

House Price Prediction Using Machine Learning

Mr. Pathipati Chandrasekhar¹, Vishwasree Sunkara², Shettiwar Bhavitha³, Tulluru Mounika⁴

¹ Assistant Professor Computer Science Engineering & Guru Nanak Institute of Technology

² Student Computer Science Engineering & Guru Nanak Institute of Technology

³ Student Computer Science Engineering & Guru Nanak Institute of Technology

⁴ Student Computer Science Engineering & Guru Nanak Institute of Technology

Abstract - The rapid growth of the real estate sector has increased the demand for accurate house price prediction systems. This research presents a machine learning-based approach for predicting house prices using the Random Forest Regressor algorithm. The proposed model utilizes a dataset containing 42,703 housing records with features such as number of bedrooms, bathrooms, square footage, garage type, location, and amenities. Data preprocessing techniques including feature selection, categorical data conversion, and normalization were applied to improve prediction accuracy. The dataset was divided into 80% training data and 20% testing data. Experimental results demonstrated that the Random Forest Regressor achieved high prediction accuracy with a Mean Absolute Error of 1.4606 on the training set and 3.74 on the testing set. The system provides reliable predictions that can assist buyers, sellers, investors, and real estate professionals in making informed decisions. The proposed approach outperformed traditional regression methods due to its ability to handle large datasets, reduce overfitting, and identify feature importance effectively.

Key Words: Machine Learning, House Price Prediction, Random Forest Regressor, Real Estate, Data Mining, Python.

1.INTRODUCTION

House price prediction has become an important research area in the field of machine learning and data analytics. Accurate prediction of property prices helps customers, investors, and real estate organizations make better financial decisions. Traditional estimation methods depend heavily on human expertise and market assumptions, which may lead to inaccurate results. Machine learning techniques can analyze large volumes of historical housing data and identify patterns that

influence property prices. In this research, a Random Forest Regressor model is proposed for predicting house prices. The system uses multiple property features such as area, number of bedrooms, bathrooms, garage facilities, and city information to estimate prices accurately. Python programming language and machine learning libraries were used for implementation.

2.SCOPE OF THE PROJECT

The scope of the project is to create a comprehensive system capable of predicting property prices based on a multitude of features that influence real estate valuations. These features encompass a wide range, including property size, location, amenities, bedrooms, bathrooms, and more. Through careful data collection, preprocessing, and feature engineering, the project strives to transform raw property data into meaningful input for the predictive model.

3.OBJECTIVE

The primary objective of the project is to build a predictive model that accurately estimates property prices. The project aims to achieve this by:

Developing a robust and accurate Random Forest Regressor-based predictive model.

Ensuring the model's ability to generalize well to new and unseen property data. Providing an intuitive user interface that allows users to input property features and obtain estimated prices.

4.EXISTING SYSTEM

Gradient Boosting Regressor is a machine learning algorithm that falls under the ensemble learning category. It builds a predictive model in the form of an ensemble of weak learners, typically decision trees, to make accurate predictions.

The idea behind gradient boosting is to sequentially add new models to correct the errors of the existing ensemble. The algorithm minimizes a loss function, such as mean squared error for regression problems, by iteratively fitting new models to the residuals.

5. EXISTING SYSTEM DISADVANTAGES

Gradient Boosting Regressors, especially when using a large number of trees or deep trees, can be computationally expensive and time-consuming.

Gradient Boosting models, if not properly tuned, are susceptible to overfitting.

Gradient Boosting Regressors have several hyperparameters that need to be tuned for optimal performance.

6. LITERATURE REVIEW

Title: Time-Aware Latent Hierarchical Model for Predicting House Prices

Author: Tan F, Cheng C, Wei Z

Year: 2017.

Description: It is widely acknowledged that the value of a house is the mixture of a large number of characteristics. House price prediction thus presents a unique set of challenges in practice. While a large body of works are dedicated to this task, their performance and applications have been limited by the shortage of long time span of transaction data, the absence of real-world settings and the insufficiency of housing features. To this end, a time-aware latent hierarchical model is introduced to capture underlying spatiotemporal interactions behind the evolution of house prices. The hierarchical perspective obviates the need for historical transaction data of exactly same houses when temporal effects are considered. The proposed framework is examined on a large-scale dataset of the property transaction in Beijing. The whole experimental procedure strictly complies with the real-world scenario. The empirical evaluation results demonstrate the outperformance of our approach over alternative competitive methods.

Title: Predicting the housing price direction using machine learning techniques

Author: D. Banerjee and S. Dutta

Year: 2017.

Description: The phenomenon of the falling or rising of the house prices has attracted interest from the researcher as well as many other interested parties. There have been many previous research works that used various regression techniques to address the question of the changes house price. This work considers the issue of changing house price as a classification problem and applies machine learning techniques to predict whether house prices will rise or fall. This work applies various feature selection techniques such as variance influence factor, Information value, principle component analysis and data transformation techniques such as outlier and missing value treatment as well as box-cox transformation techniques. The performance of the machine learning techniques is measured by the four parameters of accuracy, precision, specificity and sensitivity. The work considers two discrete values 0 and 1 as respective classes. If the value of the class is 0 then we consider that the price of the house has decreased and if the value of the class is 1 then we consider that the price of the house has increased.

Title: Multivariate regression modelling for home value estimates with evaluation using maximum information coefficient.

Author: G. Hu, J. Wang and W. Feng

Year: 2013.

Description: Predictive modelling is a statistical data mining approach that builds a prediction function from the observed data. The function is then used to estimate a value of a dependent variable for new data. A commonly used predictive modelling method is regression that has been applied to a wide range of application domains. In this paper, we build multivariate regression models of home prices using a dataset composed of 81 homes. We then applied the maximum information coefficient (MIC) statistics to the observed home values (Y) and the predicted values (X) as an evaluation of the regression models. The results showed very high strength of the relationship between the two variables X and Y.

Title: Twitter Sentiment Analysis Using Naive Bayes based Machine learning Technique

Author: Priya Gour, Sudhanshu Vashistha and Pradeep Jha

Year: 2023

Description: "Computational" sentiment analysis can determine whether a sentiment is favourable, negative, or neutral. Another term for this approach is "opinion mining," or obtaining a speaker's sentiments. Businesses

use it to develop strategies, learn what customers think about products or brands, how people react to campaigns or new product releases, and why they do not buy certain products. It is used in politics to keep track of political ideas and to check for contradictions between government claims and actions. It can even be used to predict election results! It is also used to track and analyze social phenomena like recognizing dangerous circumstances and evaluating blogging mood. In this paper, we look tackle the problem of sentiment categorization using the Twitter dataset. To analyze sentiment, preprocessing and Naive Bayes classifier approaches are utilized. As a result, we applied a text preprocessing classification accuracy classifying strategy and improved our classification accuracy score on the Kaggle public leaderboard. The aim of this paper is to classify the twitter sentiments using machine learning algorithm based on Naïve Bayes Classifier. The proposed model indicated better accuracy and precision based on performance parameters such as precision, recall and accuracy.

Title: Predicting House Price with a Memristor-Based Artificial Neural Network

Author: J. J. Wang et al

Year: 2018.

Description: Synaptic memristor has attracted much attention for its potential applications in artificial neural networks (ANNs). However useful applications in real life with such memristor-based networks have seldom been reported. In this paper, an ANN based on memristors is designed to learn a multi-variable regression model with a back-propagation algorithm. A weight unit circuit based on memristor, which can be programmed as an excitatory synapse or inhibitory synapse, is introduced. The weight of the electronic synapse is determined by the conductance of the memristor, and the current of the synapse follows the charge-dependent relationship. The ANN has the ability to learn from labelled samples and make predictions after online training. As an example, the ANN was used to learn a regression model of the house prices of several Boston towns in the USA and the predicted results are found to be close to the target data.

7.PROPOSED SYSTEM

Random Forest is a robust ensemble learning algorithm that leverages the strength of multiple decision trees. By constructing a diverse set of trees through techniques like bagging and random feature selection, it achieves

higher predictive accuracy and resilience compared to individual trees.

Known for its versatility, Random Forest is applicable to both classification and regression tasks. Its ability to handle high-dimensional datasets, mitigate overfitting, and provide insights into feature importance makes it a popular and effective choice across various domains.

8.PROPOSED SYSTEM ADVANTAGES

Random Forests often provide high accuracy in classification tasks.

Random Forests can handle large datasets with a large number of features.

They are computationally efficient and can parallelize the training process, making them suitable for big data applications.

9. METHODOLOGY

The proposed system follows several stages including data collection, preprocessing, feature engineering, model training, and evaluation. The dataset was obtained from Kaggle and consists of 42,703 records with 20 attributes. Features such as year built, stories, bedrooms, bathrooms, livable square feet, garage type, fireplace availability, swimming pool, and city were selected for prediction. Categorical features were converted into numerical values. Missing values and outliers were handled during preprocessing. The Random Forest Regressor algorithm was selected because it combines multiple decision trees to improve prediction performance and reduce overfitting. The dataset was split into training and testing datasets in an 80:20 ratio.

10.TECHNIQUE USED OR ALGORITHM USED EXISTING TECHNIQUE

Gradient Boosting Regressor

Gradient Boosting Regressor is a boosting algorithm that builds an additive model in a forward stage-wise fashion. It combines the predictions from multiple weak learners to create a strong predictive model.

The algorithm minimizes the loss function by adjusting the parameters of each weak learner during each iteration, emphasizing areas where the current model makes errors.

11. PROPOSED TECHNIQUE USED OR ALGORITHM USED

Random Forest Regressor

Random Forest is a machine learning algorithm that builds a multitude of decision trees during training, creating an ensemble model for classification or regression tasks.

It operates by employing two key sources of randomness: bagging (bootstrap aggregating) and random feature selection.

Bagging involves training each tree on a different subset of the training data, sampled with replacement, while random feature selection considers only a subset of features at each split in the decision trees.

12. HARDWARE REQUIREMENTS

The hardware requirements may serve as the basis for a contract for the implementation of the system and should therefore be a complete and consistent specification of the whole system. They are used by software engineers as the starting point for the system design. It should what the system do and not how it should be implemented.

- PROCESSOR : Intel i5.
- RAM : 4GB
- DD RAM
- HARD DISK : 250 GB

13. SOFTWARE REQUIREMENTS

The software requirements document is the specification of the system. It should include both a definition and a specification of requirements. It is a set of what the system should do rather than how it should do it. The software requirements provide a basis for creating the software requirements specification. It is useful in estimating cost, planning team activities, performing tasks and tracking the teams and tracking the team's progress throughout the development activity.

OPERATING SYSTEM : WINDOWS 7/8/10
PLATFORM : SPYDER3
PROGRAMMING LANGUAGE : PYTHON
FRONT END : SPYDER3
WEB FRAMEWORK : FLASK

14. FUNCTIONAL REQUIREMENTS

A functional requirement defines a function of a software-system or its component. A function is described as a set of inputs, the behaviour, The project's user interface streamlined the process of obtaining property price predictions. Users were empowered to input property features through a user-friendly interface, and the system responded with estimated prices based on the learned patterns from the training data. This accessibility made the system applicable to a wide range of users, including buyers, sellers, and real estate professionals.

15. NON-FUNCTIONAL REQUIREMENTS

The major non-functional Requirements of the system are as follows

Usability

The system is designed with completely automated process hence there is no or less user intervention.

Reliability

The system is more reliable because of the qualities that are inherited from the chosen platform python.

The code built by using python is more reliable.

Performance

This system is developing in the high level languages and using the advanced back-end technologies it will give response to the end user on client system with in very less time.

Supportability

The system is designed to be the cross platform supportable. The system is supported on a wide range of hardware and any software platform, which is built into the system.

Implementation

The system is implemented in web environment using Jupyter notebook software. The server is used as the intelligence server and windows 10 professional is used as the platform. Interface the user interface is based on Jupyter notebook provides server system.

16. DESIGN

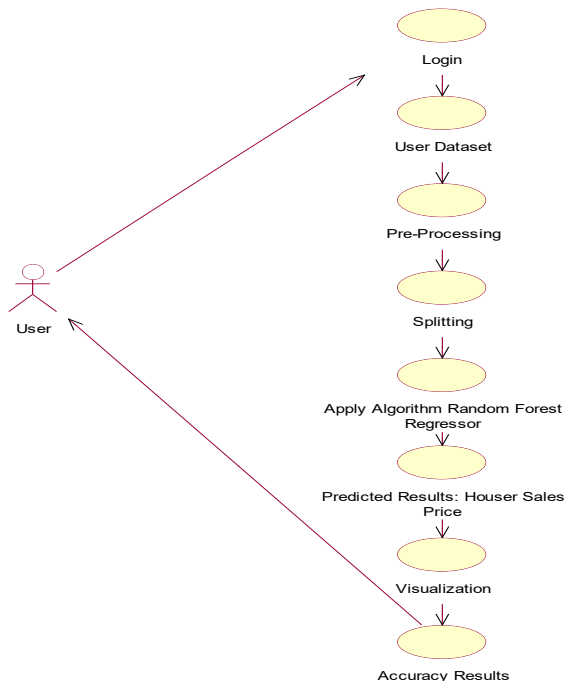
Design Engineering deals with the various UML [Unified Modelling language] diagrams for the implementation of project. Design is a meaningful engineering representation of a thing that is to be built. Software design is a process through which the requirements are translated into representation of the

software. Design is the place where quality is rendered in software engineering.

UML DIAGRAMS

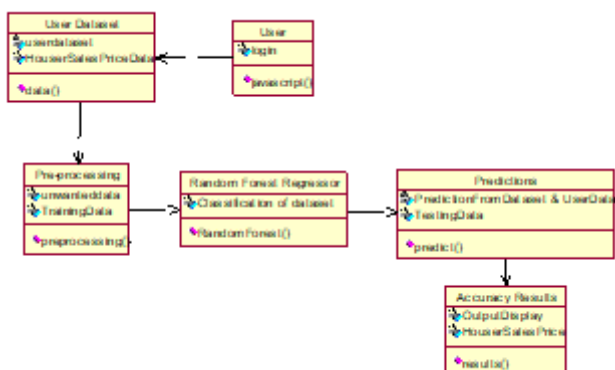
16.1 USE CASE DIAGRAM

The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted. The above diagram consists of user as actor. Each will play a certain role to achieve the concept.



16.2 CLASS DIAGRAM

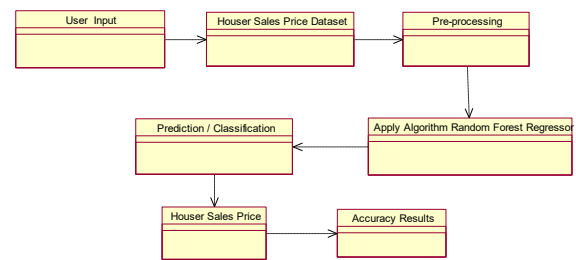
In this class diagram represents how the classes with attributes and methods are linked together to perform the verification with security. From the above diagram shown the various classes involved in our project.



16.3 OBJECT DIAGRAM

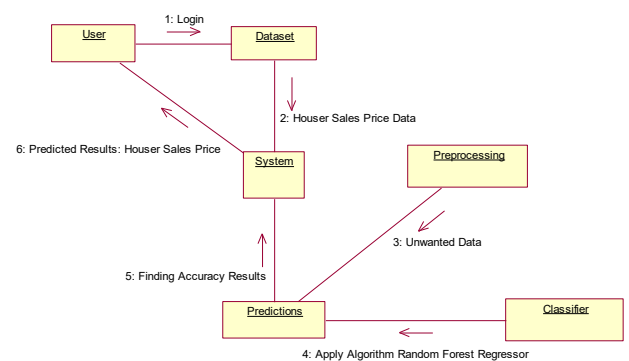
In the above diagram tells about the flow of objects between the classes. It is a diagram that shows a complete or partial view of the structure of a modelled system. In this object diagram represents how the classes

with attributes and methods are linked together to perform the verification with security.



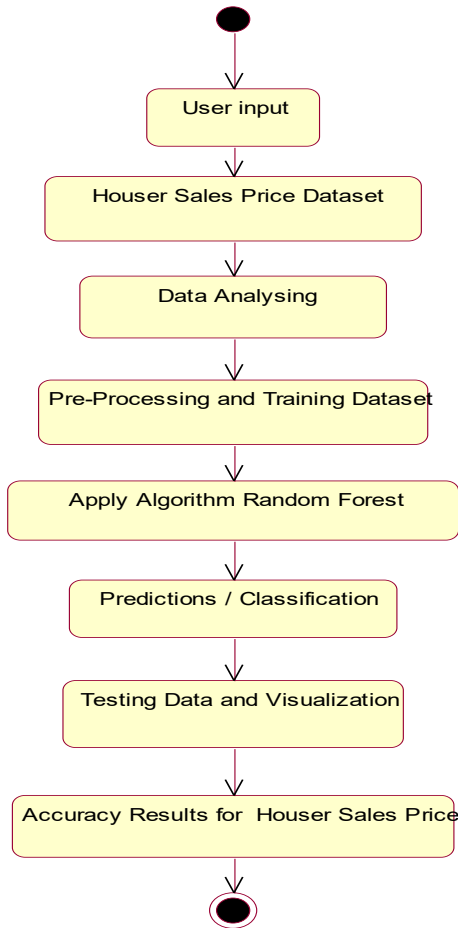
16.4 COLLABORATION DIAGRAM

A collaboration diagram, also called a communication diagram or interaction diagram, is an illustration of the relationships and interactions among software objects in the Unified Modelling Language (UML). The concept is more than a decade old although it has been refined as modelling paradigms have evolved.



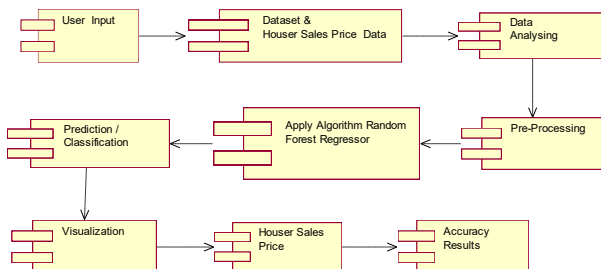
16.5 STATE DIAGRAM

State diagram are a loosely defined diagram to show workflows of stepwise activities and actions, with support for choice, iteration and concurrency. State diagrams require that the system described is composed of a finite number of states; sometimes, this is indeed the case, while at other times this is a reasonable abstraction. Many forms of state diagrams exist, which differ slightly and have different semantics.



16.6 COMPONENT DIAGRAM

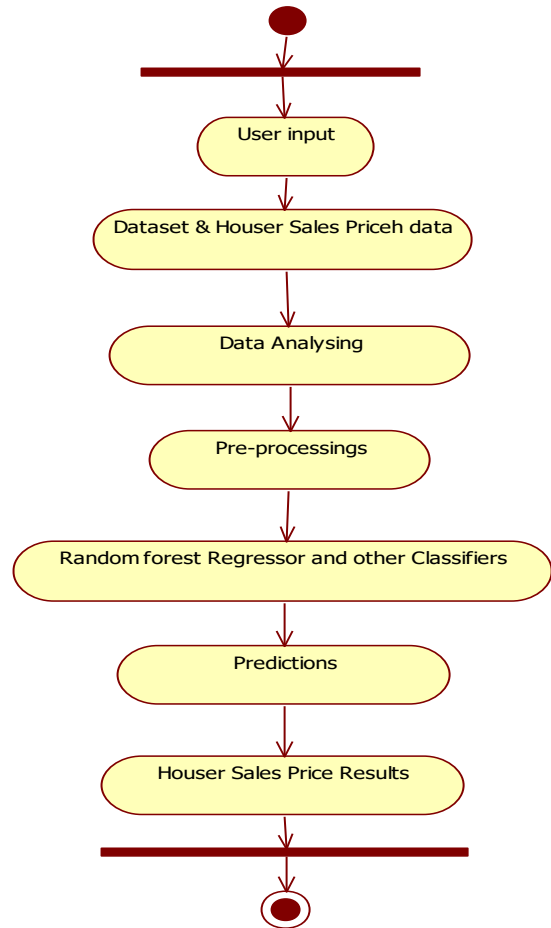
In the Unified Modelling Language, a component diagram depicts how components are wired together to form larger components and or software systems. They are used to illustrate the structure of arbitrarily complex systems. User gives main query and it converted into sub queries and sends through data dissemination to data aggregators. Results are to be showed to user by data aggregators. All boxes are components and arrow indicates dependencies.



16.7 ACTIVITY DIAGRAM

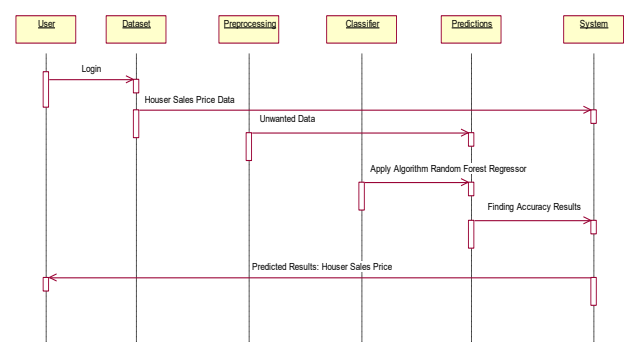
Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modelling Language, activity diagrams can be

used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.



16.8 SEQUENCE DIAGRAM

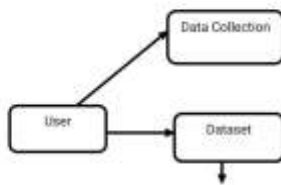
.A sequence diagram in Unified Modelling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario.



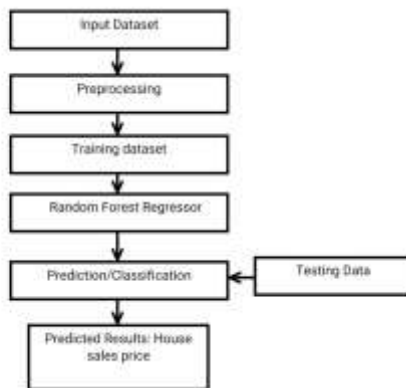
16.9 DATA FLOW DIAGRAM

A data flow diagram (DFD) is a graphical representation of the "flow" of data through an information system, modelling its process aspects. Often they are a preliminary step used to create an overview of the system which can later be elaborated. DFDs can also be used for the visualization of data processing (structured design). A DFD shows what kinds of data will be input to and output from the system, where the data will come from and go to, and where the data will be stored. It does not show information about the timing of processes, or information about whether processes will operate in sequence or in parallel.

Level 0 :

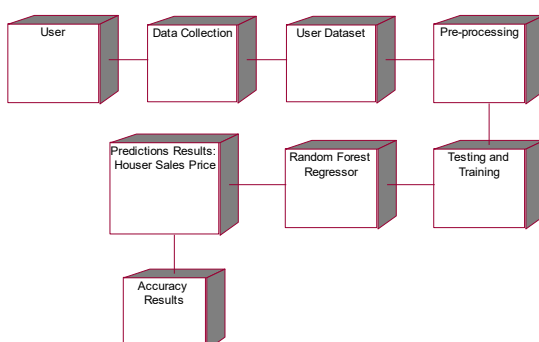


Level 1:



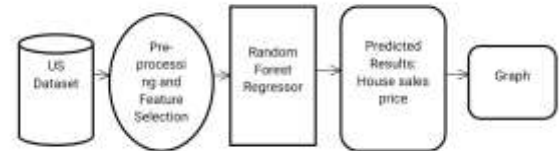
16.10 DEPLOYMENT DIAGRAM

Deployment Diagram is a type of diagram that specifies the physical hardware on which the software system will execute. It also determines how the software is deployed on the underlying hardware. It maps software pieces of a system to the device that are going to execute it.



16.11 SYSTEM ARCHITECTURE

The diagram represents the workflow of a House Price Prediction system using Machine Learning. Initially, the US housing dataset is collected and processed through preprocessing and feature selection techniques to improve data quality. Then, the Random Forest Regressor predicts house sale prices, and the predicted results are displayed graphically for better analysis and visualization.



17.SOFTWARE TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

17.1 DEVELOPING METHODOLOGIES

The test process is initiated by developing a comprehensive plan to test the general functionality and special features on a variety of platform combinations. Strict quality control procedures are used. The process verifies that the application meets the requirements specified in the system requirements document and is bug free. The following are the considerations used to develop the framework from developing the testing methodologies.

17.2 TYPES OF TESTS

17.2.1 UNIT TESTING

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program input produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application. It is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component

level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

17.2.2 FUNCTIONAL TESTING

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application
outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked.

17.2.3 SYSTEM TESTING

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

17.2.4 PERFORMANCE TESTING

The Performance test ensures that the output be produced within the time limits ,and the time taken by the system for compiling, giving response to the users and request being send to the system for to retrieve the results.

17.2.5 INTEGRATION TESTING

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software

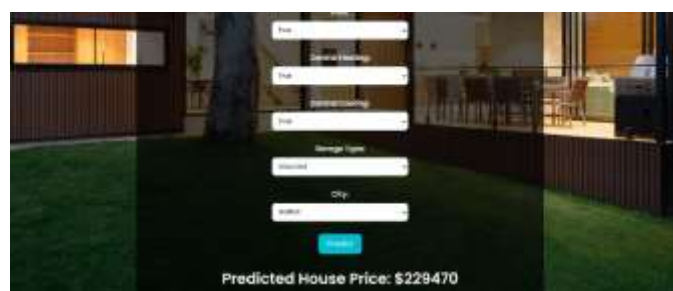
applications at the company level – interact without error.

18. RESULTS

The experimental results indicate that the Random Forest Regressor performed efficiently for house price prediction. The model achieved a prediction accuracy of 97.6%. The Mean Absolute Error on the training dataset was 1.4606, while the testing dataset produced a Mean Absolute Error of 3.74. The model successfully identified important features such as livable square footage, number of bedrooms, and city location as key contributors to house price estimation. Compared with traditional regression approaches, the proposed system demonstrated better generalization capability and robustness against noisy data.



id	year_built	bedrooms	bathrooms	full_bathrooms	half_bathrooms	total_sqft	sqft_garage	type_garage	sqft_carport	sqft_porch	pool	pool2	pool3
1	1976	3	4	3	1	1892	1892	attached	560	0	True	False	True
2	1958	3	3	3	0	1684	2061	attached	451	0	True	False	True
3	1992	3	3	2	0	1681	1978	none	0	625	False	False	True
4	1994	3	4	2	0	1825	1277	attached	479	0	True	False	True
5	1995	3	4	2	0	1680	1149	attached	430	0	True	False	True
6	1995	3	3	2	0	1621	1672	attached	420	0	True	False	True
7	1979	3	3	2	0	2280	2380	detached	523	0	True	False	True
8	1993	3	3	2	0	1748	1748	none	0	0	False	False	False



19.FUTURE ENHANCEMENTS

In the future, comparisons can be made with different classification algorithms and can measure the performance of the system using various machine learning algorithms.

20. DISCUSSIONS

The research demonstrates that machine learning algorithms can significantly improve the accuracy of house price prediction systems. Random Forest Regressor effectively handled large-scale datasets and nonlinear relationships among variables. The ensemble nature of the algorithm minimized overfitting and increased prediction reliability. The system can be extended further by integrating real-time market trends, geographic information systems, and deep learning models. Future improvements may also include deployment as a web application for real estate agencies and customers.

21. CONCLUSIONS

This paper presented a machine learning-based house price prediction system using Random Forest Regressor. The proposed model achieved high accuracy and reliable predictions using a large housing dataset. The results show that the system can assist buyers, sellers, and real estate professionals in making data-driven decisions. The research confirms that Random Forest Regressor is an effective algorithm for regression-based prediction tasks in the real estate domain.

TABLE I. PERFORMANCE METRICS

Metric	Value
Training MAE	1.4606
Testing MAE	3.74
Accuracy	97.6%
Dataset Size	42,703 records

Table -1: Performance Metrics

REFERENCES

[1] Priya Gour, Sudhanshu Vashistha and Pradeep Jha, "Twitter Sentiment Analysis Using Naive Bayes based Machine learning Technique", 2nd International

Conference on Sentiment Analysis and Deep Learning (ICSADL 2022), Springer - Advances in Intelligent Systems and Computing Series 1432, 2023.

[2] H. Arora, G. K. Soni, R. K. Kushwaha and P. Prasoon, "Digital Image Security Based on the Hybrid Model of Image Hiding and Encryption", 2021 IEEE 6th International Conference on Communication and Electronics Systems (ICCES), pp. 1153-1157, 2021.

[3] M. Jain, H. Rajput , N. Garg and P. Chawla, "Prediction of House Pricing using Machine Learning with Python", 2020 IEEE International Conference on Electronics and Sustainable Communication Systems (ICESC), pp. 570-574, 2020

[4] Himanshu Arora, Shilpi Mishra and Manish Dubey, "Development of the Framework for the Solution of the Security Problems in Data Transmission Involving Advanced Asymmetric Algorithm", International Journal of Emerging Technology and Advanced Engineering, vol. 8, no. 4, pp. 18-20, April 2018.

[5] Tan F, Cheng C, Wei Z., "Time-Aware Latent Hierarchical Model for Predicting House Prices", In2017 IEEE International Conference on Data Mining (ICDM), pp. 1111- 1116, 2017.

BIOGRAPHIES



Pathipati Chandrasekhar
,Assistant Professor ,Computer
Science and Engineering , Guru
Nanak Institute of Technology



S.Vishwasree ,Student ,
Computer Science and
Engineering , Guru Nanak
Institute of Technology



S.Bhavitha,Student,Computer
Science and Engineering , Guru
Nanak Institute of Technology



T.Mounika,Computer Science
and Engineering , Guru Nanak
Institute of Technology