HOME INSIGHTS: PRICE PREDICTION AND RECOMMENDATIONS

PHASE I REPORT

Submitted by

MADHUMITHA K 210701141
MOHANA PRASATH G 210701163

in partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE ENGINEERING

RAJALAKSHMI ENGINEERING COLLEGE

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

ANNA UNIVERSITY, CHENNAI

NOV 2024

ANNA UNIVERSITY, CHENNAI

BONAFIDE CERTIFICATE

Certified that this Report titled **"Home Insights: Price Prediction & Recommendations"** is the bonafide work of **MADHUMITHA K (210701141), MOHANA PRASATH G (210701163)** who carried out the work under my supervision. Certified further that to the best of my knowledge the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

Dr. Kumar P, Ph.D.
Professor and Head
Department of Computer Science Engineering
Rajalakshmi Engineering College
Chennai – 602 105

Dr. Senthil Pandi S, M.E.,Ph.D.
Associate Professor
Department of Computer Science Engineering
Rajalakshmi Engineering College
Chennai – 602 105

Submitted to Project-I Viva-Voce Examination held on _____

Internal Examiner **External Examiner**

ABSTRACT

The real estate market demands precise, data-driven solutions to assess property values and provide personalized recommendations for buyers and investors. This project addresses these needs in two phases, leveraging advanced machine learning (ML) models to predict property prices based on parameters like location (latitude and longitude), property size (square footage), and configuration (BHK). Enriched with factors such as market trends, local amenities, and historical pricing patterns, the system offers reliable insights into property valuation, enabling stakeholders to navigate market complexities with confidence. The Hyperparameter Tuning Module optimizes model performance using techniques like grid search and Bayesian optimization, ensuring enhanced accuracy and efficiency. Meanwhile, the Recommendation Module enhances user experience with personalized property suggestions based on predicted prices, user preferences, and contextual data. Together, these modules create a robust platform that delivers precise predictions and actionable recommendations, empowering users to make well-informed decisions in the dynamic and ever-evolving real estate market.

ACKNOWLEDGEMENT

Initially we thank the Almighty for being with us through every walk of our life and showering his blessings through the endeavor to put forth this report. Our sincere thanks to our Chairman **Mr. S.MEGANATHAN, B.E, F.I.E.**, our Vice Chairman **Mr. ABHAY SHANKAR MEGANATHAN, B.E., M.S.,** and our respected Chairperson **Dr. (Mrs.) THANGAM MEGANATHAN**, **Ph.D.,** for providing us with the requisite infrastructure and sincere endeavoring in educating us in their premier institution.

Our sincere thanks to **Dr. S.N. MURUGESAN, M.E., Ph.D.,** our beloved Principal for his kind support and facilities provided to complete our work in time. We express our sincere thanks to **Dr.P.KUMAR, Ph.D.**, Professor and Head of the Department of Computer Science and Engineering for his guidance and encouragement throughout the project work. We convey our sincere and deepest gratitude to our internal guide, **Dr. Senthil Pandi S, M.E., Ph.D.** Department of Computer Science and Engineering. Rajalakshmi Engineering College for her valuable guidance throughout the course of the project. We are very glad to thank our Project Coordinator, **Dr.Kumaragurubaran T , Ph.D** Department of Computer Science and Engineering for his useful tips during our review to build our project.

Madhumitha K : 210701141

Mohana Prasath G : 210701163

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

SNO	ABBREVIATION	EXPANSION
1	ML	Machine Learning
2	AI	Artificial Intelligence
3	R^2	Coefficient of Determination (Model Accuracy Metric)
4	GB	Gradient Boosting
5	XGBoost	Extreme Gradient Boosting
6	LightGBM	Light Gradient Boosting Machine
7	HistGB	Histogram-based Gradient Boosting
8	CatBoost	Categorical Boosting
9	RF	Random Forest
10	MAE	Mean Absolute Error (Evaluation Metric)
11	MSE	Mean Squared Error (Evaluation Metric)
12	RMSE	Root Mean Squared Error (Evaluation Metric)

CHAPTER I

INTRODUCTION

House price prediction is a crucial aspect of the real estate market, influencing decisions made by homebuyers, investors, developers, and policymakers. Accurate prediction models can provide valuable insights into market trends, helping stakeholders make informed decisions. Given the complexity of real estate data—characterized by a multitude of factors such as location, economic conditions, and property features—developing reliable predictive models is challenging yet essential.

This project focuses on leveraging advanced machine learning techniques to predict house prices with greater accuracy. By putting forward novel ideas that increase prediction accuracy and provide useful applications for the real estate sector, we hope to support the ongoing efforts in the field.

Accurately estimating property values in the quickly changing real estate market continues to be a major obstacle for potential homeowners and investors. Conventional property valuation techniques frequently lack accuracy and don't take into consideration the intricate interactions between many elements that affect property values. Developing a predictive model that successfully combines various data sources and cutting-edge machine learning algorithms to produce accurate and customized property price forecasts is the main difficulty. This initiative is driven by the growing need in a turbulent real estate market for more precise and customized real estate information. This project intends to develop an efficient ML-based platform by utilizing a large dataset that contains real estate listings, geographic data, and additional external aspects like market trends and local amenities.

The objective of this project is to develop an advanced machine learning (ML)-based website designed to accurately forecast property values by leveraging cutting-edge algorithms and comprehensive datasets. The project's goal is to develop a prediction model by combining information from many sources, such as real estate listings, geographic information (latitude, longitude), property features (BHK - Bedrooms, Hall, Kitchen), and additional external factors such as market trends and local amenities. The website will have an easy-to-use design that lets visitors explore property alternatives, enter search parameters, and get forecasts that are specific to them. The ultimate goal is to deliver a valuable resource for homeowners, investors, and real estate professionals, enabling more

precise property assessments and informed decision-making in the dynamic real estate market.

In the second phase of this project, the focus shifts towards enhancing the performance and usability of the prediction model through hyperparameter tuning and the integration of a recommendation system. Hyperparameter tuning ensures the optimization of machine learning models by fine-tuning parameters to achieve the best possible accuracy and efficiency. Simultaneously, the recommendation module provides users with personalized property suggestions based on their preferences, predicted prices, and contextual factors like market trends and location features. This dual-phase approach not only improves the predictive accuracy but also elevates user experience, making the platform a comprehensive solution for stakeholders in the real estate sector. By combining robust prediction capabilities with intelligent recommendations, this project aims to redefine property valuation and decision-making processes in the ever-evolving real estate landscape.

CHAPTER II

LITERATURE REVIEW

Anders Hjort et al.[1] proposed a tree boosting model, Locally Interpretable Tree Boosting (LitBoost) that combines the strengths of Generalized Additive models(GAM) and Gradient Boosted Trees(GBT). It improves predictive performance by avoiding overfitting and also enhances predictability. It was observed that GAM displayed poor performance when the observations were small. The proposed method LitBoost outperforms GAM when the number of observations per group is small and performs better than GBT when specific interactions are included in the data1. Abigail Bola Adtunji et al[2]. studied the performance of Random Forest for house price prediction and have shown that the model is able to achieve a minor difference between the predicted value and the actual price2. Lu Wang et al[3]. proposed a spatiotemporal model(FSTM) to predict the prices of houses using the spatiotemporal characteristics of small cities in China. The model was designed to handle the relationship between space and time reflecting the variation in house prices across different locations and time periods. A flexible structure that combined the long term trends with spatially correlated random factors enabled the model to predict accurately.

Anders Hjort et al[4]. introduced an improvement in XGBoost by introducing a loss function Squared percentage error (SPE). The introduction of the SPE loss function improves model performance from 89.4% to 90.0% under the 20% measure. XGBoost-SPE reduces incorrect predictions by 4.9% compared to XGBoost-SE, particularly improving in lower price segments, though it performs slightly worse in higher price segments. Combining both models into a hybrid yields even better results, achieving 90.4% accuracy and further reducing errors by 9.3%, demonstrating the effectiveness of this combined approach.

Jun Liu[5] proposed a hybrid method by combining advanced neural network models with spatial analysis techniques. that includes spatial factors in house price predictions. This integration not only improves predictive accuracy but also provides a new perspective on incorporating geographical factors into real estate forecasting. To forecast the cost of used homes in Sanghai, Zhongyun Jiang et al[6]. suggested a multilayer feedforward neural network trained using the error inverse propagation approach. SCR is a hybrid approach

that was suggested by Sureyya Ozogur Akyur et al[7]. that combines support vector regression, closest neighbor classification, clustering analysis, and linear regression. One method's output is used as another's input. The research will classify the houses for which the cluster is unknown, generate distinct housing clusters using the data at hand, and anticipate prices by developing distinct prediction models for each class7 in order to provide a hybrid method. Salim Lahmiri et al[8]. used Gaussian process regression, support vector regression, and ensemble regression trees to forecast housing prices in Taiwan. The experimental results showed that boosting ensemble regression trees performed better than Gaussian process regression and support vector regression. All three prediction algorithms fared better than artificial neural networks.

In 2024, Yaping Zhao[9], Jichang Zhao[9], and Edmund Y. Lam[9] explored the application of multiple machine learning models, including Multilayer Perceptron (MLP), Support Vector Machine (SVM), Linear Regression, XGBoost, and Random Forest. Their focus was on evaluating the models' accuracy in predictive tasks, highlighting the effectiveness of combining various algorithms for improved performance. In 2019, Feng Wang[10], Yang Zou[10], Haoyu Zhang[10], and Haodong Shi[10] investigated the use of Support Vector Machine (SVM) and Autoregressive Integrated Moving Average (ARIMA) models for predictive analysis. They evaluated the performance of these models using metrics such as Root Mean Squared Error (RMSE) and accuracy, aiming to enhance the reliability of their predictions in various applications.

In 2023, B. Usha Sri[11], Chaganti Santhosh Kumar Reddy[11], Chiluveri Rithish Kumar[11], Akula Vyshnavi[11], Buttagalla Vinod[11], and Boojala Kiran Reddy[11] conducted a comprehensive study utilizing various machine learning models, including Random Forest Algorithm, Lasso Regression, Linear Regression, and Decision Tree. They assessed the performance of these models through multiple evaluation metrics, such as Mean Squared Error (MSE), Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and R-squared Error, to determine their effectiveness in predictive analytics.

In 2022, Shruti Goswami[12], Vijendra Singh Bramhe[12], and Shaveta Khepra[12] explored advanced neural network architectures, including Long Short-Term Memory (LSTM), Convolutional Neural Network (CNN), and Recurrent Neural Network (RNN), to enhance predictive accuracy in their study. They evaluated the performance of these models using metrics such as Mean Absolute Percentage Error (MAPE) and Training Loss,

providing valuable insights into their effectiveness and reliability in handling complex data patterns.

In 2021, Yihao Chen[13], Runtian Xue[13], and Yu Zhang[13] investigated the efficacy of various predictive modeling techniques, including Linear Regression, Support Vector Machine (SVM), and Deep Neural Network (DNN). They assessed the performance of these models using metrics such as Root Mean Square Error (RMSE) and Mean Square Error (MSE), offering insights into their accuracy and suitability for different predictive tasks. In 2020, Winky K.O. Ho[14], Bo-Sin Tang[14], and Siu Wai Wong[14] conducted a study that evaluated the performance of several machine learning algorithms, specifically Support Vector Machine (SVM), Random Forest, and Gradient Boosting Machine. They utilized performance metrics such as Mean Squared Error (MSE), Root Mean Square Error (RMSE), and Mean Absolute Percentage Error (MAPE) to analyze the accuracy and effectiveness of these models in predictive tasks.

CHAPTER III

SYSTEM DESIGN

3.1 SYSTEM FLOW DIAGRAM

The proposed model follows a structured workflow, beginning with Data Collection, where relevant data is gathered from various sources. This is followed by Exploratory Data Analysis (EDA) to uncover key patterns, relationships, and insights within the data. Next, the process involves Data Preprocessing, where the data is cleaned, transformed, and made suitable for modeling. In the Model Selection phase, different machine learning models are evaluated to determine the best fit for the problem. Once the model is chosen, Model Training takes place, allowing the model to learn from the data. This is followed by Validation and Testing to assess the model's performance and fine-tune it for optimal results. Finally, the trained model is used to Predict the Output, providing accurate forecasts for house prices based on user input parameters. Fig.1 depicts the workflow of the model.

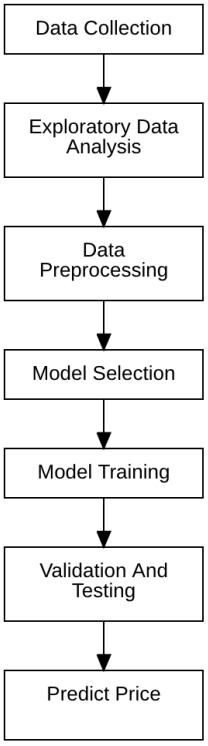

Fig. 1. Workflow model

3.2 SYSTEM ARCHITECTURE DIAGRAM

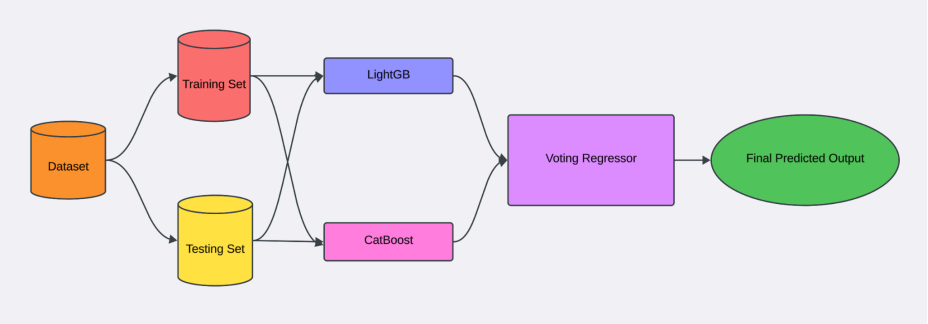

Fig. 2. Architecture Diagram

Fig 2 presents a comprehensive workflow for predicting house prices using an ensemble of two powerful machine learning models - LightGBM and CatBoost - combined through a Voting Regressor. The system is structured to maximize predictive accuracy by integrating the strengths of both models within a carefully organized pipeline, which begins with data collection and proceeds through multiple stages of data preparation, model training, and prediction generation.

The first stage of the system is Data Collection, where a dataset is compiled with key property attributes, such as location (latitude and longitude), property size (in square feet or square meters), and configuration (number of bedrooms, hall, and kitchen). These features are essential for accurate house price prediction, as they capture both the physical characteristics of the property and its geographic location. After data collection, the dataset is split into two parts: a Training Set and a Testing Set. The Training Set is used to help the model learn from historical patterns, while the Testing Set is reserved for evaluating the accuracy and robustness of the final model.

Once the data is prepared, the Model Training phase begins. Here, two machine learning models - LightGBM and CatBoost - are trained on the Training Set. LightGBM, or Light Gradient Boosting Machine, is a high-performance gradient boosting algorithm that is optimized for speed and efficiency. It handles large datasets well and uses a leaf-wise growth strategy that can reduce computational costs, making it ideal for this application. CatBoost, short for Categorical Boosting, is another gradient boosting algorithm known for its ability to work effectively with categorical features and handle complex relationships within the data with minimal preprocessing. Both models independently learn patterns in the data to predict house prices.

After individual training, the outputs of LightGBM and CatBoost are combined using a Voting Regressor, an ensemble technique designed to improve predictive accuracy. The Voting Regressor consolidates predictions from both models, either by averaging them or assigning weights to each model's output based on their individual performance. This approach capitalizes on the complementary strengths of LightGBM and CatBoost: while LightGBM's efficiency makes it strong in handling high-dimensional data, CatBoost excels in managing categorical variables and complex interactions. By integrating both models, the Voting Regressor reduces errors associated with single-model predictions and enhances the system's robustness.

In the final stage, the Voting Regressor outputs a Final Predicted Output—the house price estimate. This output represents the combined predictive capability of both LightGBM and CatBoost, offering an accurate and reliable estimate. Such a prediction is invaluable to users like prospective homeowners and real estate investors, who can use this information to make informed decisions based on current market trends and specific property characteristics. Additionally, the Voting Regressor's ensemble approach ensures that the final prediction is not only accurate but also generalizes well to new data, reducing the risk of overfitting.

In summary, this architecture leverages the strengths of LightGBM and CatBoost to provide a robust and accurate system for house price prediction. By combining these models in a Voting Regressor ensemble, the system benefits from a balanced approach that draws on the best qualities of both algorithms. This comprehensive and systematic design ensures the system's reliability, making it a powerful tool for real-world applications in real estate analytics. The architecture's structured workflow - encompassing data collection, model training, ensemble prediction, and output generation - demonstrates a thoughtful approach to solving complex regression problems in the domain of property price prediction.

3.3 ACTIVITY DIAGRAM

The activity diagram represents the workflow of the system, outlining the processes involved. The Fig.3 represents the activity diagram which outlines the process involved in property price prediction and recommendation generation. It starts with the user entering property details and preferences. The input is validated, and if found valid, the system

performs predictions using a trained machine learning model. Subsequently, hyperparameters are optimized to enhance model performance. Recommendations are generated based on predictions and user preferences. The workflow ends with the system displaying the predictions and recommendations to the user. In case of invalid input, an error message is displayed, and the process loops back to the start. This diagram clearly shows the step-by-step functionality and decision points of the system.

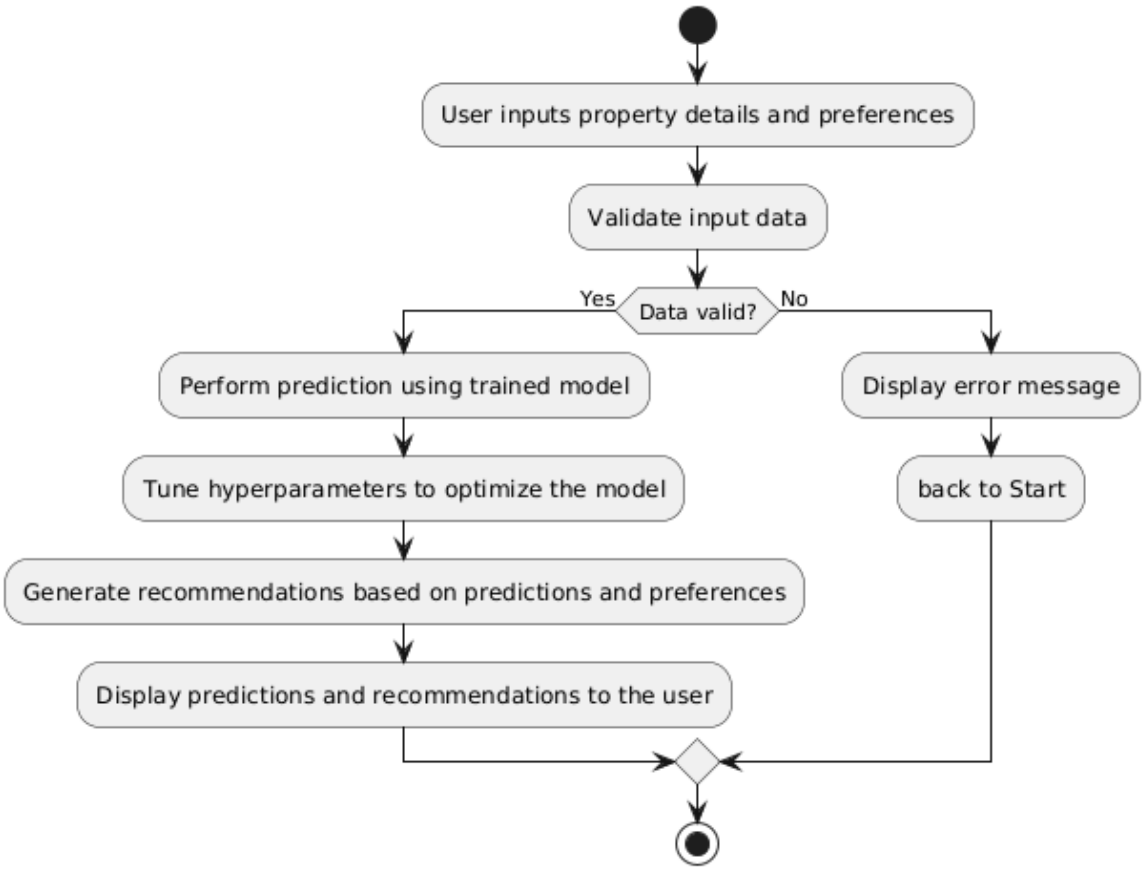

Fig. 3. Activity diagram

3.4 CLASS DIAGRAM

The class diagram shows the static structure of the system, highlighting the key components and their relationships. Fig. 4 represents the class diagram of the model with elements DataProcessor, PredictionModel, HyperparamterTuner, RecommendationEngine and userInterface. DataProcessor: Responsible for cleaning the raw data and performing feature engineering to prepare it for prediction. PredictionModel: Handles training and prediction of property prices. HyperparameterTuner: Optimizes the model's parameters to

improve accuracy. RecommendationEngine: Generates personalized property recommendations for the user. UserInterface: Manages user input and output interactions. The relationships illustrate how each class interacts: the PredictionModel relies on the DataProcessor for clean data, while the HyperparameterTuner assists in optimizing the model. The UserInterface interacts with both the PredictionModel and RecommendationEngine, enabling a seamless user experience.

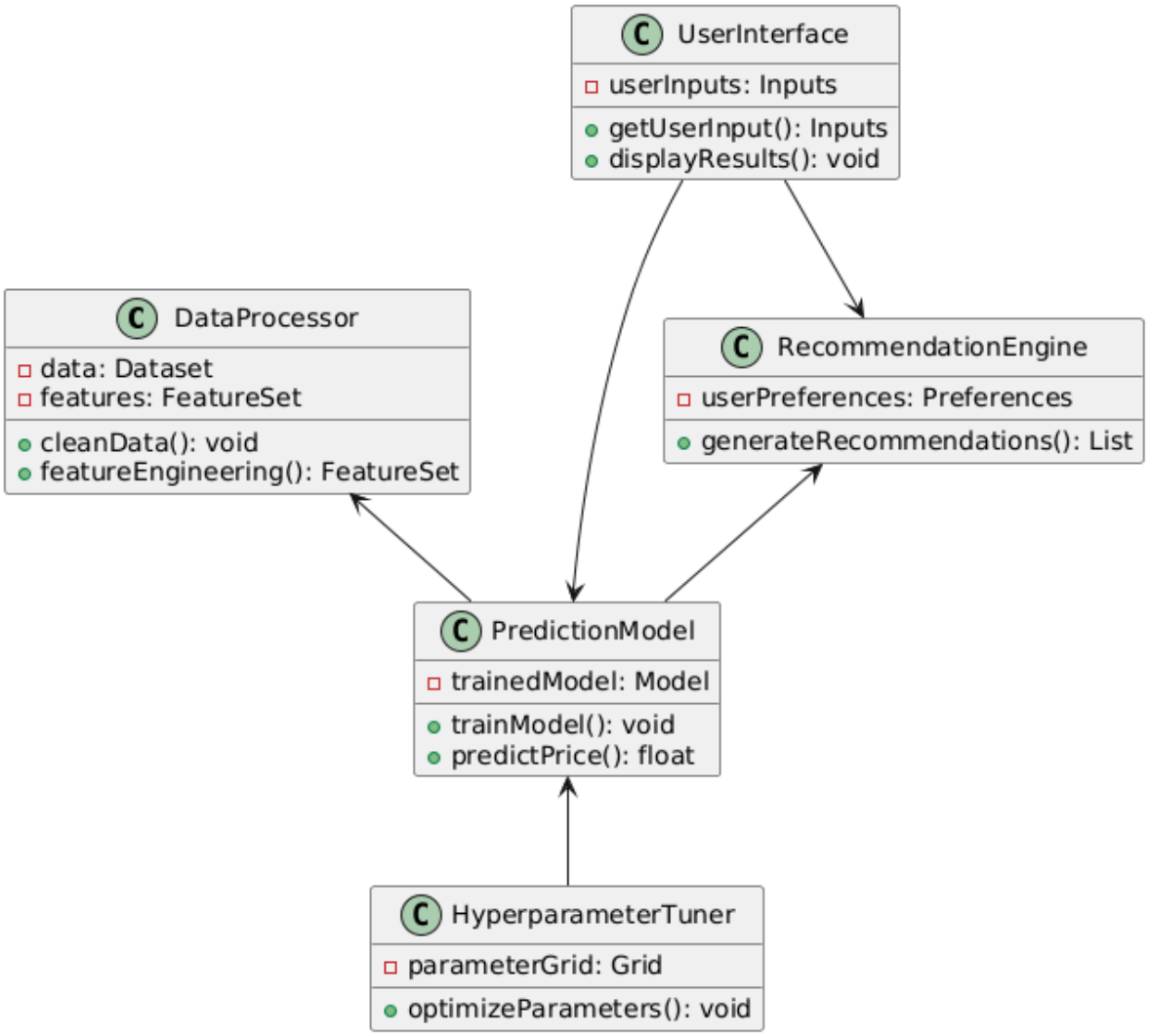

Fig. 4. Class diagram

3.5 COMPONENT DIAGRAM

The component diagram highlights the physical structure of the system, showing how the main components interact and depend on each other. Fig. 5 represents the component diagram which is divided into four key modules. User Interface: The entry point for user

inputs and the module responsible for displaying results. Prediction Module: Contains the trained ML models used for property price predictions. Hyperparameter Tuning Module: Optimizes model parameters for enhanced performance. Recommendation Module: Generates property recommendations based on predictions and user preferences. All modules are connected to a central Database, which stores property data, user preferences, and model parameters. The diagram illustrates the high-level architecture of the system, ensuring modularity and ease of scalability.

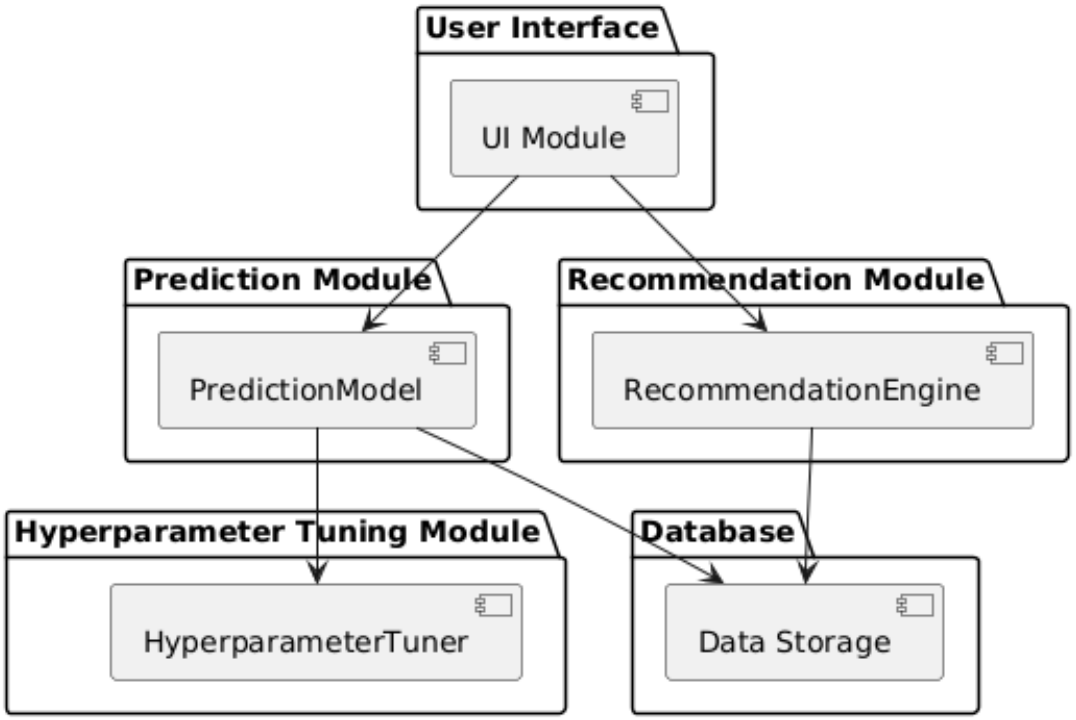

Fig. 5. Component diagram

3.6 COLLABORATION DIAGRAM

The collaboration diagram describes how objects in the system interact to achieve the overall functionality. Fig. 6 represents the collaboration diagram with the elements UserInterface, RecommendationEngine, PredictionModel and HyperparamterTuning. The UserInterface initiates the process by sending user inputs to the PredictionModel. The PredictionModel interacts with the HyperparameterTuner to optimize parameters, and the tuned results are used to make predictions. Concurrently, the UserInterface requests recommendations from the RecommendationEngine, which processes the inputs and

returns personalized property suggestions. The UserInterface aggregates the results and displays them to the user. This diagram emphasizes the interactions between objects and their roles in the system.

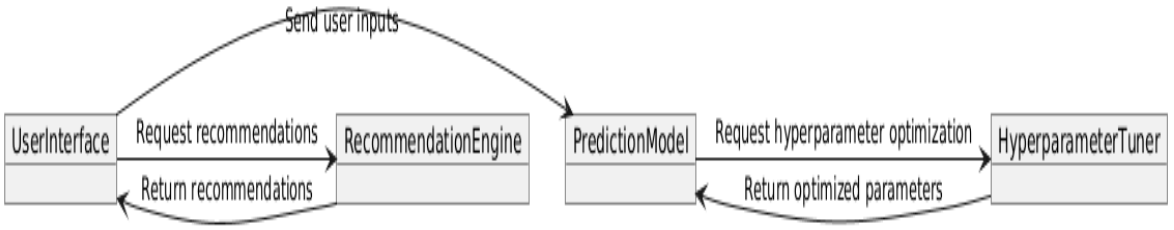

Fig. 6. Collaboration diagram

CHAPTER IV

METHODOLOGY

4.1 DESCRIPTION OF THE DATASET

The dataset was gathered from Kaggle. The dataset used in this study came from Kaggle, a website that offers a wide variety of datasets for analysis and research.Fig. 7 represents the description of the dataset. ID, date, number of bedrooms, number of bathrooms, living area, lot area, number of floors, presence of the waterfront, number of views, house condition, house grade, area of the house (excluding basement), area of the basement, built area, year of renovation, postal code, latitude, longitude, living_area_enov, lot_area_renov, number of schools nearby, and distance from the airport are some of the input features that make up the dataset. "Price" is the label assigned to the target variable, or output feature, which represents the house's price. The dataset consists of 14620 rows and 23 columns, including the output feature.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 14620 entries, 0 to 14619
Data columns (total 23 columns):
 #   Column                                 Non-Null Count  Dtype
---  ------                                 --------------  -----
 0   id                                     14620 non-null  int64
 1   Date                                   14620 non-null  int64
 2   number of bedrooms                     14620 non-null  int64
 3   number of bathrooms                    14620 non-null  float64
 4   living area                            14620 non-null  int64
 5   lot area                               14620 non-null  int64
 6   number of floors                       14620 non-null  float64
 7   waterfront present                     14620 non-null  int64
 8   number of views                        14620 non-null  int64
 9   condition of the house                 14620 non-null  int64
 10  grade of the house                     14620 non-null  int64
 11  Area of the house(excluding basement)  14620 non-null  int64
 12  Area of the basement                   14620 non-null  int64
 13  Built Year                             14620 non-null  int64
 14  Renovation Year                        14620 non-null  int64
 15  Postal Code                            14620 non-null  int64
 16  Lattitude                              14620 non-null  float64
 17  Longitude                              14620 non-null  float64
 18  living_area_renov                      14620 non-null  int64
 19  lot_area_renov                         14620 non-null  int64
 20  Number of schools nearby               14620 non-null  int64
 21  Distance from the airport              14620 non-null  int64
 22  Price                                  14620 non-null  int64
dtypes: float64(4), int64(19)
memory usage: 2.6 MB
```

Fig.7. Dataset Description

4.2 EXPLORATORY DATA ANALYSIS

Exploratory Data Analysis- Initially, the dataset should be analyzed to extract meaningful insights. The number of attributes, relationships between features, and the relationship between each input attribute and the output feature are evaluated to better understand the data. This step provides a foundation for model training. Here, techniques like the

Confusion matrix, box plot and mutual information for regression are used to analyze the data. Confusion matrices show the relationship of each attribute in the dataset with every other attribute in the dataset.

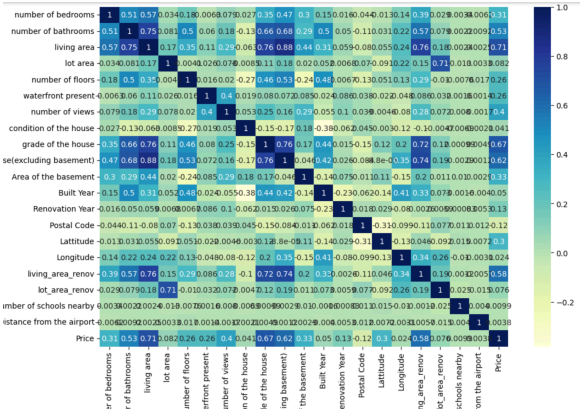

Fig. 8. Confusion Matrix

In the above confusion matrix in Fig 8, the output feature Price has a strong positive correlation with the living area.The price also has a strong positive correlation with grade of the house and area of the house (excluding basement). The Price has a weak negative correlation with Postal code.

A box plot helps to visualize the distribution of numerical data, highlighting any outliers, and can be used with both numerical and categorical features. It provides insights into data symmetry, spread, and skewness.

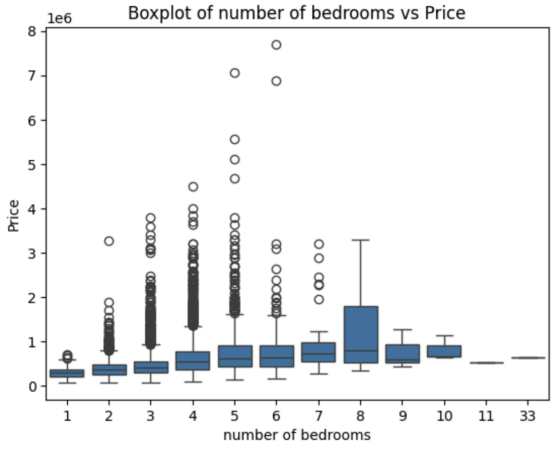

Fig. 9. Box Plot of Number of bedrooms Vs Price

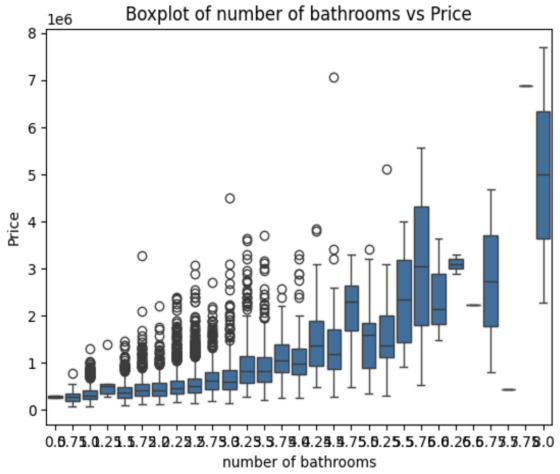

Fig. 10. Box Plot of Number of bathrooms Vs Price

Fig 9 represents the box plot of Price versus number of bedrooms. The price of the home somewhat rises as the number of bedrooms increases. Fig 10 represents the box plot of price versus number of bathrooms. The number of bathrooms and the price have a linear connection. The cost likewise rises linearly with the number of bathrooms.

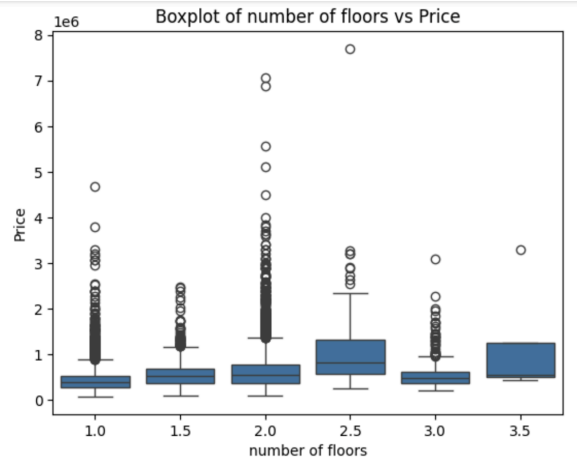

Fig. 11. Box Plot of Number of floors Vs Price

Fig 11 represents the box plot of Price versus number of floors. There is no relation between the number of floors and the price of the price.Fig 12 represents the box plot of condition of the house versus the price. House condition doesn't seem to have a strong linear relationship with the median price. Although better conditions (4 and 5) have more expensive houses, the price distribution is still concentrated towards lower values with a few outliers being much higher.

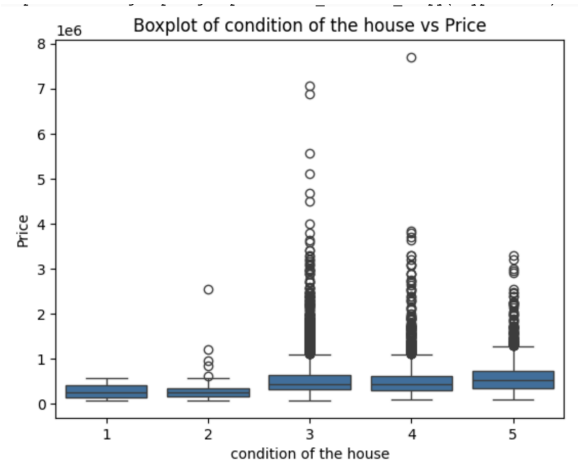

Fig. 12. Box Plot of Condition of the house Vs Price

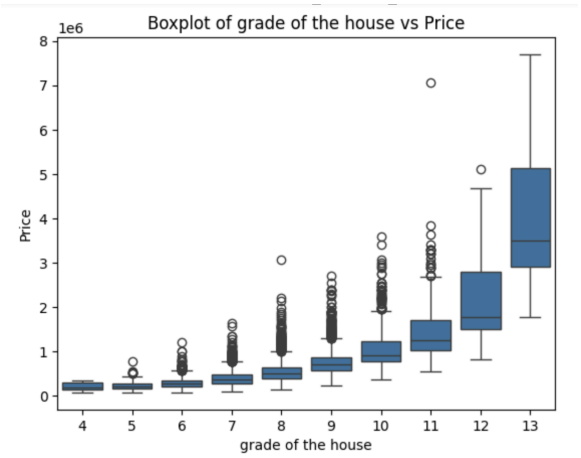

Fig. 13. Box Plot of Grade of the house Vs Price

The box plot showing the house's grade against price is shown in Fig. 13. The grade of the house and the property's price have a strong linear relationship. Mutual information measures the dependency between variables, giving an idea of how much information about the target is provided by each feature. Fig. 14 depicts the mutual information graph. Here the postal code gives high mutual information. Other strong mutual information are given by the features - Living area, Grade of the house, Latitude and living_area_renov.

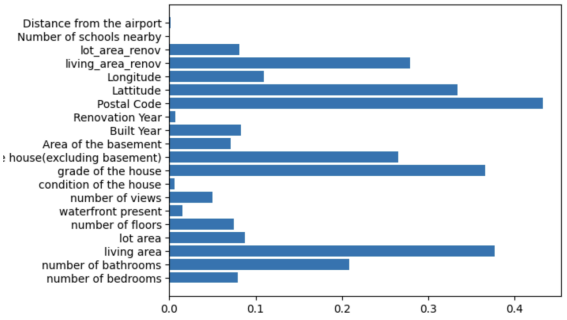

Fig. 14. Mutual Information

4.3 DATASET PREPROCESSING

Dataset Preprocessing- Before proceeding with model training, it's essential to check for null values and categorical variables in the dataset. Null values can either be removed or filled with appropriate default values (e.g., mean, median, or mode). It is necessary to translate categorical variables into numerical values, frequently with the aid of methods like Label Encoding or One-Hot Encoding. One-Hot Encoding generates binary columns for every category, whereas Label Encoding gives each category a distinct integer. This ensures that the data is clean and ready for the machine learning algorithms, which typically require numerical inputs.

4.4 MODEL SELECTION AND MODEL TRAINING

Model Selection and Model Training- In the model selection phase, several machine learning algorithms were evaluated to determine the best approach for predicting house prices. The models considered included Random Forest, XGBoost, Gradient Boosting, Extra Trees, LightGBM, Hist Gradient Boosting, and CatBoost. These models were chosen due to their strength in handling complex, non-linear relationships, making them well-suited for a regression problem like house price prediction.

Each model was trained on the dataset after performing feature selection and preprocessing steps such as handling missing values and encoding categorical variables. The training process involved feeding the models with the input features and their corresponding house prices (target variable). Hyperparameter tuning was also conducted to optimize the performance of each model, ensuring that they could learn patterns in the data effectively.

After training, the performance of each model was evaluated based on the R-squared score, which measures how well the model predicts the variance in house prices. The results indicated strong performance across the board, with CatBoostand LightGBM standing out with the highest R-squared scores.

In the next step, an ensemble method using a Voting Regressor was employed to combine the predictions of the top-performing models—CatBoost and LightGBM. This technique allowed the model to take advantage of the strengths of both algorithms, resulting in improved predictive accuracy and robustness.

The final trained model was then validated using cross-validation, confirming that it generalized well to unseen data, making it suitable for real-world house price prediction tasks.

In the testing phase, several machine learning models were trained and evaluated based on their R-squared scores, which measure how well each model explains the variance in the house price data. The individual R-squared scores were as follows: 88% for Random Forest, 87% for XGBoost, 86% for Gradient Boosting, 88% for Extra Trees, 89% for LightGBM, 87% for Hist Gradient Boosting, and 90% for CatBoost.

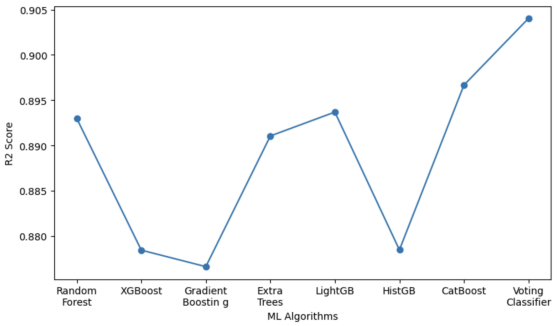

Fig. 15. R2 score of all models

These results show that while each model performed well in predicting house prices, CatBoost achieved the highest R-squared score of 90%, followed closely by LightGBM at 89%. Fig. 15 provides a visual representation of the R-squared scores of all the models tested, allowing for easy comparison of their relative performance.

To further improve prediction accuracy, an ensemble approach was considered. Based on the individual model performance, CatBoost and LightGBM were selected for combination using a Voting Regressor. This technique leverages the strengths of both models, leading to a combined R-squared score of 90% on average, showing that the ensemble approach slightly enhanced the performance over individual models.

During cross-validation, the dataset was split into multiple folds, and each model was trained and evaluated across these different folds to ensure that the results were not influenced by a particular split of the data. This process helps in estimating the model's

generalizability to unseen data. The R-squared scores from cross-validation were plotted for all models, as shown in Fig. 16, to visualize their consistency and robustness across different subsets of the dataset.

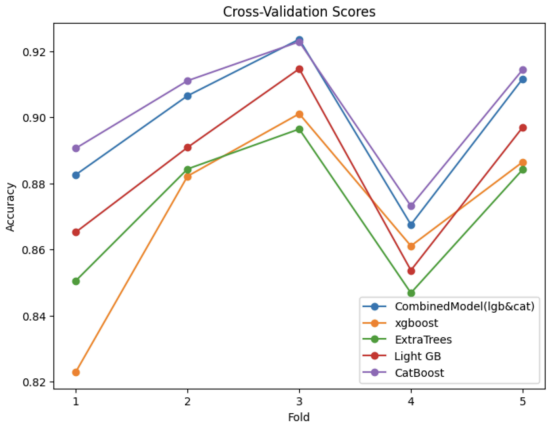

Fig. 16. Cross validation

The graph presented is a line chart depicting the cross-validation scores of various machine learning models across different folds. Each line represents a model: CombinedModel (an ensemble of LightGBM and CatBoost), XGBoost, Extra Trees, LightGBM, and CatBoost. The y-axis shows the accuracy of the models, while the x-axis represents the fold number in the cross-validation process, with values ranging from 1 to 5.

The graph illustrates the effectiveness of the CombinedModel (LightGBM & CatBoost) in achieving more consistent and higher accuracy across cross-validation folds. It suggests that the combination of these two models, leveraging their boosting capabilities, leads to improved performance compared to using them individually. The variability in performance across folds for individual models like LightGBM and XGBoost indicates that while they perform well on some folds, they may be less stable when faced with different subsets of the data, reinforcing the benefit of the ensemble approach for better generalization.

CombinedModel (LightGBM & CatBoost) consistently outperforms the other models in most folds, especially in Fold 2 and Fold 3, where its accuracy peaks at around 0.92, making it the most accurate model overall. It exhibits strong generalization, as its performance remains stable across all folds, except for a slight dip in Fold 4. LightGBM and CatBoost follow closely behind, with LightGBM peaking at around 0.91 in Fold 3 but showing more variation across folds, particularly a noticeable drop in accuracy in Fold 4. Extra Trees and XGBoost show slightly lower performance, with Extra Trees showing the most volatility, dropping below 0.84 in Fold 1 before recovering in subsequent folds.

The final ensemble model, which combined CatBoost and LightGBM, delivered an average R-squared score of 90%. This indicates that the model is able to explain 90% of the variance in house prices, making it a reliable choice for accurate price prediction.

This combination of boosting algorithms like CatBoost and LightGBM proved to be effective due to their ability to handle large datasets, capture nonlinear relationships, and optimize through iterations. The final model not only performed well on the test set but also showed stability during cross-validation, confirming that the model can generalize well to unseen data.

CHAPTER V

CONCLUSION AND WORK SCHEDULE FOR PHASE II

5.1 CONCLUSION

In conclusion, the project successfully demonstrated the effectiveness of combining multiple machine learning models to predict house prices with high accuracy. By evaluating a diverse range of models—including Random Forest, XGBoost, Gradient Boosting, Extra Trees, LightGBM, Hist Gradient Boosting, and CatBoost—it was observed that while each model individually performed well, the best results were achieved through an ensemble of CatBoost and LightGBM. With an R-squared score of 90%, this ensemble approach—which capitalized on the complimentary boosting characteristics of both models—showed that the model was very trustworthy, explaining 90% of the variance in home prices.

The Voting Regressor ensemble technique allowed for the effective combination of each model's predictive capabilities, leading to significantly improved performance compared to individual models. These results were further validated through cross-validation, where the model's stability, reliability, and consistency were confirmed across different data folds, demonstrating its capacity to effectively generalize to unknown data and perform well across varied scenarios.

Additionally, the careful handling of data—including meticulous preprocessing, feature selection, and the use of advanced machine learning techniques—ensured that the model was both accurate and robust. This project demonstrates how effective group learning can be and sophisticated algorithms in tackling complex regression tasks, offering a highly effective, scalable solution for real-world house price prediction and personalized recommendation systems that can assist buyers and investors alike in making well-informed decisions.

5.2 FUTURE ENHANCEMENT

The work schedule for **Phase II** begins with Week 1, focusing on planning and dataset preparation. During this stage, objectives for hyperparameter tuning and the recommendation system are clearly defined. Additional data required for contextual

insights, such as user preferences, market trends, and amenities, is collected, and existing model performance is reviewed to identify parameters for tuning. Strategies for property recommendations, like collaborative filtering or content-based filtering, are outlined to ensure personalized outputs.

In Week 2, hyperparameter tuning techniques such as Grid Search, Random Search, or Bayesian Optimization are implemented. Various hyperparameter combinations for models like LightGBM, CatBoost, and Voting Regressor are tested, and the best-performing configurations are documented. Week 3 shifts focus to the development of the recommendation module. This involves designing a recommendation logic that ranks properties based on predicted price, user preferences, and proximity to amenities, and integrating it seamlessly with the prediction pipeline. Week 4 emphasizes integration and testing, ensuring the tuning and recommendation modules work cohesively with the existing system. End-to-end testing and debugging are conducted to refine the solution. Finally, in Week 5, comprehensive validation is performed, feedback is gathered from domain experts or users, and detailed documentation of workflows and outcomes is prepared, ensuring a polished and ready-to-deploy solution.

APPENDIX I

PUBLICATION STATUS: Submitted to a conference.

TITLE OF THE PAPER: Ensemble-Based House Price Prediction using Boosting Regressors

AUTHORS: S. Senthil Pandi, Madhumitha K, Mohana Prasath G

NAME OF THE CONFERENCES: International Conference On Emerging Research In Computational Science - 2024

APPENDIX II

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sb
from sklearn.metrics import r2_score
from sklearn.model_selection import cross_val_score
import numpy as np
df=pd.read_csv("House_Price.csv")
df.isnull().sum()
df.info()
df.head()
df.describe()
y=df['Price']
df.drop(['id','Date'],axis=1,inplace=True)
X=df.drop(['Price'],axis=1)
pip install catboost
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
from xgboost import XGBRegressor
from sklearn.ensemble import GradientBoostingRegressor
from sklearn.ensemble import ExtraTreesRegressor
import lightgbm as lgb
from sklearn.ensemble import HistGradientBoostingRegressor
from sklearn.ensemble import VotingRegressor
from catboost import CatBoostRegressor
X_train,X_val,y_train,y_val=train_test_split(X,y,test_size=0.2)

rf=RandomForestRegressor()
rf.fit(X_train,y_train)
r1=rf.score(X_val,y_val)

xgb=XGBRegressor()
```

```
xgb.fit(X_train,y_train)
r2=xgb.score(X_val,y_val)

gb=GradientBoostingRegressor()
gb.fit(X_train,y_train)
r3=gb.score(X_val,y_val)

et = ExtraTreesRegressor(n_estimators=100)
et.fit(X_train, y_train)
r4=et.score(X_val,y_val)

lgb_r = lgb.LGBMRegressor()
lgb_r.fit(X_train, y_train)
r5=lgb_r.score(X_val,y_val)

hist = HistGradientBoostingRegressor()
hist.fit(X_train,y_train)
r6=hist.score(X_val,y_val)

cat=CatBoostRegressor()
cat.fit(X_train,y_train)
r7=cat.score(X_val,y_val)

estim=[('m5',lgb_r),('m7',cat)]

model=VotingRegressor(estimators=estim)
model.fit(X_train,y_train)
r_final=model.score(X_val,y_val)

cv_scores = cross_val_score(model, X, y, cv=5, scoring='r2')
print("Cross-validated R-squared: ", cv_scores.mean())

plt.figure(figsize=(8, 6))
```

```
plt.plot(range(1, 6), cv_scores, marker='o', linestyle='-', color='b')
plt.title('Cross-Validation Scores for Voting Classifier')
plt.xlabel('Fold')
plt.ylabel('Accuracy')
plt.xticks(range(1, 6))
plt.show()


predictions=model.predict(X_val)
r2_s = r2_score(y_val, predictions)
print(f"R-squared: {r2}")

a=np.array([r1,r2,r3,r4,r5,r6,r7,r_final])
b=np.array(['Random\nForest','XGBoost','Gradient\nBoosting','Extra\nTrees','LightGB','His
tGB','CatBoost','Voting\nClassifier'])
plt.figure(figsize=(9,5))
plt.xlabel('ML Algorithms')
plt.ylabel('R2 Score')
plt.plot(b,a,marker = 'o')
plt.show()

import seaborn as sns
import matplotlib.pyplot as plt

plt.figure(figsize=(12, 8))
sns.heatmap(df.corr(), annot=True, cmap='YlGnBu')
plt.show()

from sklearn.feature_selection import mutual_info_regression
mi = mutual_info_regression(X_train, y_train)
plt.barh(X_train.columns, mi)
plt.show()
```

```
cv_rf = cross_val_score(rf, X, y, cv=5, scoring='r2')
print("Cross-validated R-squared: ", cv_rf.mean())

cv_xgb = cross_val_score(xgb, X, y, cv=5, scoring='r2')
print("Cross-validated R-squared: ", cv_xgb.mean())

cv_gb = cross_val_score(gb, X, y, cv=5, scoring='r2')
print("Cross-validated R-squared: ", cv_gb.mean())

cv_et = cross_val_score(et, X, y, cv=5, scoring='r2')
print("Cross-validated R-squared: ", cv_et.mean())

cv_lgb = cross_val_score(lgb_r, X, y, cv=5, scoring='r2')
print("Cross-validated R-squared: ", cv_lgb.mean())

cv_hist = cross_val_score(hist, X, y, cv=5, scoring='r2')
print("Cross-validated R-squared: ", cv_hist.mean())

cv_cat = cross_val_score(cat, X, y, cv=5, scoring='r2')
print("Cross-validated R-squared: ", cv_cat.mean())

plt.figure(figsize=(8, 6))
plt.plot(range(1, 6), cv_scores, marker='o', linestyle='-',label="CombinedModel(lgb&cat)")
#plt.plot(range(1, 6), cv_rf, marker='o', linestyle='-',label="rf")
plt.plot(range(1, 6), cv_xgb, marker='o', linestyle='-',label="xgboost")
#plt.plot(range(1, 6), cv_gb, marker='o', linestyle='-',label="gb")
plt.plot(range(1, 6), cv_et, marker='o', linestyle='-',label="ExtraTrees")
plt.plot(range(1, 6), cv_lgb, marker='o', linestyle='-',label="Light GB")
#plt.plot(range(1, 6), cv_hist, marker='o', linestyle='-',label="hist")
plt.plot(range(1, 6), cv_cat, marker='o', linestyle='-',label="CatBoost")
plt.title('Cross-Validation Scores')
plt.xlabel('Fold')
plt.ylabel('Accuracy')
```

```
plt.xticks(range(1, 6))
plt.legend()
plt.show()

c=np.array([cv_rf.mean(),cv_xgb.mean(),cv_gb.mean(),cv_et.mean(),cv_lgb.mean(),cv_his
t.mean(),cv_cat.mean(),cv_scores.mean()])

plt.figure(figsize=(8, 6))
plt.plot(b, c, marker='o', linestyle='-')
plt.title('Cross-Validation Scores for Voting Classifier')
plt.xlabel('Fold')
plt.ylabel('Accuracy')
plt.legend()
plt.show()
```

REFERENCES

[1] Hjort, Anders, Ida Scheel, Dag Einar Sommervoll, and Johan Pensar. "Locally interpretable tree boosting: An application to house price prediction." *Decision Support Systems* 178 (2024): 114106.

[2] Adetunji, Abigail Bola, Oluwatobi Noah Akande, Funmilola Alaba Ajala, Ololade Oyewo, Yetunde Faith Akande, and Gbenle Oluwadara. "House price prediction using random forest machine learning technique." *Procedia Computer Science* 199 (2022): 806-813.

[3] Wang, Lu, Guangxing Wang, Huan Yu, and Fei Wang. "Prediction and analysis of residential house price using a flexible spatiotemporal model." *Journal of Applied Economics* 25, no. 1 (2022): 503-522.

[4] Hjort, Anders, Johan Pensar, Ida Scheel, and Dag Einar Sommervoll. "House price prediction with gradient boosted trees under different loss functions." *Journal of Property Research* 39, no. 4 (2022): 338-364.

[5] Liu, Jun, and Zihan Ma. "Forecasting Housing Price Using GRU, LSTM and Bi-LSTM for California." In *2024 IEEE 2nd International Conference on Control, Electronics and Computer Technology (ICCECT)*, pp. 1033-1037. IEEE, 2024.

[6] Jiang, Zhongyun, and Guoxin Shen. "Prediction of house price based on the back propagation neural network in the keras deep learning framework." In *2019 6th International Conference on Systems and Informatics (ICSAI)*, pp. 1408-1412. IEEE, 2019.

[7]Özöğür Akyüz, Süreyya, Birsen Eygi Erdogan, Özlem Yıldız, and Pınar Karadayı Ataş. "A novel hybrid house price prediction model." Computational economics 62, no. 3 (2023): 1215-1232.

[8] Lahmiri, Salim, Stelios Bekiros, and Christos Avdoulas. "A comparative assessment of machine learning methods for predicting housing prices using Bayesian optimization." *Decision Analytics Journal* 6 (2023): 100166.

[9] Y. Zhao, J. Zhao and E. Y. Lam, "House Price Prediction: A Multi-Source Data Fusion Perspective," in Big Data Mining and Analytics, vol. 7, no. 3, pp. 603-620, September 2024.

[10] F. Wang, Y. Zou, H. Zhang and H. Shi, "House Price Prediction Approach based on Deep Learning and ARIMA Model," *2019 IEEE 7th International Conference on Computer Science and Network Technology (ICCSNT)*, Dalian, China, 2019.

[11] B. U. Sri, C. S. K. Reddy, C. R. Kumar, A. Vyshnavi, B. Vinod and B. K. Reddy, "Random Forest-based House Price Prediction," 2023 Second International Conference on Augmented Intelligence and Sustainable Systems (ICAISS), Trichy, India, 2023

[12]S. Goswami, V. S. Bramhe and S. Khepra, "Prediction of House Price Using Stacked LSTM Model," 2022 4th International Conference on Advances in Computing, Communication Control and Networking (ICAC3N), Greater Noida, India, 2022.

[13]Y. Chen, R. Xue and Y. Zhang, "House price prediction based on machine learning and deep learning methods," 2021 International Conference on Electronic Information Engineering and Computer Science (EIECS), Changchun, China, 2021

[14]Ho, W. K. O., Tang, B.-S. and Wong, S. W. (2020) 'Predicting property prices with machine learning algorithms', Journal of Property Research, 38(1), pp. 48–70. doi: 10.1080/09599916.2020.1832558.

Madhu

ORIGINALITY REPORT

14%	9%	10%	9%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to National College of Ireland Student Paper	1%
2	Submitted to University of Cincinnati Student Paper	1%
3	Submitted to CSU, San Jose State University Student Paper	1%
4	Shahriar Afandizadeh, Farhad Sedighi, Navid Kalantari, Hamid Mirzahossein. "Modeling of the effect of transportation system accessibility on residential real estate prices: The case of Washington metropolitan area, USA", Case Studies on Transport Policy, 2024 Publication	1%
5	www.researchgate.net Internet Source	1%
6	Submitted to Georgia Institute of Technology Main Campus Student Paper	1%
7	ntnuopen.ntnu.no Internet Source	1%

Ensemble-Based House Price Prediction using Boosting Regressors

Senthil Pandi S
Department of CSE
Rajalakshmi Engineering College
Chennai, India
mailtosenthil.ks@gmail.com

Madhumitha K
Department of CSE, REC
Chennai, India
210701141@rajalakshmi.edu.in

Mohana Prasath G
Department of CSE, REC
Chennai, India
210701163@rajalakshmi.edu.in

Abstract— Prospective homeowners and real estate investors are always on the lookout for solutions that deliver precise, data-driven, and highly customized insights into property prices, especially in the dynamic and ever-fluctuating real estate market. This project centers on the development of a powerful machine learning (ML) website that forecasts home values based on a wide range of user-specified parameters, including location (latitude and longitude), property size (square footage), and configuration (BHK - Bedrooms, Hall, Kitchen). The platform utilizes a vast dataset that not only encompasses geographic data and real estate listings but also integrates additional external data sources, such as current market trends, local amenities, neighborhood features, economic conditions, and historical pricing patterns. By combining these rich data sources with advanced machine learning algorithms, the project aims to create a highly effective tool for optimizing property assessment, improving real estate market insights, and empowering users to make well-informed, strategic decisions in buying or investing in property.

***Keywords*— House Price Prediction, Ensemble Technique, Boosting**

I. Introduction

Propoerty price prediction is a important aspect of the real estate market, influencing decisions made by homebuyers, investors, developers, and policymakers. Accurate prediction models can provide valuable insights into market trends, helping stakeholders make informed decisions. Given the complexity of real estate data—characterized by a multitude of factors such as location, economic conditions, and property features—developing reliable predictive models is challenging yet essential.

This project focuses on leveraging advanced machine learning techniques to predict house prices with greater accuracy. By putting forward novel ideas that increase prediction accuracy and provide useful applications for the real estate sector, we hope to support the ongoing efforts in the field.

Accurately estimating property values in the quickly changing real estate market continues to be a major obstacle for potential homeowners and investors. Conventional property valuation techniques frequently lack accuracy and don't take into consideration the intricate interactions between many elements that affect property values. Developing a predictive model that successfully combines various data sources and cutting-edge machine learning algorithms to produce accurate and customized property price forecasts is the main difficulty. This initiative is driven by the growing need in a turbulent real estate market for more precise and customized real estate information. This project intends to develop an efficient ML-based platform by utilizing a large dataset that contains real estate listings, geographic data, and additional external aspects like market trends and local amenities.

To develop an advanced machine learning (ML)-based website designed to accurately forecast property values by leveraging cutting-edge algorithms and comprehensive datasets. The research goal is to develop a prediction model by combining information from many sources, such as real estate listings, geographic information (latitude, longitude), property features (BHK - Bedrooms, Hall, Kitchen), and additional external factors such as market trends and local amenities. The website will have an easy-to-use design that lets visitors explore property alternatives, enter search parameters, and get forecasts that are specific to them. The ultimate goal is to deliver a valuable resource for homeowners, investors, and real estate professionals, enabling more precise property assessments.

II. Literature Survey

Anders Hjort et al.,[1] proposed a tree boosting model, Locally Interpretable Tree Boosting (LitBoost) that combines the strengths of Generalized Additive models(GAM) and Gradient Boosted Trees(GBT). It improves predictive performance by avoiding overfitting and also enhances predictability. It was observed that GAM displayed poor performance when the observations were small. The proposed method LitBoost outperforms GAM when the number of observations per group is small and performs better than GBT when specific interactions are included in the data1. The performance of Random Forest for house price prediction and have shown that the model is able to achieve a minor difference. Lu Wang et al. proposed a spatiotemporal model (FSTM) to predict the prices of houses using the spatiotemporal characteristics of small cities in China. The model was designed to handle the relationship between space and time reflecting the variation in house prices across different locations and time periods. A flexible structure that combined the long term trends with spatially correlated random factors enabled the model to predict accurately3.

Anders Hjort et al., [2] introduced an improvement in XGBoost by introducing a loss function Squared percentage error (SPE). The introduction of the SPE loss function improves model performance from 88.24% to 91.02% under the 22% measure. XGBoost-SPE reduces incorrect predictions by 4.9% compared to XGBoost-SE, particularly improving in lower price segments, though it performs slightly worse in higher price segments. Combining both models into a hybrid yields even better results, achieving 90.4% accuracy and further reducing errors by 9.3%, demonstrating the effectiveness of this combined approach.

Jun Liu and Zihan Ma [3] proposed a hybrid method by combining advanced neural network models with spatial analysis techniques. that includes spatial factors in house price predictions. This integration not only improves

predictive accuracy but also provides a new perspective on incorporating geographical factors into real estate forecasting. To forecast the cost of used homes in Sanghai, Zhongyun Jiang et al., [4] suggested a multilayer feedforward neural network trained using the error inverse propagation approach. SCR is a hybrid approach that was suggested by Sureyya Ozogur Akyur et al., [5] that combines support vector regression, closest neighbor classification, clustering analysis, and linear regression. One method's output is used as another's input. The research will classify the houses for which the cluster is unknown, generate distinct housing clusters using the data at hand, and anticipate prices by developing distinct prediction models for each class in order to provide a hybrid method. Salim Lahmiri et al., [6] used Gaussian process regression, support vector regression, and ensemble regression trees to forecast housing prices in Taiwan. The experimental results showed that boosting ensemble regression trees performed better than Gaussian process regression and support vector regression. All three prediction algorithms fared better than artificial neural networks.

In 2024, Y Zhao et al.,[7] explored the application of multiple machine learning models, including MLP, SVM, Linear Regression, XGBoost, and Random Forest. Their focus was on evaluating the models' accuracy in predictive tasks, highlighting the effectiveness of combining various algorithms for improved performance. In 2019, Feng Wang et al.,[8] investigated the use of SVM and ARIMA models for predictive analysis. They evaluated the accuracy, aiming to enhance the reliability of their predictions in various applications. Wang et al.,[9] conducted a comprehensive study utilizing various machine learning models, including Random Forest Algorithm, Lasso Regression, Linear Regression, and Decision Tree. Advanced neural network architectures, including LSTM, CNN, and RNN, to enhance predictive accuracy in their study. Training Loss, providing valuable insights into their effectiveness and reliability in handling complex data patterns.

III. METHODOLOGY

The proposed model follows a structured workflow, the process involves Data Preprocessing, where the data is cleaned, transformed, and made suitable for modeling. In the Model Selection phase, different machine learning models are evaluated to determine the best fit for the problem. Once the model is chosen, Model Training takes place, allowing the model to learn from the data. This is followed by Validation and Testing to assess the model's performance and fine-tune it for optimal results. Finally, the trained model is used to Predict the Output, providing accurate forecasts for house prices based on user input parameters.

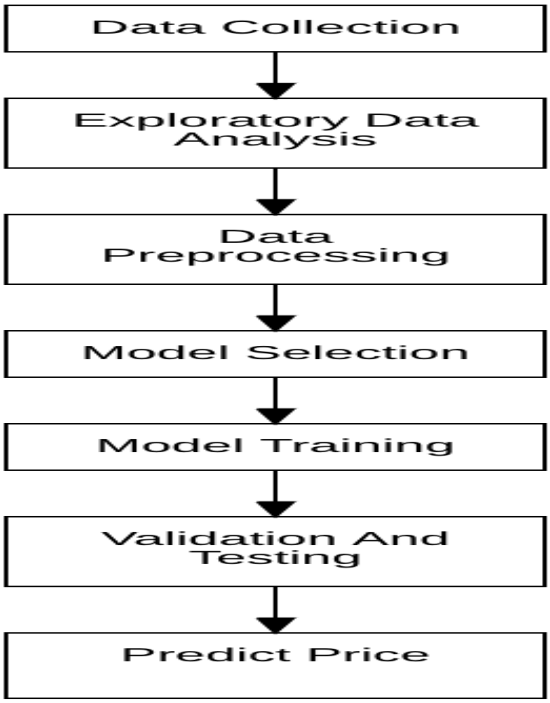

Figure. 1. Workflow model

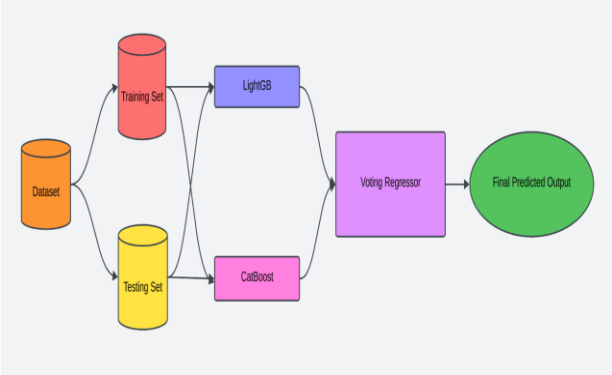

Fig. 2. Architecture Diagram

This architecture diagram presents a comprehensive workflow for predicting house prices using an ensemble of two powerful machine learning models - LightGBM and CatBoost - combined through a Voting Regressor. The system is structured to maximize predictive accuracy by integrating the strengths of both models within a carefully organized pipeline, which begins with data collection and proceeds through multiple stages of data preparation, model training, and prediction generation.

The first stage of the system is Data Collection, where a dataset is compiled with key property attributes, such as location (latitude and longitude), property size (in square feet or square meters), and configuration (number of bedrooms, hall, and kitchen). These features are essential for accurate house price prediction, as they capture both the physical characteristics of the property and its geographic location.

Once the data is prepared, the Model Training phase begins. Here, two machine learning models - LightGBM and CatBoost - are trained on the Training Set. LightGBM, or

Light Gradient Boosting Machine, is a high-performance gradient boosting algorithm that is optimized for speed and efficiency. It handles large datasets well and uses a leaf-wise growth strategy that can reduce computational costs, making it ideal for this application. CatBoost, short for Categorical Boosting, is another gradient boosting algorithm known for its ability to work effectively with categorical features and handle complex relationships within the data with minimal preprocessing. Both models independently learn patterns in the data to predict house prices.

After individual training, the outputs of LightGBM and CatBoost are combined using a Voting Regressor, an ensemble technique designed to improve predictive accuracy. The Voting Regressor consolidates predictions from both models, either by averaging them or assigning weights to each model's output based on their individual performance. This approach capitalizes on the complementary strengths of LightGBM and CatBoost: while LightGBM's efficiency makes it strong in handling high-dimensional data, CatBoost excels in managing categorical variables and complex interactions. By integrating both models, the Voting Regressor reduces errors associated with single-model predictions and enhances the system's robustness.

In the final stage, the Voting Regressor outputs a Final Predicted Output—the house price estimate. This output represents the combined predictive capability of both LightGBM and CatBoost, offering an accurate and reliable estimate. Such a prediction is invaluable to users like prospective homeowners and real estate investors, who can use this information to make informed decisions based on current market trends and specific property characteristics. Additionally, the Voting Regressor's ensemble approach ensures that the final prediction is not only accurate but also generalizes well to new data, reducing the risk of overfitting.

In summary, this architecture leverages the strengths of LightGBM and CatBoost to provide a robust and accurate system for house price prediction. By combining these models in a Voting Regressor ensemble, the system benefits from a balanced approach that draws on the best qualities of both algorithms. This comprehensive and systematic design ensures the system's reliability, making it a powerful tool for real-world applications in real estate analytics. The architecture's structured workflow - encompassing data collection, model training, ensemble prediction, and output generation - demonstrates a thoughtful approach to solving complex regression problems in the domain of property price prediction.

IV. Results & Discussions

The dataset was gathered from Kaggle. The dataset used in this study came from Kaggle, a website that offers a wide variety of datasets for analysis and research. ID, are some of the input features that make up the dataset. "Price" is the label assigned to the target variable, or output feature, which represents the house's price. The dataset consists of 14620 rows and 23 columns, including the output feature shown in figure.3.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 14620 entries, 0 to 14619
Data columns (total 23 columns):
 #   Column                                 Non-Null Count  Dtype
---  ------                                 --------------  -----
 0   id                                     14620 non-null  int64
 1   Date                                   14620 non-null  int64
 2   number of bedrooms                     14620 non-null  int64
 3   number of bathrooms                    14620 non-null  float64
 4   living area                            14620 non-null  int64
 5   lot area                               14620 non-null  int64
 6   number of floors                       14620 non-null  float64
 7   waterfront present                     14620 non-null  int64
 8   number of views                        14620 non-null  int64
 9   condition of the house                 14620 non-null  int64
 10  grade of the house                     14620 non-null  int64
 11  Area of the house(excluding basement)  14620 non-null  int64
 12  Area of the basement                   14620 non-null  int64
 13  Built Year                             14620 non-null  int64
 14  Renovation Year                        14620 non-null  int64
 15  Postal Code                            14620 non-null  int64
 16  Lattitude                              14620 non-null  float64
 17  Longitude                              14620 non-null  float64
 18  living_area_renov                      14620 non-null  int64
 19  lot_area_renov                         14620 non-null  int64
 20  Number of schools nearby               14620 non-null  int64
 21  Distance from the airport              14620 non-null  int64
 22  Price                                  14620 non-null  int64
dtypes: float64(4), int64(19)
memory usage: 2.6 MB
```

Figure.3. Dataset Description

Exploratory Data Analysis- Initially, the dataset should be analyzed to extract meaningful insights. The number of attributes, relationships between features, and the relationship between each input attribute and the output feature are evaluated to better understand the data. This step provides a foundation for model training. Here, techniques like the Confusion matrix, box plot and mutual information for regression are used to analyze the data. Confusion matrices show the relationship of each attribute in the dataset with every other attribute in the dataset.

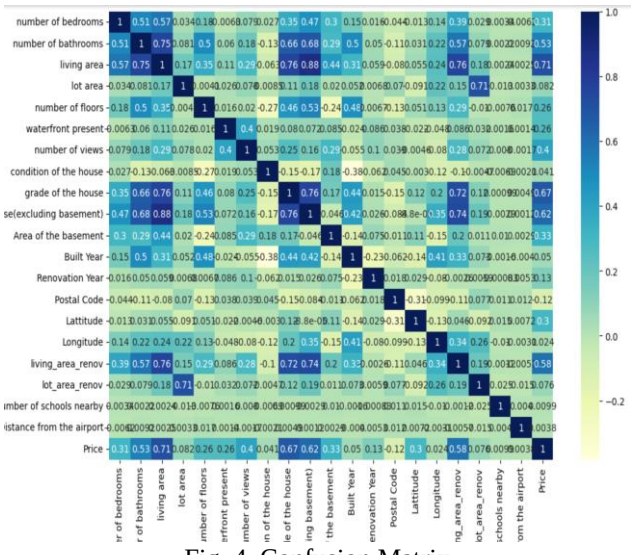

Fig. 4. Confusion Matrix

In the above confusion matrix in Fig 4, the output feature Price has a strong positive correlation with the living area. The price also has a strong positive correlation with grade of the house and area of the house (excluding basement). The Price has a weak negative correlation with Postal code.

A box plot helps to visualize the distribution of numerical data, highlighting any outliers, and can be used with both numerical and categorical features. It provides insights into data symmetry, spread, and skewness.

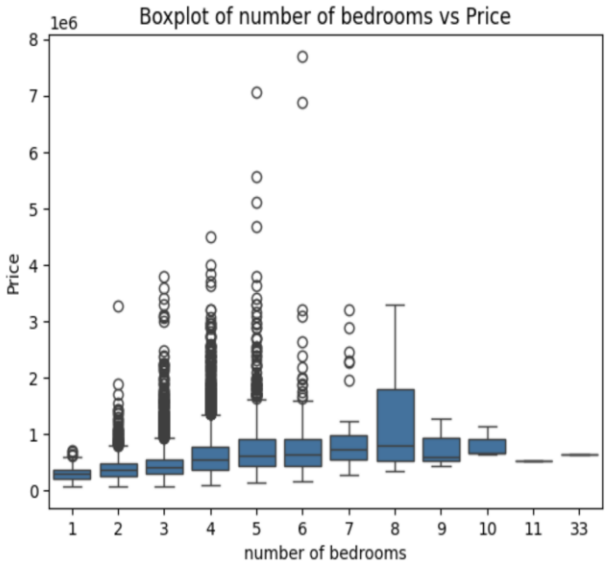

Fig. 5. Box Plot of Number of bedrooms Vs Price

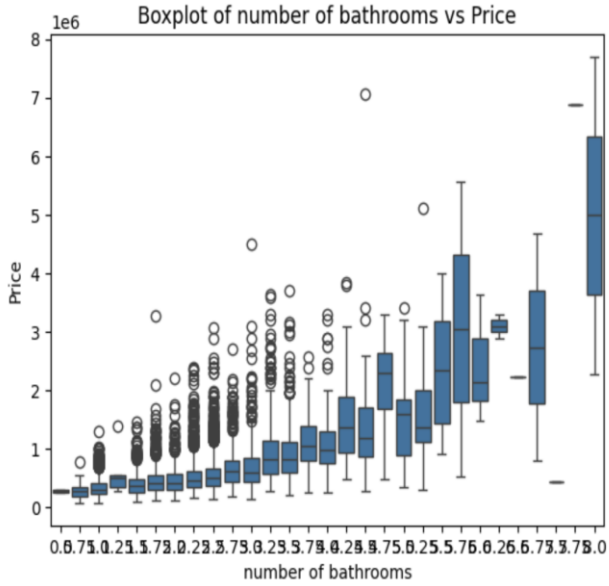

Fig. 6. Box Plot of Number of bathrooms Vs Price

Fig 5 represents the box plot of Price versus number of bedrooms. The price of the home somewhat rises as the number of bedrooms increases. Fig 6 represents the box plot of price versus number of bathrooms. The number of bathrooms and the price have a linear connection. The cost likewise rises linearly with the number of bathrooms.

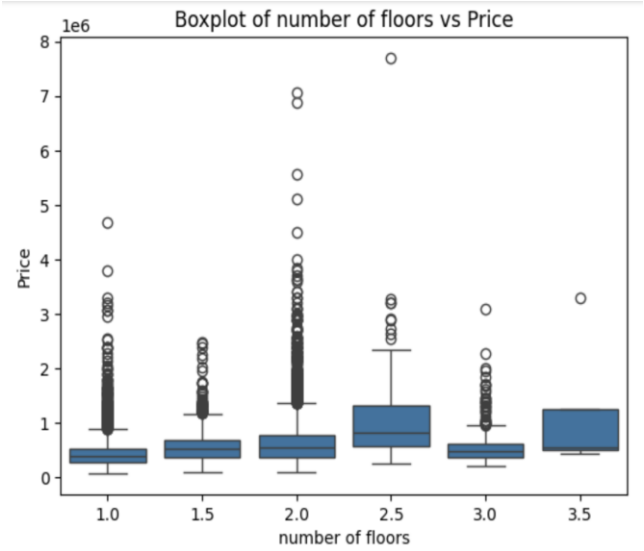

Fig. 7. Box Plot of Number of floors Vs Price

Fig 7 represents the box plot of Price versus number of floors. There is no relation between the number of floors and the price of the price.Fig 8 represents the box plot of condition of the house versus the price. House condition doesn't seem to have a strong linear relationship with the median price. Although better conditions (4 and 5) have more expensive houses, the price distribution is still concentrated towards lower values with a few outliers being much higher.

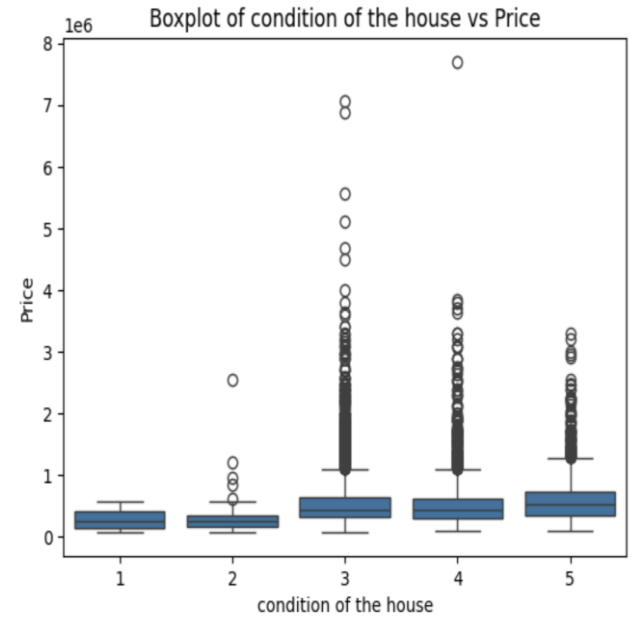

Fig. 8. Box Plot of Condition of the house Vs Price

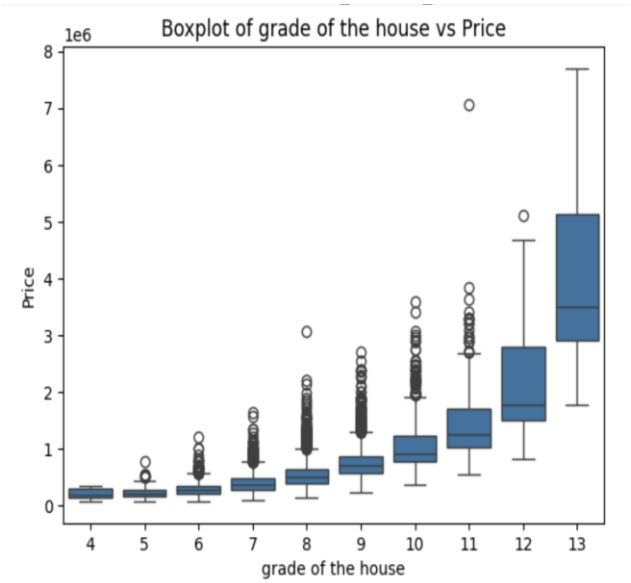

Fig. 9. Box Plot of Grade of the house Vs Price

The box plot showing the house's grade against price is shown in Fig. 9. The grade of the house and the property's price have a strong linear relationship. Mutual information measures the dependency between variables, giving an idea of how much information about the target is provided by each feature. Here postal code gives the high mutual information. Other strong mutual information are given by the features - Living area, Grade of the house, Latitude and living_area_renov.

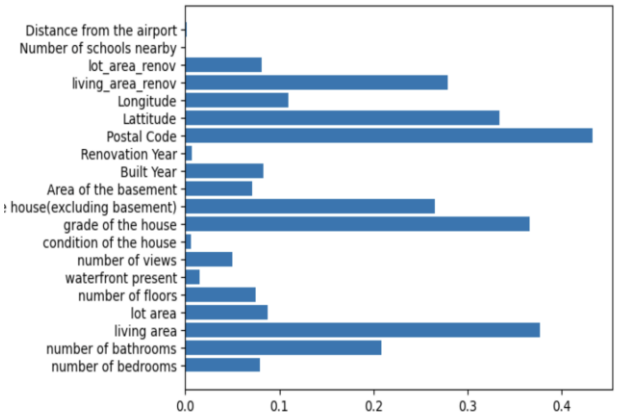

Fig. 10. Mutual Information

Dataset Preprocessing- Before proceeding with model training, it's essential to check for null values and categorical variables in the dataset. Null values can either be removed or filled with appropriate default values (e.g., mean, median, or mode). It is necessary to translate categorical variables into numerical values, frequently with the aid of methods like Label Encoding or One-Hot Encoding. This ensures that the data is clean and ready for the machine learning algorithms, which typically require numerical inputs.

Model Selection and Model Training- In the model selection phase, several machine learning algorithms were evaluated to determine the best approach for predicting house prices. The models considered included Random Forest, XGBoost, Gradient Boosting, Extra Trees, LightGBM, Hist Gradient Boosting, and CatBoost. These models were chosen due to their strength in handling complex, non-linear relationships, making them well-suited for a regression problem like house price prediction.

Each model was trained on the dataset after performing feature selection and preprocessing steps such as handling missing values and encoding categorical variables. The training process involved feeding the models with the input features and their corresponding house prices (target variable). Hyperparameter tuning was also conducted to optimize the performance of each model, ensuring that they could learn patterns in the data effectively.

After training, the performance of each model was evaluated based on the R-squared score, which measures how well the model predicts the variance in house prices. The results indicated strong performance across the board, with CatBoostand LightGBM standing out with the highest R-squared scores.

In the next step, an ensemble method using a Voting Regressor was employed to combine the predictions of the top-performing models—CatBoost and LightGBM. This technique allowed the model to take advantage of the strengths of both algorithms, resulting in improved predictive accuracy and robustness.

The final trained model was then validated using cross-validation, confirming that it generalized well to unseen data, making it suitable for real-world house price prediction tasks.

In the testing phase, several machine learning models were trained and evaluated based on their R-squared scores, which measure how well each model explains the variance in the house price data. The individual R-squared scores were as follows: 88% for Random Forest, 87% for XGBoost, 86% for Gradient Boosting, 88% for Extra Trees, 89% for LightGBM, 87% for Hist Gradient Boosting, and 90% for CatBoost.

These results show that while each model performed well in predicting house prices, CatBoost achieved the highest R-squared score of 90%, followed closely by LightGBM at 89%. Fig. 11 provides a visual representation of the R-squared scores of all the models tested, allowing for easy comparison of their relative performance.

To further improve prediction accuracy, an ensemble approach was considered. Based on the individual model performance, CatBoost and LightGBM were selected for combination using a Voting Regressor. This technique leverages the strengths of both models, leading to a combined R-squared score of 90% on average, showing that the ensemble approach slightly enhanced the performance over individual models.

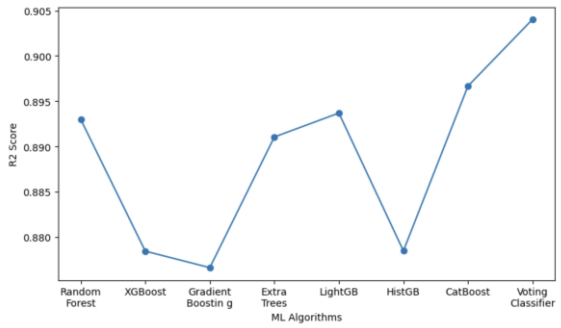

Fig. 11. R2 score of all models

During cross-validation, the dataset was split into multiple folds, and each model was trained and evaluated across these different folds to ensure that the results were not influenced by a particular split of the data. This process helps in estimating the model's generalizability to unseen data. The R-squared scores from cross-validation were plotted for all models, as shown in Fig. 12, to visualize their consistency and robustness across different subsets of the dataset.

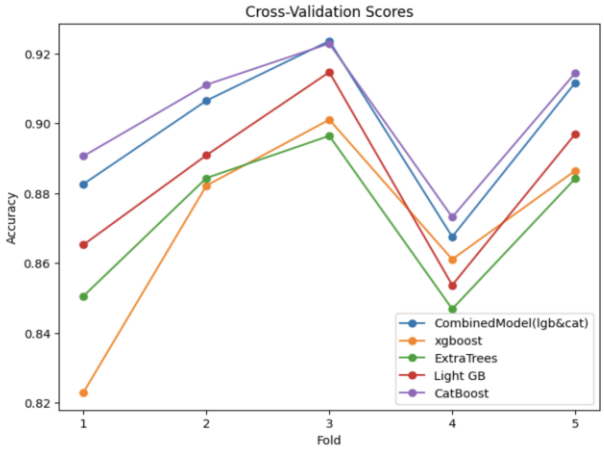

Fig. 12. Cross validation

The graph presented is a line chart depicting the cross-validation scores of various machine learning models across different folds. Each line represents a model: CombinedModel (an ensemble of LightGBM and CatBoost), XGBoost, Extra Trees, LightGBM, and CatBoost. The y-axis shows the accuracy of the models, while the x-axis represents the fold number in the cross-validation process, with values ranging from 1 to 5.

The graph illustrates the effectiveness of the CombinedModel (LightGBM & CatBoost) in achieving more consistent and higher accuracy across cross-validation folds. It suggests that the combination of these two models, leveraging their boosting capabilities, leads to improved performance compared to using them individually. The variability in performance across folds for individual models like LightGBM and XGBoost indicates that while they perform well on some folds, they may be less stable when faced with different subsets of the data, reinforcing the benefit of the ensemble approach for better generalization.

CombinedModel (LightGBM & CatBoost) consistently outperforms the other models in most folds, especially in Fold 2 and Fold 3, where its accuracy peaks at around 0.92, making it the most accurate model overall. It exhibits strong generalization, as its performance remains stable across all folds, except for a slight dip in Fold 4. LightGBM and CatBoost follow closely behind, with LightGBM peaking at around 0.91 in Fold 3 but showing more variation across folds, particularly a noticeable drop in accuracy in Fold 4. Extra Trees and XGBoost show slightly lower performance, with Extra Trees showing the most volatility, dropping below 0.84 in Fold 1 before recovering in subsequent folds.

The final ensemble model, which combined CatBoost and LightGBM, delivered an average R-squared score of 90%. This indicates that the model is able to explain 90% of the variance in house prices, making it a reliable choice for accurate price prediction.

This combination of boosting algorithms like CatBoost and LightGBM proved to be effective due to their ability to handle large datasets, capture nonlinear relationships, and optimize through iterations. The final model not only performed well on the test set but also showed stability during cross-validation.

V. Conclusion

By evaluating a diverse range of models—including Random Forest, XGBoost, Gradient Boosting, Extra Trees, LightGBM, Hist Gradient Boosting, and CatBoost—it was observed that while each model individually performed well, the best results were achieved through an ensemble of CatBoost and LightGBM. With an R-squared score of 90%, this ensemble approach—which capitalized on the complimentary boosting characteristics of both models—showed that the model was very trustworthy, explaining 90% of the variance in home prices. The Voting Regressor ensemble technique allowed for the effective combination of each model's predictive capabilities, leading to significantly improved performance compared to individual models. These results were further validated through cross-validation, where the model's stability, reliability, and consistency were confirmed across different data folds, demonstrating its capacity to effectively generalize to unknown data and perform well across varied scenarios. Additionally, the careful handling of data—including meticulous preprocessing, feature selection, and the use of advanced machine learning techniques—ensured that the model was both accurate and robust. This research demonstrates how effective group learning can be and sophisticated algorithms in tackling complex regression tasks, offering a highly effective, scalable solution for real-world house price prediction and personalized recommendation systems that can assist buyers and investors alike in making well-informed decisions.

References

[1] Hjort, Anders, Ida Scheel, Dag Einar Sommervoll, and Johan Pensar. "Locally interpretable tree boosting: An application to house price prediction." Decision Support Systems 178 (2024): 114106.

[2] Anders Hjort, Johan Pensar, Ida Scheel, and Dag Einar Sommervoll. "House price prediction with gradient boosted trees under different loss functions." Journal of Property Research 39, no. 4 (2022): 338-364.

[3] Jun Liu, and Zihan Ma. "Forecasting Housing Price Using GRU, LSTM and Bi-LSTM for California." In 2024 IEEE ICCECT, pp. 1033-1037. IEEE, 2024.

[4] Jiang, Zhongyun, and Guoxin Shen. "Prediction of house price based on the back propagation neural network in the keras deep learning framework." In 2019 ICSAI, pp. 1408-1412. IEEE, 2019.
[5] Süreyya Özöğür Akyüz, Birsen Eygi Erdogan, Özlem Yıldız, and Pınar Karadayı Ataş. "A novel hybrid house price prediction model." Computational economics 62, no. 3 (2023): 1215-1232.
[6] Salim Lahmiri, Stelios Bekiros, and Christos Avdoulas. "A comparative assessment of machine learning methods for predicting housing prices using Bayesian optimization." Decision Analytics Journal 6 (2023): 100166.
[7] Y. Zhao, J. Zhao and E. Y. Lam, "House Price Prediction: A Multi-Source Data Fusion Perspective," in Big Data Mining and Analytics, vol. 7, no. 3, pp. 603-620, September 2024.
[8] F. Wang, Y. Zou, H. Zhang and H. Shi, "House Price Prediction Approach based on Deep Learning and ARIMA Model," 2019 IEEE ICCSNT, Dalian, China, 2019.
[9] Wang, Lu, Guangxing Wang, Huan Yu, and Fei Wang. "Prediction and analysis of residential house price using a flexible spatiotemporal model." Journal of Applied Economics 25, no. 1 (2022): 503-522.
[10] Adetunji, Abigail Bola, Oluwatobi Noah Akande, Funmilola Alaba Ajala, Ololade Oyewo, Yetunde Faith Akande, and Gbenle Oluwadara. "House price prediction using random forest machine learning technique." Procedia Computer Science 199 (2022): 806-813.

Madhu

ORIGINALITY REPORT

9%	8%	10%	7%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to National College of Ireland Student Paper	1%
2	Submitted to University of Cincinnati Student Paper	1%
3	Submitted to CSU, San Jose State University Student Paper	1%
4	Shahriar Afandizadeh, Farhad Sedighi, Navid Kalantari, Hamid Mirzahossein. "Modeling of the effect of transportation system accessibility on residential real estate prices: The case of Washington metropolitan area, USA", Case Studies on Transport Policy, 2024 Publication	1%
5	www.researchgate.net Internet Source	1%
6	Submitted to Georgia Institute of Technology Main Campus Student Paper	1%
7	ntnuopen.ntnu.no Internet Source	1%