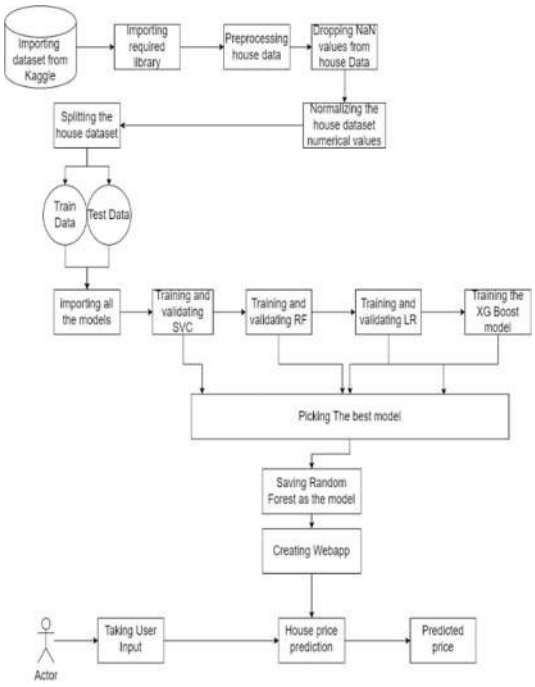

Fig 1.1 Architecture diagram

IV. THE PROPOSED SYSTEM :

A. Data gathering and preprocessing

The first stage is data collecting, which may be done using Kaggle. The dataset for the research should be quite effective in predicting the price of a home. The gathering of data for the ML model's training is the core step in the machine learning pipeline. The quality of the training data determines how accurately ML systems can predict the future. Preprocessing can be used to enhance a dataset's precision and quality, which also increases the dataset's dependability. It guarantees the consistency of the data. The raw data is converted into a format that can be used effectively and efficiently using a data mining approach referred to as "data preparation."

Preprocessing alters the format of the data in order to speed up and simplify data mining, machine learning, and other data science procedures.
To ensure precise results, the techniques are frequently deployed right at the beginning of the machine learning and AI development pipeline. Although there are many additional measures that could have been utilised, we have chosen the following ones for our dataset.

PRT_ID	AREA	INT_SQFT	DIST_MAIN	N_BEDROO	N_BATHRO	N_ROOM
P03210	Karapakkan	1004	131	1	1	3
P09411	Anna Nagar	1986	26	2	1	5
P01812	Adyar	909	70	1	1	3
P05346	Velachery	1855	14	3	2	5
P06210	Karapakkan	1226	84	1	1	3
P00219	Chrompet	1220	36	2	1	4
P09105	Chrompet	1167	137	1	1	3
P09679	Velachery	1847	176	3	2	5
P03377	Chrompet	771	175	1	1	2
P09623	Velachery	1635	74	2	1	4
P09540	Chrompet	1203	78	2	1	4
P07121	Chrompet	1054	143	1	1	3
P05512	Chrompet	1196	137	1	1	3
P09370	Adyar	1056	83	1	1	3
P04085	Velachery	1865	157	3	2	5
P06328	Velachery	1868	148	3	2	5
P06039	Karapakkan	1639	175	2	2	4
P02016	Chrompet	796	134	1	1	2
P08160	Adyar	1136	69	1	1	3
P01372	Anna Nagar	1902	168	2	1	5
P00936	Chrompet	1069	53	1	1	3
P07346	Chrompet	931	96	1	1	3
P06851	KK Nagar	2010	114	3	2	5
P04665	Chrompet	1074	100	1	1	3
P00902	TNagar	1972	111	2	1	5
P00293	T Nagar	1685	105	1	1	4
P02235	Adyar	1130	29	1	1	3
P04080	Karapakkan	1301	59	1	1	3

Fig 1.2 Snippet of the dataset

There are several ways in which the present Kaggle dataset is lacking. This could be the result of inadequate data collection techniques or a lack of pertinent data. These values in the dataset are represented by NaN (not a number) or None. Regardless of the causes, this makes our computing more difficult and distorted. As a result, we locate the missing data and swap it out for fully functional components.Pandas treats None and NaN as equivalent representations of missing or null values. Pandas DataFrame has a number of useful utilities that make finding, deleting, and modifying null values more easier. The functions isnull() and notnull are used in Pandas DataFrame to recognise null values (). Any one of these routines can be used to check whether a number is NaN. Another potential use of these techniques when working with Pandas Series is finding the missing values in a series.

Most machine learning algorithms cannot function without first having their category input translated into numerical data. Some of the columns in our datasets contain string values that can be interpreted as category attributes. For instance, male, female, married, and single will all be options for the marital status criterion, and the same is true for the gender criterion. Machine learning models incorrectly assumed a hierarchy in the labels since the data is composed of string labels, even if the labels do not follow any particular order of preference.Label encoding is a potential method for dealing with this issue. Here, we'd use numeric labels like "male" and "female," for example, to indicate their distinct genders.

First, we take care of the values that are missing. The data set has a bothersome issue with missing values, often known as NaNs, which stands for "not a number." We get rid of any and all nan values that were found in the columns. Second, in order to simplify the computations and reduce the amount of error in the model, we standardize the values of the dataset. The term "normalization" refers to the act of altering the data, more specifically translating the data at its source into another format that is more conducive to efficient data processing. Third, in order to feed the model, we convert the category labels, which are either "Yes" or "No," into numerical values, which are either "1" or "0." Models used in machine learning are almost always constructed using mathematical equations. Because of this, we are able to comprehend, on an instinctual level, that include the categorical data in the equation will result in certain problems, given that the equations themselves require only numerical values. Finally, we divided the data set into two halves in order to train and validate our model.

Once the null values have been located, the rows and columns holding the null values can be eliminated or changed to the mean, median, and mode. To remove every null from a dataset, we utilised the dropna() method. This operation can be used to remove null values from table columns and rows. The mean or median value can be used to replace the missing data if the relevant columns have integer or float data types.The value or mode that occurs most frequently could be used in place of another value in the absence of specific information. This can use both floating-point numbers and integers. But if the relevant columns also have strings, the usefulness rises.

B. Feature analysis

To enhance performance, certain missing data or data that, from the standpoint of data analysis, is not suitable for modelling can be found and eliminated during the preprocessing stage. Additionally, knowing specific statistical measurements can help a data scientist or expert decide quickly which columns to include in a given case's Machine Learning model.

This feature analysis algorithm was created with this objective in mind. It divides the input columns into three types—Identifier, Numerical (Int/Float) Feature, and Categorical Feature—and then presents crucial statistical data that can be used in future analysis. Additionally, it shows how the columns relate to one another so that unnecessary ones can be removed.

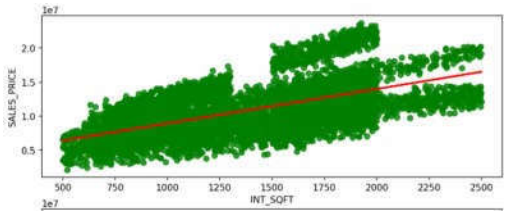

Fig 1.3 Square feet vs Sales price

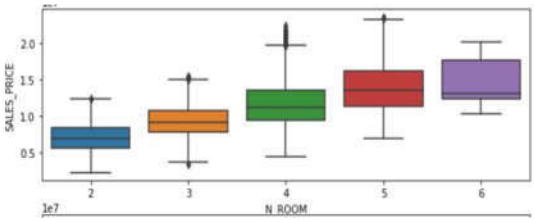

Fig 1.4 No of rooms vs Sales Price

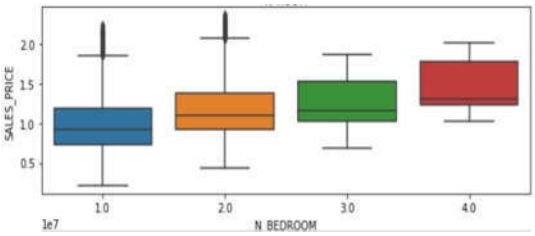

Fig 1.5 No of bedrooms vs Sales Price

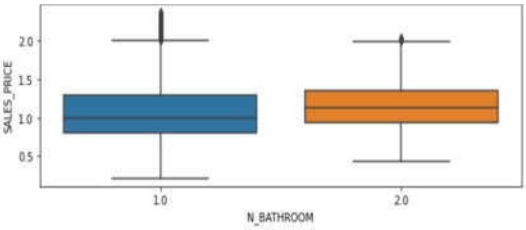

Fig 1.6 No of Bathrooms vs Sales Price

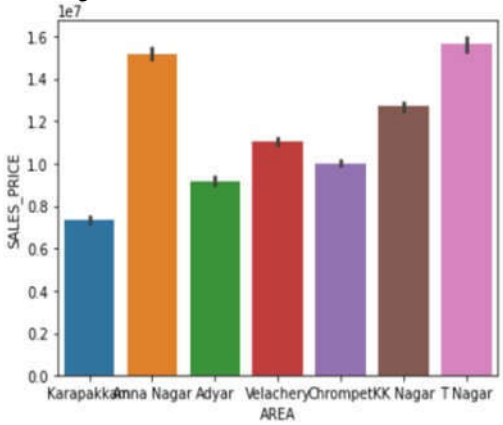

Fig 1.7 Area vs Sales Price

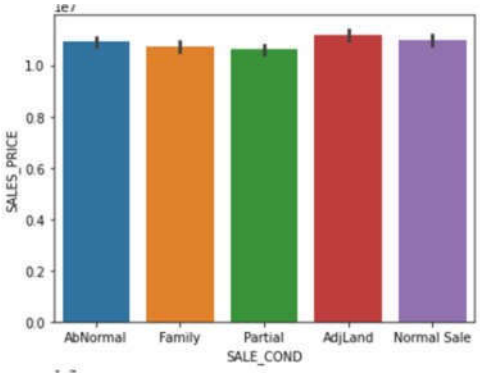

Fig 1.8 Sales Condition vs Sales Price

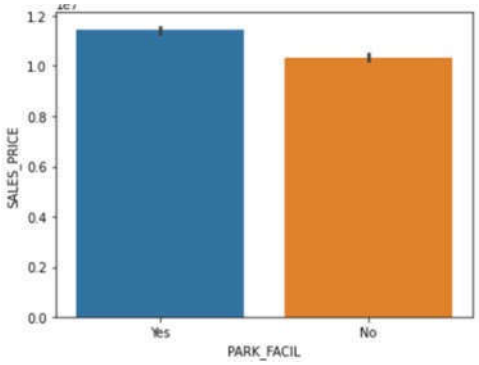

Fig 1.9 Parking Facility vs Sales Price

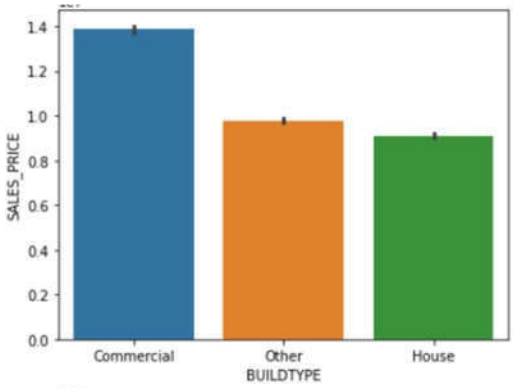

Fig 1.10 Building Type vs Sales Price

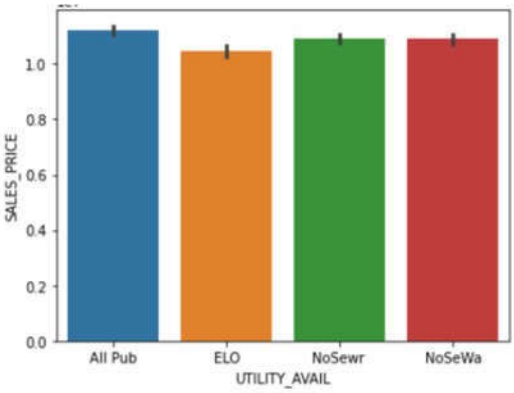

Fig 1.11Utilities Available vs Sales Price

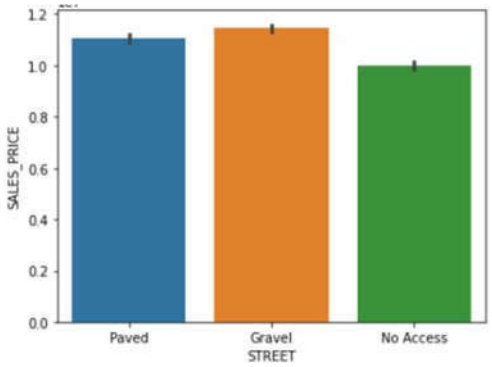

Fig 1.12 Street vs Sales Price

The following Fig 1.3 - Fig 1.12 contains features which affect the prediction value. Area, Square feet price, sales condition and parking facility, these factors will affect the prediction of the house.

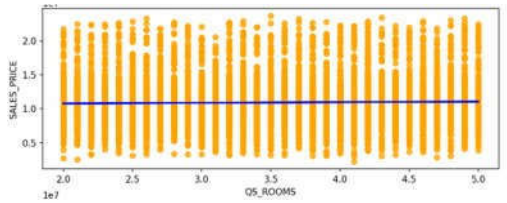

Fig 1.13 QS Rooms vs Sales Price

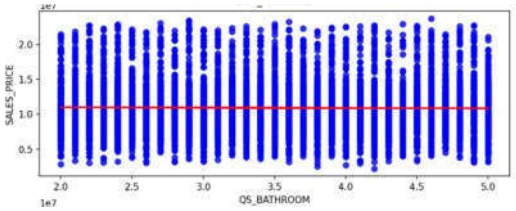

Fig 1.14 QS Bathroom vs Sales Price

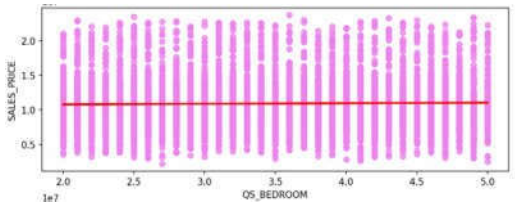

Fig 1.15 QS Bedroom vs Sales Price

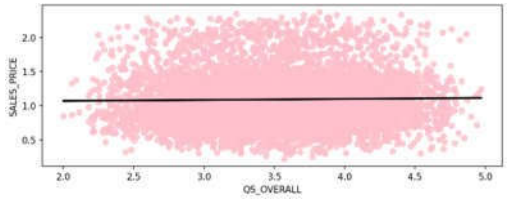

Fig 1.16 QS Overall vs Sales Price

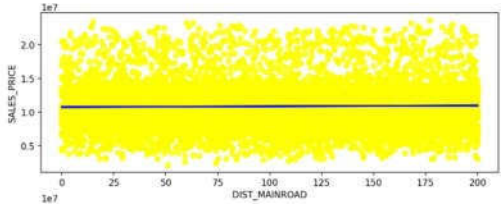

Fig.1.17 Distance from Main road vs Sales Price

The above Fig 1.13 - Fig 1 contains the features such as distance from main road, masking values of rooms and bathrooms which are not used in the prediction of the results and have no significance role to be played.

C. Training the model and result evaluation

Model training in machine learning is the process of feeding an algorithm with data to help it learn and choose optimal values for all linked attributes. The train-test split is used to evaluate how well prediction-based methods and applications employ machine learning algorithms. Using this simple technique, we may compare the predictions of our own machine learning model against those of other machines. The Test set is only guaranteed to have 30% real data, compared to the Training set's 70% assurance of raw data. A dataset can be split into train and test sets, allowing us to assess how well our machine learning model is doing. These little characters from the train set are recognised by the model and used to good advantage. The "test data set," the second batch of data, is only utilised for projections.

We use the next technique to split up our dataset into train and test sets. The Pandas and Sklearn software programmes are introduced. Sklearn is the best and most dependable machine learning package for Python. For instance, the splitter function train test split is included in the model selection module of the Scikit-Learn package (). The read csv() function is then used to import the CSV file. The df variable now contains the data frame. Then, we employ a test size of 0.3, which reserves 70% of our data for training and 30% of our data for evaluation. To ensure an equal distribution of records between the two collections, we also set random state=0.

We will start by importing the required modules. A Python module can access the source code that is included in another module by importing a file or function. After importing these modelling strategies, we will train and evaluate Linear Regression, Random Forest, and XGBoost on our data. We will then train our models after that. The industry standard for predictive analysis is linear regression. Predictions are made in light of the linear relationships between the objective and one or more independent factors . The first model, a linear regression model, will be trained initially. The

model will next be put to the test to determine its accuracy. The second model, a Random Forest Model, will then be trained, and its accuracy will be tested afterward.

The (random forest) algorithm decides the outcome as the decision trees make their forecasts. By averaging the outcomes of numerous trees, it generates forecasts. As the number of trees increases, the prediction becomes more accurate. The third model, an XGBoost model, will be trained last, after which we will test it to determine its accuracy. Both a graphics processing unit (GPU) predictor and a central processor unit (CPU) predictor are used by XGBoost. The default setting is auto, which enables XGBoost to utilise some heuristics for sparing GPU RAM during training. The best model is chosen based on some parameters like R2 score, MSE, RMSE, MAE, MAPE.

Using the R2 score as a measure,

$$R2 = 1 - SSres / SStot \quad (1)$$

The square root of the residual errors is denoted by SSres.
The total sum of the errors is known as SStot.
The value obtained for random forest regressor is 0.990356 (1). The former is the highest among all the other equations compared.

The MSE, RMSE, MAE, MAPE are calculated using

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2 \quad (2)$$

$$\mathrm{RMSD} = \sqrt{\frac{\sum_{i=1}^{N}(x_i - \hat{x}_i)^2}{N}} \quad (3)$$

$$\mathrm{MAE} = \frac{\sum_{i=1}^{n}|y_i - x_i|}{n} \quad (4)$$

$$M = \frac{1}{n}\sum_{t=1}^{n}\left|\frac{A_t - F_t}{A_t}\right| \quad (5)$$

The former formulas are used to calculate the error and the outcomes are recorded in the table Fig. 1.18.

ALGORITHM	R²SCORE	MSE	RMSE	MAE	MAPE	PREDICTION TIME
Linear Regression	0.910872	1.100110e+12	1.048861e+06	8.373689e+05	0.089101	0.000656
Support Vector Machine	-0.021866	1.261295e+13	3.551471e+06	2.705817e+06	0.283017	0.814266
Random Forest Regressor	0.990356	1.190331e+11	3.450118e+05	2.722876e+05	0.030754	0.043023
XGBoost	0.944554	6.843750e+11	8.272696e+05	6.722235e+05	0.075763	0.007191

Fig.1.18 Error Values Comparison Table

Random forest regression is finalized based on the performance analysis. As a result, "k" features are randomly selected from a total of "m" features to provide the pseudocode for random forest regression. To identify the node "d" among the "k" features, decide the best split point and when to apply it. Repeat the preceding technique up to the "l" number of nodes, using the optimum split to split the node into daughter nodes.To produce a "n" number of trees, repeat steps through a "n" number of times.Starting with a random selection of "k" features from a total of "m," the random forest algorithm begins. The graphics of selected characteristics and observations at random are analyzed.

The model is trained and fitted in accordance with random forest regression. The model is trained and tested using high impact features. The accuracy of the prediction is evaluated. The evaluated sample of actual and predicted price is attached as Fig 1.19

	PRT_ID	AREA	PRICE_ACTUAL	PRICE_PREDICTION
2860	P04737	KK NAGAR	13178340	13390064.00
107	P06329	CHROMPET	7703200	7360700.00
2249	P05327	KARAPAKAM	4894125	4525256.50
2802	P05488	KARAPAKAM	8507750	8466581.00
5824	P06292	ADYAR	6036800	5604670.50
3481	P05425	KARAPAKAM	8090500	7718704.00
6218	P05725	KARAPAKAM	3267125	3497543.25
2272	P07500	CHROMPET	13328250	13433819.00
6099	P03512	KARAPAKAM	7629750	7780697.00
3050	P010032	VELACHERY	6307030	5452638.00
5074	P09034	ANNA NAGAR	19897420	19587430.00
4001	P02034	KARAPAKAM	10095000	10789617.00
1652	P09005	CHROMPET	8537500	8025007.00
6243	P04691	KARAPAKAM	4577750	4711120.50
4140	P06696	KK NAGAR	10056300	9674508.00
5825	P03290	CHROMPET	10145540	10066949.00
5501	P01961	KK NAGAR	8876760	8838347.00
2603	P04503	KARAPAKAM	7928625	7869017.00
6306	P05961	ADYAR	9012665	7919560.00
5936	P08780	CHROMPET	8383570	8724420.00

Fig.1.19 Predicted Price Comparison Table

In the end, we'll use the web application's best model. It is clear from the table in Fig. 1.19 that Random

Forest is the best model for more accurate and efficient home price prediction. Once random forest is chosen as the best model to predict the result they are used as the model to predict the housing price. The model analyses to find the important factors that contribute towards the price. Also, it gives a forecast range (range of permissible values) rather than a single estimate.

D. Implementing the system in a WebApp

Sharing our developed machine learning model with others is an essential component of the process. No matter how many models we produce, very few people will be able to see what we are accomplishing if they remain offline. We should therefore make our models available so that anyone can use them through a beautiful User Interface (UI). We use Flask as the system's UI to construct a single-page web application for this system. It will accept information and make a prediction about the user's likelihood of developing chronic heart disease in ten years. Python-based Flask is a microweb framework. As it doesn't need any particular tools or libraries, it is referred to as a microframework. It lacks any component where already-existing third-party libraries would normally provide common functionalities, such as a database abstraction layer, form validation, etc.

V. RESULTS

The price of a house in a specific neighbourhood is predicted using machine learning methods such the Support Vector Machine, Linear Regressor, XG Boost Regressor, and Random Forest Regressor models. As a result of using several algorithms and selecting the best one, good prediction accuracy is seen. The experimental findings show that the suggested model outperforms the other models and achieves a low prediction error. The final prediction will be displayed to the user as shown the Fig 1.20

Fig.1.20 WebApp Output

VI. CONCLUSION

We develop a trustworthy housing prediction model in this work. Heterogeneous data is incorporated into the model to complete the information about the house, and we offer an attention method to automatically assign weights based on various traits or samples.

Therefore, it is necessary to anticipate real estate customers' preferences and financial constraints while setting prices.This research effectively forecasts future pricing by analysing historical industry trends and price ranges. The proposed solution outsmarts the existing system in terms of the accuracy and is proved to be the best model as it chooses the best available Machine Learning models.

REFERENCES:

[1] P. Singh, K. Rastogi and S. Rajpoot, 3rd International Conference on Advances in Computing, Communication Control and Networking - House Price Prediction Using Machine Learning.

[2] R. Madhuri, G. Anuradha and M. V. Pujitha, 2019 International Conference on Smart Structures and Systems - House Price Prediction Using Regression Techniques: A Comparative Study.

[3] Varma, A. Sarma, S. Doshi and R. Nair, 2018 Second International Conference on Inventive Communication and Computational Technologies - House Price Prediction Using Machine Learning and Neural Networks.

[4] J. J. Wang et al., IEEE Access - Predicting House Price With a Memristor-Based Artificial Neural Network.

[5] N. N. Ghosalkar and S. N. Dhage, 2018 Fourth International Conference on Computing Communication Control and Automation - Real Estate Value Prediction Using LR.

[6] J. Manasa, R. Gupta and N. S. Narahari, International Conference on Innovative Mechanisms for Industry Applications - ML based Predicting House Prices using Regression.

[7] Y. Piao, A. Chen and Z. Shang, International Conference on Information Science and Technology - Housing Price Prediction on CNN.

[8] Y. Feng and K. Jones, 2nd IEEE International Conference on Spatial Data Mining and Geographical Knowledge Services - Comparing multilevel modelling and artificial neural networks in house price.

[9] Banerjee and S. Dutta, IEEE International Conference on Power, Control, Signals and Instrumentation Engineering - Predicting the housing price direction using machine learning.

[10] J. Bency, S. Rallapalli, R. K. Ganti, M. Srivatsa and B. S. Manjunath, IEEE Conference on Applications of Spatial Models: Predicting Housing Prices with Satellite Imagery.