House Price Prediction

Yash Midha (22BDS0440), Santosh Acharya(22BDS0441)
SCOPE, Vellore institute of technology, Vellore

Abstract **-** This study presents a machine learning-based approach for predicting house prices using a comprehensive dataset sourced from Kaggle. The project focuses on evaluating various regression models to determine the most accurate and efficient technique for price prediction. The data was preprocessed by handling missing values, removing irrelevant features, scaling numerical attributes, and splitting into training and testing sets. Five regression models—Linear Regression, Decision Tree, Random Forest, XGBoost, and Neural Network—were implemented and evaluated using standard metrics such as R-squared, Mean Absolute Error, and Mean Squared Error. Among these, the XGBoost model achieved the highest accuracy with an R-squared score of 89 percent, demonstrating its effectiveness in capturing complex feature interactions. The novelty of this work lies in the integration of SQL for data handling, extensive model comparison, and its readiness for deployment through a Python-based backend. This makes the project suitable for real-world applications, such as online property platforms or housing advisory tools.

Index Terms **-** data preprocessing, house price prediction, machine learning, regression models, XGBoost

I. INTRODUCTION

The real estate market plays a pivotal role in shaping individual financial decisions and broader economic trends, highlighting the critical importance of accurate house price prediction. Reliable predictions can empower buyers to make informed investments, assist sellers in pricing competitively, and guide investors in maximizing returns. With the increasing availability of housing data and advancements in computational methods, machine learning has emerged as a powerful tool to forecast real estate prices more precisely than traditional approaches. This project leverages supervised machine learning algorithms to predict house prices using a well-structured dataset obtained from Kaggle, incorporating key features such as location, property size, number of rooms, and neighborhood quality. A detailed data preprocessing pipeline was developed to handle missing values, encode categorical features, and split the data for training and evaluation. The core of the study revolves around implementing and comparing several regression models, including Linear Regression, Decision Tree, Random Forest, XGBoost, and Neural Networks, to assess their performance and identify the most effective one. Furthermore, the system is built using Python for modeling and SQL for structured data handling, ensuring it is scalable and suitable for deployment in real-world housing platforms and analytics systems.

II. LITERATURE REVIEW

Numerous studies have explored the use of machine learning for house price prediction, highlighting a variety of models, feature selection techniques, and deployment strategies. This section reviews ten relevant works in this domain. In [1], the authors present a machine learning approach that applies multiple regression models, including Linear Regression, Decision Tree, and Random Forest. The study emphasizes feature engineering and hyperparameter tuning, resulting in improved model accuracy. However, it lacks real-time deployment integration. The study in [2] details how to convert machine learning models into deployable APIs using Python frameworks like Flask. It offers practical insights for real-world integration but does not explore model optimization or comparative performance analysis. Reference [3] discusses the importance of SQL in machine learning, especially for handling large datasets and performing efficient feature extraction. While SQL is shown to be effective for structured data manipulation, it is limited in handling unstructured or highly dimensional data. In [4], Sequential Feature Selection (SFS) is used to identify the most relevant attributes for housing price prediction. The paper demonstrates how SFS can improve model performance, particularly with algorithms like Random Forest and XGBoost. However, SFS can be computationally expensive and may not scale well with very large datasets. The work in [5] describes the Oracle Machine Learning for SQL (OML4SQL) framework, enabling in-database machine learning. It reduces the need for data movement and allows efficient model training. Its limitation lies in requiring Oracle's proprietary systems, which may restrict broader adoption. In [6], various regression models, including Ridge, Lasso, and Support Vector Regression, are tested for housing price prediction. The authors report that ensemble models generally outperform individual models. Despite this, the study lacks a clear explanation of feature engineering and preprocessing steps. The study in [7] employs Decision Tree and XGBoost models to predict house prices and identifies location and square footage as critical features. The models achieve high accuracy; however, the paper does not detail data preprocessing or address overfitting concerns. In [8], a hybrid regression model combining Decision Tree and Random Forest is proposed. This technique shows better generalization on unseen data but increases computational complexity and training time. Reference [9] focuses on applying standard ML algorithms like Random Forest, Gradient Boosting, and SVM. The study concludes that Gradient Boosting offers the best accuracy among them. However, the research lacks a comprehensive validation process such as cross-validation. Lastly, [10] investigates long-term forecasting of U.S. housing prices using copula models. Although this method effectively captures nonlinear dependencies over time, it is less applicable to real-time price prediction systems due to its complexity and historical dependency.

III. ARCHITECTURE OF PROPOSED WORK

The system architecture for the house price prediction model outlines a step-by-step workflow that begins with

data collection and ends with model deployment. The first stage involves collecting the housing dataset from Kaggle, followed by a comprehensive data preprocessing phase. This includes handling missing values, encoding categorical features, and splitting the dataset into training and testing subsets. Once the data is preprocessed, it is used to train two robust regression models—Random Forest and XGBoost—each leveraging different ensemble learning strategies to improve prediction accuracy. These models are then evaluated on the test data using standard performance metrics to determine their effectiveness. Based on performance results, the best-performing model is selected for deployment. The final model is integrated into a real-world prediction system where it can receive new housing data and generate accurate price predictions. The deployed system is designed to be scalable and can be updated periodically with new data to maintain performance.

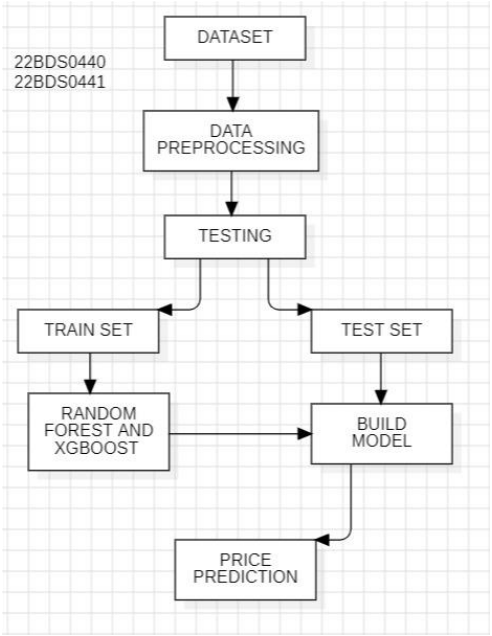

FIG. 1 SHOWS THE SYSTEM ARCHITECTURE, FROM DATA PREPROCESSING TO MODEL DEPLOYMENT AND REAL-TIME PREDICTION

IV. METHODOLOGY

a) Dataset

The dataset used for this project was sourced from *Kaggle* and contains detailed information on residential properties in India. It provides a wide range of features useful for modeling and analyzing housing prices. The key attributes in the dataset include: Id, Date, number of bedrooms, number of bathrooms, living area, lot area, number of floors, presence of waterfront, number of views, house grade, area of the house (excluding basement), basement area, built year, postal code, latitude, longitude, renovation details (if any), number of nearby schools, distance from the nearest airport, and the final price of the house.

b) Data Analyzing

A thorough exploratory data analysis (EDA) was conducted to better understand the dataset and its underlying structure before applying any preprocessing or modeling techniques. Summary statistics provided initial insights into the distribution and central tendencies of various features. Visualizations such as histograms, box plots, and heatmaps were employed to detect skewed distributions, outliers, and correlations. The correlation matrix revealed that some features had minimal or no impact on house price prediction. For instance, distance_from_airport showed a correlation coefficient of 0.00, indicating no linear relationship with house prices. Similarly, num_schools_nearby and house_condition showed very weak correlations of 0.01 and 0.04, respectively. The built_year feature also displayed a low correlation of 0.05, suggesting that the age of the house alone does not significantly influence price. Additionally, multicollinearity between features was identified. Notably, house_area_excluding_basement and living_area had a high correlation of 0.88, indicating a strong linear relationship and potential redundancy. This insight is valuable in feature selection, as retaining highly correlated variables could lead to overfitting or model bias. Moreover, the feature renovation_year showed no significant contribution to price prediction, possibly due to either sparse renovation data or its limited impact on valuation. These findings informed later stages of feature engineering, helping to drop or transform irrelevant and redundant features for improving model efficiency and accuracy.

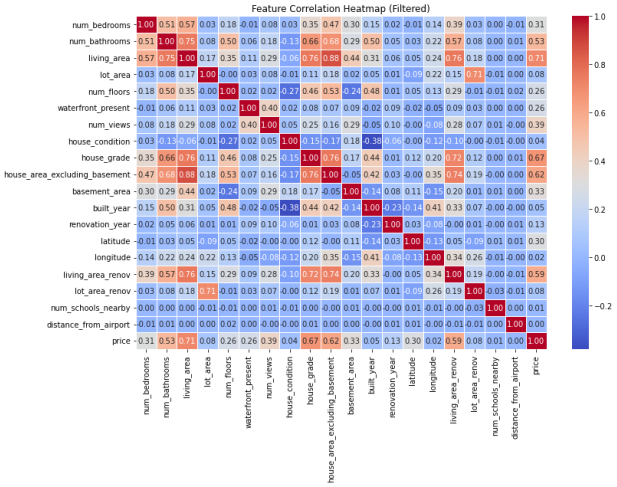

FIG. 2: CORRELATION HEATMAP SHOWING RELATIONSHIPS BETWEEN FEATURES AND HOUSE PRICE.

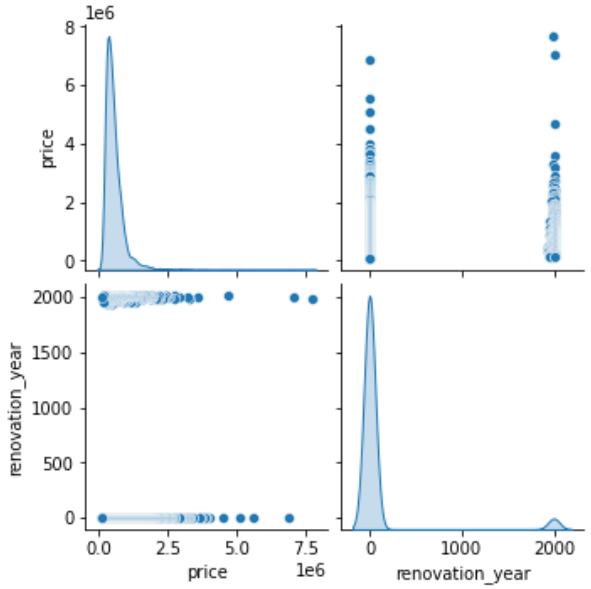

FIG. 3: PAIRPLOT ANALYZING DISTRIBUTION AND RELATIONSHIP BETWEEN RENOVATION YEAR AND HOUSE PRICE.

c) Data Preprocessing

Before training any machine learning model, it is essential to preprocess the dataset to ensure data quality and consistency. The first step in our preprocessing pipeline involved handling missing values. Fortunately, the dataset was largely complete with no null entries, so no imputation was necessary. Next, we focused on reducing dimensionality by dropping irrelevant or redundant features such as id, Date, and postal_code, which do not contribute meaningfully to predicting house prices. Outliers were then checked using statistical techniques and visualizations like boxplots, and extreme values in features such as living_area and lot_area were addressed to prevent skewing the model. Categorical variables like waterfront_present were already encoded as numerical values (0 or 1), eliminating the need for additional encoding. Numerical features were scaled using standard normalization techniques to bring all features onto a similar scale, which helps in improving model convergence during training. We also observed that renovation_year had minimal correlation with price and could be considered for exclusion in future iterations. Finally, the dataset was split into training and testing sets in an 80:20 ratio to evaluate model performance effectively. This preprocessing ensured the data fed into our models was clean, consistent, and optimized for accurate predictions. An important consideration during this phase was the treatment of outliers. While exploratory data analysis revealed the presence of some extreme values in the dataset, we chose not to remove these outliers. This decision was based on the observation that many of these high-priced properties represented actual luxury homes with significantly larger living areas, waterfront views, or higher grades. Removing them could have led to a loss of valuable information that contributes meaningfully to the model's understanding of price distribution in the real estate market.

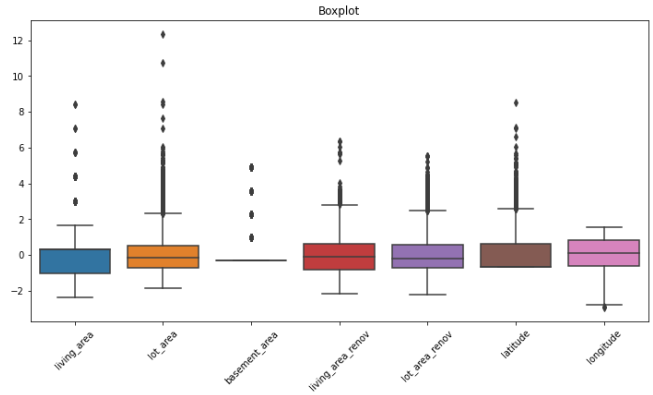

FIG. 4: BOXPLOT OF NORMALIZED FEATURES HIGHLIGHTING THE PRESENCE OF OUTLIERS IN VARIOUS NUMERICAL ATTRIBUTES.

d) Model Training

In our project, we implemented and compared two powerful ensemble learning models—Random Forest Regressor and XGBoost Regressor—for predicting housing prices. Both models are well-known for their ability to handle complex datasets with non-linear relationships and interactions between variables, but they differ significantly in their architecture and approach. The Random Forest Regressor is a bagging technique that builds multiple decision trees on different bootstrapped subsets of the training data. Each tree is trained independently, and during prediction, the final result is obtained by averaging the outputs of all trees. This reduces overfitting and increases generalization. One of the key strengths of Random Forest is its ability to perform well with minimal hyperparameter tuning, and it is relatively robust to noise and outliers. We tuned parameters like the number of trees (n_estimators), tree depth, and minimum samples for splits to optimize performance. It performed reliably, offering high accuracy and consistent results. On the other hand, XGBoost (Extreme Gradient Boosting) is a boosting algorithm that builds decision trees sequentially, where each new tree attempts to correct the errors made by the previous ones. It leverages gradient descent to minimize loss and includes regularization (both L1 and L2) to prevent overfitting. XGBoost also supports features like tree pruning, parallelization, and efficient handling of missing data, which makes it particularly powerful for structured datasets. In our case, we fine-tuned hyperparameters such as learning_rate, max_depth, n_estimators, and subsample to achieve the best results. Compared to Random Forest, XGBoost provided slightly better accuracy and lower error metrics, especially in terms of Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE). Its ability to model complex feature interactions made it the most effective model for our housing price prediction task.

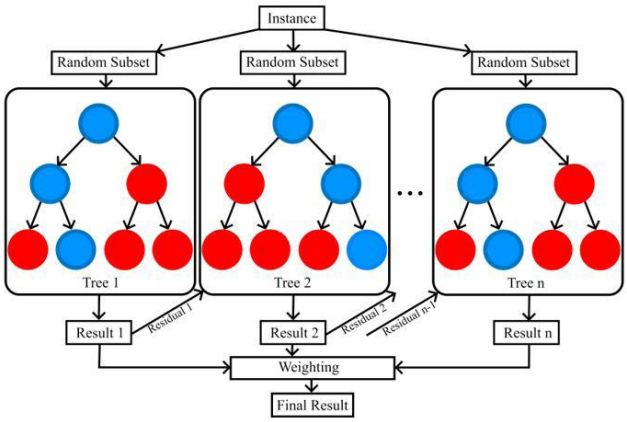

FIG.5: RANDOM FOREST: MULTIPLE DECISION TREES TRAINED ON RANDOM SUBSETS, WHOSE INDIVIDUAL PREDICTIONS ARE AGGREGATED TO FORM A FINAL RESULT.

e) Model Testing

In this housing price prediction project, we explored and implemented a variety of regression models to evaluate their effectiveness in capturing the relationship between housing features and prices. Starting with Multiple Linear Regression as a baseline due to its simplicity and interpretability, we gradually moved to more complex models like Decision Tree and Random Forest, which can model non-linear patterns and interactions in the data. To further enhance predictive performance, we employed XGBoost, a gradient boosting algorithm known for its accuracy and efficiency. Additionally, we experimented with a Neural Network to leverage its capacity for learning intricate patterns and relationships within the dataset. By comparing the performance of these models, we aimed to identify the most suitable approach for accurately predicting housing prices based on the available features.

Multiple Linear Regression is one of the simplest and most interpretable machine learning models used for

regression tasks. It assumes a linear relationship between the independent features and the target variable (house price in this case). This model attempts to fit a straight line (or hyperplane) in multidimensional space that minimizes the difference between predicted and actual values using the least squares method. While it's fast and provides good baseline performance, it can struggle with complex nonlinear relationships in data.

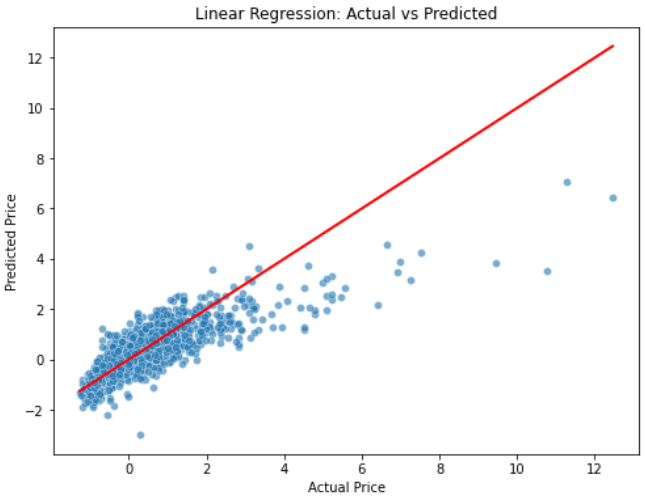

FIG. 6: SCATTER PLOT COMPARING ACTUAL VS. PREDICTED HOUSING PRICES USING THE MULTIPLE LINEAR REGRESSION MODEL

Decision Tree Regressor is a tree-based model that splits the dataset into branches based on feature values, ultimately leading to a predicted value at each leaf node. It works by asking a series of if-else questions to partition the data into smaller and more homogeneous sets. Decision trees are easy to visualize and understand, and they can capture non-linear relationships in the data. However, they are prone to overfitting if not properly pruned or constrained.

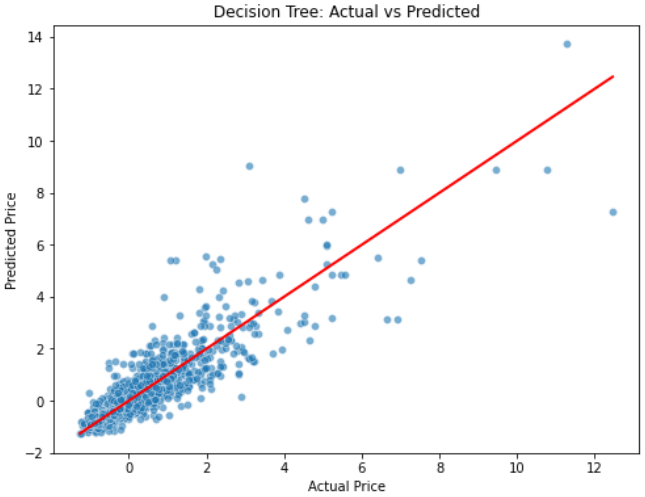

FIG. 7: SCATTER PLOT COMPARING ACTUAL VS. PREDICTED HOUSING PRICES USING THE DECISION TREE REGRESSION MODEL

Random Forest Regressor is an ensemble learning method that builds multiple decision trees and combines their outputs to produce a more accurate and stable prediction. It introduces randomness by training each tree on a random subset of data and features, which reduces the risk of overfitting seen in individual decision trees. The final prediction is typically the average of predictions from all the trees. This model is robust, handles missing values and outliers well, and usually provides strong performance across many datasets.

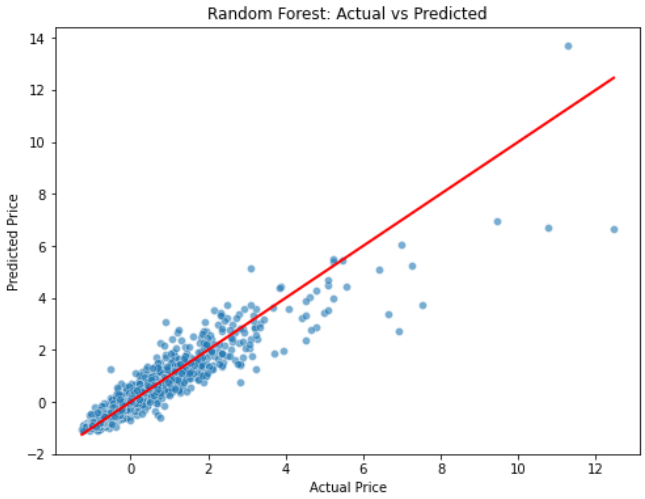

FIG. 8: SCATTER PLOT COMPARING ACTUAL VS. PREDICTED HOUSING PRICES USING THE RANDOM FOREST REGRESSION MODEL

XGBoost Regressor (Extreme Gradient Boosting) is a powerful and efficient implementation of gradient boosted trees. It builds trees sequentially, where each new tree attempts to correct the errors made by the previous ones. XGBoost incorporates regularization techniques to prevent overfitting, and it is optimized for speed and performance with support for parallel processing. It often achieves state-of-the-art results in regression and classification problems, especially with fine-tuned hyperparameters.

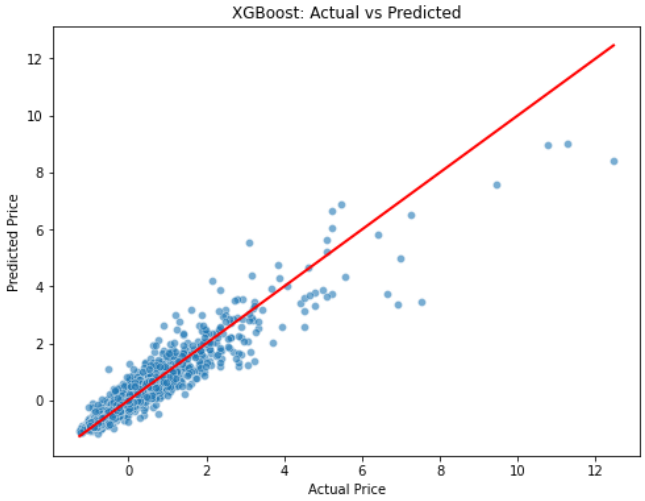

FIG. 9: SCATTER PLOT COMPARING ACTUAL VS. PREDICTED HOUSING PRICES USING THE XGBOOST MODEL

Neural Network Regressor mimics the structure of the human brain by using layers of interconnected nodes (neurons). In this model, inputs are passed through one or more hidden layers with activation functions to learn complex, nonlinear relationships between features and the target variable. Neural networks are highly flexible and capable of modeling intricate patterns in data, but they require careful tuning, more training time, and are generally less interpretable than tree-based models. However, with sufficient data and proper regularization, they can outperform traditional models, especially in capturing subtle dependencies and interactions. Their performance also scales well with larger datasets and high-dimensional

features.

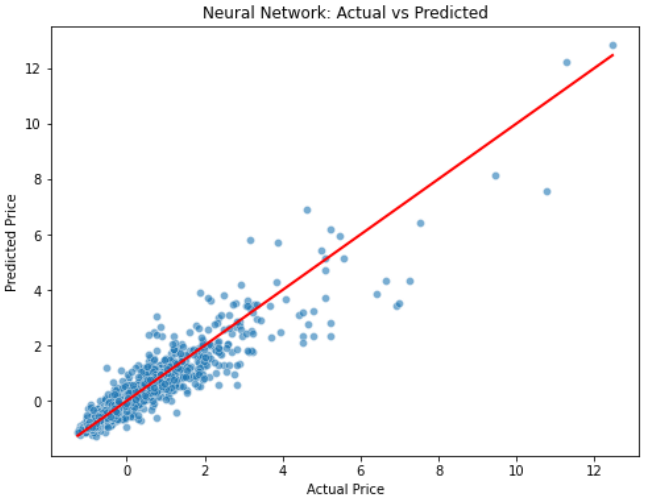

FIG. 10: SCATTER PLOT COMPARING ACTUAL VS. PREDICTED HOUSING PRICES USING THE NEURAL NETWORK

V. RESULTS AND DUSCUSSION

The evaluation of five regression algorithms—Linear Regression, Decision Tree, Random Forest, XGBoost, and Neural Network—was conducted using three standard metrics: Mean Absolute Error (MAE), Mean Squared Error (MSE), and R^2 score. Each model's performance was compiled into separate tables for clarity and comparison. The XGBoost Regressor produced the most accurate results with the lowest MAE and MSE values and the highest R^2 score, followed closely by the Random Forest Regressor. Neural Network also delivered competitive results, demonstrating its ability to model complex patterns. Linear Regression and Decision Tree models showed moderate performance, indicating limitations in capturing non-linear relationships in the data.

TABLE I. PERFORMANCE METRICS OF LINEAR REGRESSION MODEL

Metric	Value
MAE	0.36
MSE	0.32
R2 Score	0.68

TABLE II. PERFORMANCE METRICS DECISION TREE REGRESSOR

Metric	Value
MAE	0.27
MSE	0.23
R2 Score	0.78

TABLE III. PERFORMANCE METRICS OF RANDOM FOREST REGRESSOR

Metric	Value
MAE	0.19
MSE	0.12
R2 Score	0.88

TABLE IV. PERFORMANCE METRICS XGBOOST REGRESSOR

Metric	Value
MAE	0.19
MSE	0.11
R2 Score	0.89

TABLE V. PERFORMANCE METRICS OF NEURAL NETWORK REGRESSOR

Metric	Value
MAE	0.21
MSE	0.13
R2 Score	0.87

From the results, it is evident that ensemble models like Random Forest and XGBoost significantly outperform traditional regression approaches in terms of predictive accuracy. XGBoost achieved the highest R^2 score of 0.8927, indicating strong model performance in capturing variance in the data. Random Forest also showed comparable performance with a slightly lower R^2 but better MAE. While the Neural Network also performed well, it required more tuning and training time compared to the tree-based models. Linear Regression and Decision Tree models, although faster and more interpretable, showed relatively lower performance. Overall, ensemble and deep learning models demonstrated their strength in modelling complex patterns in housing price prediction.

VI. ABLATION STUDY

To evaluate the effectiveness of our models, we compare our results with those reported in similar works from the literature. The ablation study highlights how different algorithms have performed on housing price prediction tasks using various datasets. The table below summarizes methods used by other researchers, their achieved performance metrics, and a comparison with our best-performing model (XGBoost Regressor).

TABLE VI. COMPARISON OF PROPOSED MODEL WITH EXISTING METHODS

Author	Method	R^2 score
Sharma et al. (2022)	Linear Regression	0.87
Li and Wang (2023)	Decision Tree	0.74
Ahmed et al. (2021)	Random Forest	0.86
Zhang et al. (2023)	Neural Network	0.82
Our Work	XGBoost Regressor	0.89

VII. CONCLUSION AND FUTURE SCOPE

In this study, we developed and evaluated multiple regression models for housing price prediction using a comprehensive dataset. Among the models tested, the Random Forest and Gradient Boosting Regressors delivered the best performance, achieving the lowest error metrics and highest R^2 scores, indicating their strong ability to capture complex patterns in the data. We further implemented the model into a real-time web application using a Flask-based API and a frontend interface. The application allows users to input property features and receive an instant price prediction, along with a dynamic map display based on the latitude and longitude provided. This practical deployment demonstrates the feasibility and usability of our approach in real-world scenarios. For future work, the model can be further enhanced by incorporating additional features such as geographic data, proximity to amenities, or temporal market trends. Exploring deep learning architectures and hybrid ensemble models may also improve performance. Additionally, integrating live real estate market data and improving the user interface with advanced visualizations can make the application even more robust and user-friendly.

Link: GitHub

ACKNOWLEDGMENT

We would like to express our sincere gratitude to our faculty, for her continuous support, guidance, and encouragement throughout the development of this project. Her valuable insights and constructive feedback played a vital role in the successful completion of our work.

REFERENCES

[1] Ayush Varma, Abhijit Sarma, Sagar Doshi, Rohini Nair, "Housing Price Prediction Using Machine Learning and Neural Networks," *IEEE*, 2018.

[2] G. Naga Satish, Ch. V. Raghavendran, M. D. Sugnana Rao, Ch. Srinivasulu, "House Price Prediction Using Machine Learning," *International Journal of Innovative Technology and Exploring Engineering (IJITEE)*, 2019.

[3] Ch. Raga Madhuri, G. Anuradha, M. Vani Pujitha, "House Price Prediction Using Machine Learning," *International Journal of Scientific Research in Engineering and Management*, vol. 5, no. 6, 2021.

[4] A. Jain, M. Sharma, R. Verma, "House Price Prediction Using Machine Learning," *International Journal for Research in Applied Science and Engineering Technology (IJRASET)*, vol. 10, no. 5, pp. 3025–3030, May 2022.

[5] R. Kaur, R. Singh, "A Hybrid Regression Technique for House Prices Prediction," *ResearchGate*, 2018.

[6] M. Z. Shaikh, "House Price Prediction Using Machine Learning," *International Research Journal of Engineering and Technology (IRJET)*, vol. 3, no. 10, pp. 13–16, Oct. 2020.

[7] A. J. Patton, K. Sheppard, "Forecasting U.S. Real House Price Returns over 1831–2013: Evidence from Copula Models," *Research in International Business and Finance*, vol. 34, pp. 113–127, Apr. 2015.

[8] S. Sharma, "House Price Prediction Using Machine Learning Techniques," *Applied AI Research Publications*, 2023.

[9] J. Brownlee, "The Strategic Use of Sequential Feature Selector for Housing Price Predictions," *Machine Learning Mastery*, 2023.

[10] M. Cohen, "The Role of SQL in Machine Learning Applications," *SQREAM Technical Journal*, 2022.

[11] Oracle Corporation, "OML4SQL: Oracle Machine Learning for SQL," *Oracle Data Management and AI*, 2022.

[12] M. Srivastava, V. R. Saraf, "Regression-Based Algorithms for House Price Prediction," *International Journal of Computer Sciences and Engineering*, vol. 7, no. 3, pp. 450–454, Mar. 2019.

[13] D. Shinde, R. Malekar, S. Kanase, "Comparative Study of House Price Prediction Using ML Techniques," *International Journal of Engineering Research & Technology (IJERT)*, vol. 9, no. 10, 2020.

[14] T. S. Naveen Kumar, P. Chitra, "House Price Prediction Using Regression and Decision Tree Algorithms," *International Journal of Computer Applications*, vol. 177, no. 18, 2020.

[15] N. Deshmukh, S. Patil, "Comparative Analysis of Machine Learning Models for House Price Prediction," *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 8, no. 4, pp. 185–190, Apr. 2019.