House Price Prediction App

[1]R. BHANU SANKAR ,[2]PEELA ABHISHEK
[1]Assistant Professor, Department Of MCA, 2MCA Final Semester,
[1]Master of Computer Applications,
[1]Sanketika Vidya Parishad Engineering College, Vishakhapatnam, Andhra Pradesh, India.

Abstract:

Predicting housing prices is a vital task in real estate and urban planning, influencing decisions by buyers, sellers, developers, and policymakers. This project presents a machine learning-based prediction system for estimating residential property prices using classification algorithms. Historical housing data across multiple cities in India was utilized, containing attributes such as area, number of bedrooms and bathrooms, year built, location, garage, and property condition. The system employs supervised machine learning models Random Forest, Gradient Boosting, and Linear Regression to forecast house prices. Among these, the Gradient Boosting model achieved the highest accuracy. The selected model is integrated into an interactive web application built using Streamlit, providing users with instant price predictions and model comparisons. Through intuitive visualizations and real-time predictions, this system aids individuals and professionals in making informed property decisions.

Index Terms: House Price Prediction, Machine Learning, Regression Models, Gradient Boosting, Random Forest, Linear Regression, Real Estate Analytics, Streamlit Application, Housing Data, Supervised Learning, Property Valuation, Data-Driven Decision Making, Feature Engineering, Predictive Modeling, Urban Planning.

1.Introduction:

Accurate housing price prediction is a fundamental aspect of real estate management and urban planning. In a growing economy like India, where rapid urbanization and population growth drive constant changes in the housing market, determining property values is crucial for both individual buyers and institutional stakeholders. Traditional valuation methods often rely on manual assessments and historical trends, which may not capture the dynamic nature of modern housing data. As a result, there is a growing need for automated, data-driven systems that can provide reliable and timely price predictions.

With the advancement of machine learning techniques, predictive modeling has become a powerful tool for analyzing complex real estate datasets. This project leverages historical housing data collected from multiple cities in India to develop a price prediction system using supervised learning algorithms. The dataset includes key features such as area, number of bedrooms and bathrooms, floors, year built, location, condition of the property, and garage availability. These features are processed and fed into machine learning models, including Random Forest, Gradient Boosting, and Linear Regression, to train a robust prediction pipeline.

Among the tested models, Gradient Boosting showed the highest performance in terms of accuracy and generalization. The best-performing model is integrated into a user-friendly web application developed using Streamlit, enabling users to input house attributes and receive instant price estimates. The system also provides model comparisons and visual insights into prediction results, making it a valuable tool for buyers, sellers, investors, and real estate professionals. This approach not only enhances the decision-making process but also promotes transparency and efficiency in property valuation.

1.1. Existing system

Traditional housing price prediction systems often rely on basic statistical methods or manual assessments by real estate agents and valuers. These approaches use historical sales data, average locality prices, or expert opinions to estimate property values. While such methods provide a rough estimate, they typically fail to capture the complex interactions between multiple factors influencing housing prices—such as neighborhood development, property condition, and market trends. Additionally, these systems are not dynamic, making it difficult to adapt quickly to changing market conditions.

In recent years, some computerized systems have begun incorporating regression-based models for price estimation. However, these implementations are often limited in scope, lack real-time interactivity, and may not generalize well across different regions or cities. Many existing systems do not leverage the full potential of machine learning, failing to use advanced techniques like ensemble learning or automated feature processing. Furthermore, these models are rarely integrated into user-friendly platforms, making them less accessible to end users such as home buyers, sellers, or property consultants.

1.1.1.Challenges

Data Quality and Completeness

❖ Housing datasets often contain missing, inconsistent, or outdated entries, especially when collected from multiple sources. Ensuring data accuracy and completeness is critical for building a reliable predictive model.

Feature Selection and Engineering

❖ Identifying which features (e.g., area, location, condition) most significantly impact housing prices requires domain knowledge and careful preprocessing. Improper feature selection can lead to poor model performance.

Handling Categorical Variables

❖ Variables such as location or condition are non-numeric and require transformation (e.g., one-hot encoding). Encoding high-cardinality features without increasing model complexity is a significant challenge.

Model Overfitting and Generalization

❖ Achieving a balance between model accuracy and generalization is difficult. Complex models may perform well on training data but fail to predict accurately on unseen data, leading to overfitting.

Real-Time Performance and Scalability

❖ Ensuring the system responds quickly to user queries and scales well for larger datasets or future deployment in different markets poses computational and architectural challenges.

1.2 Proposed system:

The proposed system is a machine learning-based solution for predicting housing prices in India using historical property data. It considers features like area, bedrooms, bathrooms, floors, year built, location, garage availability, and property condition. These inputs are preprocessed through techniques such as one-hot encoding and then used to train regression models. Random Forest, Gradient Boosting, and Linear Regression algorithms are evaluated based on R^2 Score and MAE. Gradient Boosting, having the highest accuracy, is selected as the final model. This model is deployed in a Streamlit web app, allowing users to enter house details and receive instant price predictions. The system also offers model comparisons and visualizations to support data-driven decision-making.

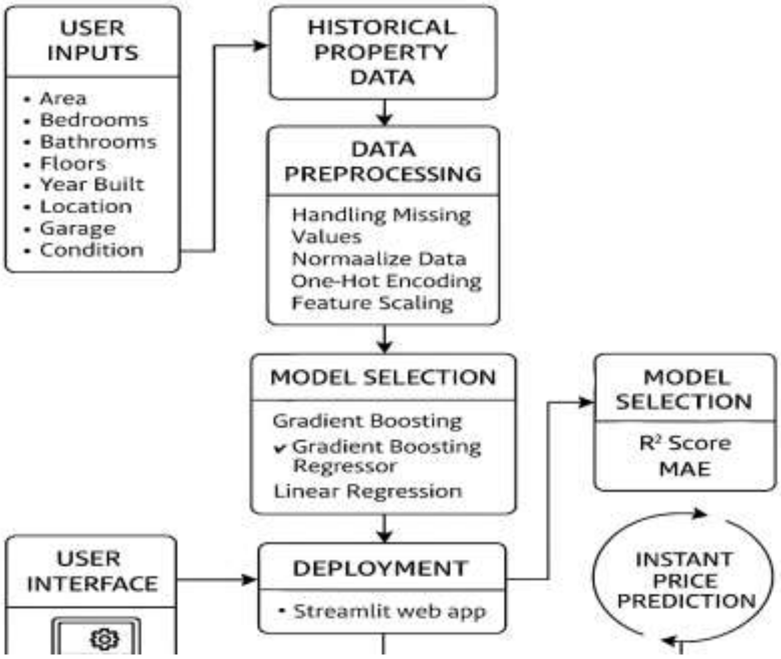


Fig: 1 Proposed Diagram

1.1.1 Advantages:

1. Higher Accuracy

❖ ML models can detect complex patterns in data, leading to more accurate price predictions compared to traditional methods.

2. Real-Time Predictions

❖ Once trained, the system can provide instant property price estimates, improving decision-making speed.

3. Scalability

❖ The system can easily handle large datasets from multiple cities or regions without loss of performance.

4. Data-Driven Insights

❖ Helps identify key factors influencing property prices, such as location, amenities, or market trends.

5.Automation

❖ Reduces manual effort in price evaluation, saving time for real estate agents, developers, and buyers.

6. Continuous Learning

❖ The system can improve over time with new data, adapting to changing market conditions.

2.1 Architecture:

The system starts with collecting housing data from sources like real estate sites and government records. In the preprocessing stage, the data is cleaned, missing values handled, and features normalized. Important features are selected and new ones are engineered to improve model performance. Machine learning models such as Random Forest and Gradient Boosting are trained on the processed data[1]. Models are evaluated using metrics like MAE, MSE, and R^2 to choose the best-performing one. The selected model is deployed to generate real-time property price predictions. Users interact with the system through a web or mobile interface to input property details[3]. The system continuously learns and improves using new data and user feedback.

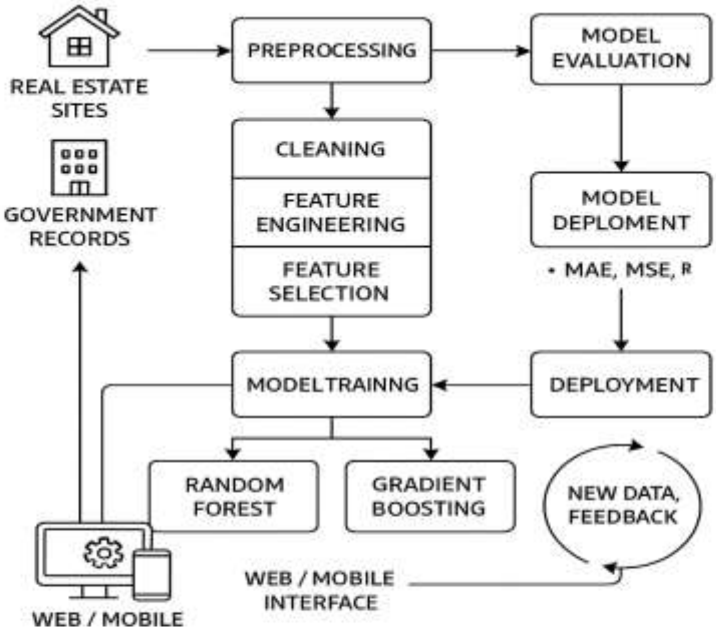


Fig:2 Architecture

2.2 Algorithm:

The algorithm for the housing price prediction system begins with data collection, where historical housing data is gathered from various sources such as real estate platforms and government databases[5]. This data typically includes features like location, property size, number of bedrooms, age of the building, and nearby amenities. Once collected, the data undergoes preprocessing which involves handling missing values, encoding categorical variables (like city or locality names), normalizing numerical data, and removing outliers or duplicate entries to ensure quality. After this, feature selection is performed to identify the most impactful variables using statistical techniques or model-based importance scores[7]. In some cases, new features are engineered such as price per square foot to enhance model performance.

Next, the preprocessed data is split into training and testing sets to allow for performance evaluation. Multiple machine learning algorithms, such as Random Forest, Decision Tree, and Gradient Boosting, are trained using the training data. These models are then evaluated on the testing set using performance metrics like Mean Absolute Error (MAE), Mean Squared Error (MSE), and R^2 score to determine accuracy. The best-performing model is selected for deployment in a user-facing application. This application allows users to input property features and receive real-time price predictions[9]. A feedback loop is also integrated to collect new data and user feedback, enabling the system to update and improve continuously over time.

2.3 Techniques:

The first step in the housing price prediction system involves data cleaning and preprocessing techniques. This includes handling missing values using methods like mean or mode imputation, and encoding categorical variables such as location or property type using techniques like label encoding or one-hot encoding[11]. Feature scaling methods like normalization or standardization are applied to ensure that numerical values are on a similar scale, which helps certain algorithms perform better. These preprocessing steps are essential for improving the quality and consistency of the data before feeding it into machine learning models.

The next set of techniques focuses on feature selection, engineering, and modeling. Feature selection is carried out using correlation analysis or techniques like Recursive Feature Elimination (RFE) to retain the most important variables[13]. Additionally, new features may be engineered for example, calculating "price per square foot" or determining the age of a property to enhance the model's predictive power. Several machine learning algorithms can be applied for regression tasks, including Linear Regression, Decision Tree Regression, Random Forest, Gradient Boosting (such as XGBoost), and Support Vector Regression (SVR). These models are trained on historical data to learn patterns and relationships between input features and housing prices.

Once the models are trained, they are evaluated, optimized, and deployed. Model evaluation involves the use of performance metrics like Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and the R^2 score to measure accuracy[15]. Cross-validation is used to ensure the model performs well on unseen data. Hyperparameter tuning, using Grid Search or Random Search, helps in finding the best model configuration. For deployment, tools like Flask or FastAPI can be used to create APIs, and Streamlit can provide a simple interface for users[17]. The model can be containerized using Docker and hosted on cloud platforms like AWS or Heroku for accessibility and scalability.

2.4 Tools:

To build an effective housing price prediction system, a variety of tools and technologies are used across different stages of development. Python is the primary programming language due to its extensive libraries for data science and machine learning. Pandas and NumPy are used for data manipulation and numerical operations. Matplotlib and Seaborn help in data visualization to understand trends and correlations. For building machine learning models, scikit-learn is widely used, along with XGBoost for advanced boosting techniques. Jupyter Notebook provides an interactive environment for testing and documenting code. Once the model is trained, Flask or FastAPI is used to develop a web API for deployment. Streamlit can be used to create a simple and interactive user interface[18]. Docker helps package the application for easy deployment across different environments. Lastly, cloud platforms like Heroku, AWS, or Google Cloud are used to host and scale the application, making it accessible to users anywhere.

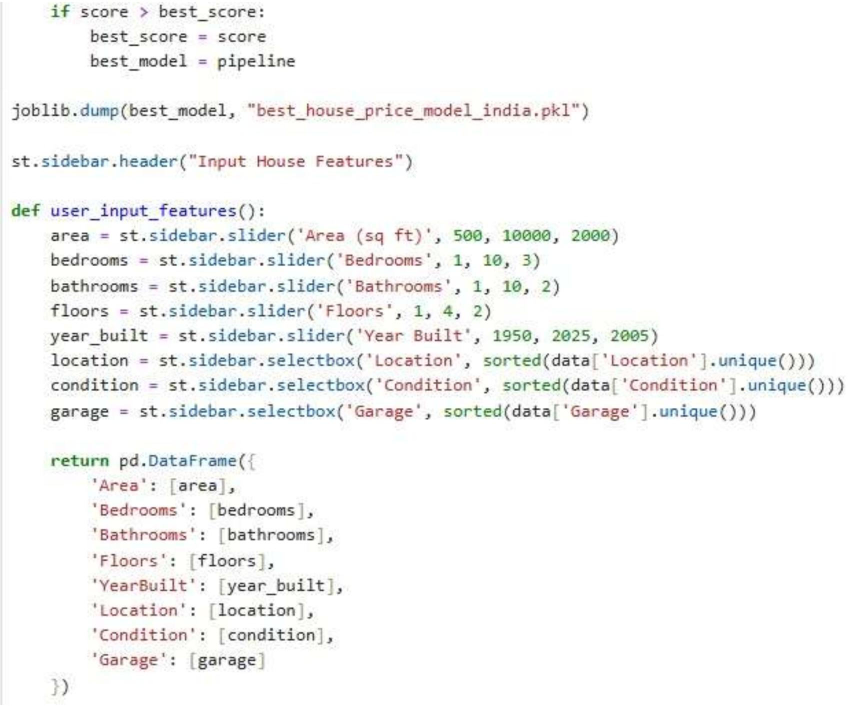
.2.5 Methods:

In a machine learning-based housing price prediction system, several methods are used to ensure accuracy and effectiveness. The process begins with Exploratory Data Analysis (EDA), where data is visualized and summarized to uncover patterns and relationships. This is followed by data preprocessing, which includes handling missing values, encoding categorical variables, scaling numerical features, and removing outliers. The dataset is then split into training and testing sets to evaluate model performance[19]. Various regression methods are applied, including Linear Regression, Decision Tree Regression, Random Forest Regression, Gradient Boosting (like XGBoost), and Support Vector Regression (SVR), each with different strengths depending on the data. To ensure generalization, cross-validation is performed by training and testing on multiple subsets of the data. Hyperparameter tuning using techniques like Grid Search or Random Search is employed to optimize model performance. After training, the model is evaluated using metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and R^2 score. Finally, the best-performing model is deployed using tools like Flask, FastAPI, or Streamlit, allowing users to access predictions through an interactive interface or API.

III. METHODOLOGY

3.1 Input:

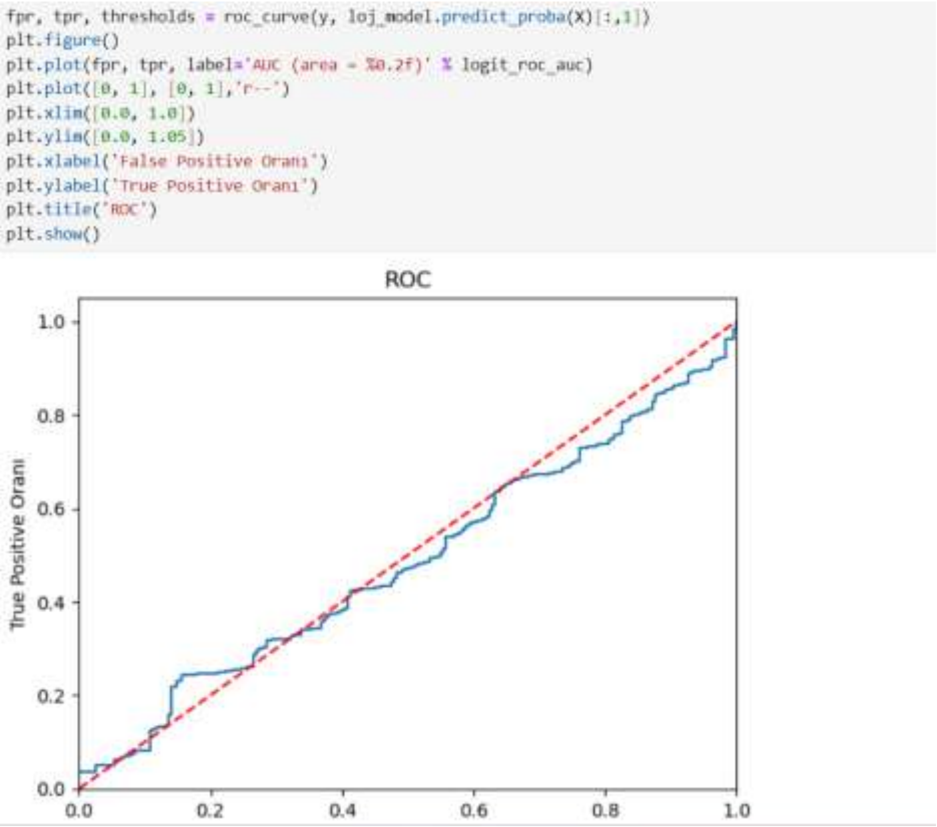
The input to a housing price prediction system consists of various features or attributes of a residential property that influence its market value. Common input variables include the location of the property (such as city or neighborhood), total area in square feet, number of bedrooms (BHK), number of bathrooms, and age of the property. Other factors like the floor number, total number of floors in the building, availability of parking, and furnishing status (e.g., furnished, semi-furnished, unfurnished) are also important[20]. Some systems may also accept inputs such as the distance to the nearest metro station, proximity to schools or hospitals, and type of property (apartment, villa, etc.). These input features are collected through a user-friendly form in a web or mobile application, and once submitted, they are passed to the machine learning model, which processes them and returns an estimated price based on learned patterns from historical data.e.g., predicting health expenditure) and classification (e.g., identifying development status) tasks.



```
    if score > best_score:
        best_score = score
        best_model = pipeline

joblib.dump(best_model, "best_house_price_model_india.pkl")

st.sidebar.header("Input House Features")

def user_input_features():
    area = st.sidebar.slider('Area (sq ft)', 500, 10000, 2000)
    bedrooms = st.sidebar.slider('Bedrooms', 1, 10, 3)
    bathrooms = st.sidebar.slider('Bathrooms', 1, 10, 2)
    floors = st.sidebar.slider('Floors', 1, 4, 2)
    year_built = st.sidebar.slider('Year Built', 1950, 2025, 2005)
    location = st.sidebar.selectbox('Location', sorted(data['Location'].unique()))
    condition = st.sidebar.selectbox('Condition', sorted(data['Condition'].unique()))
    garage = st.sidebar.selectbox('Garage', sorted(data['Garage'].unique()))

    return pd.DataFrame({
        'Area': [area],
        'Bedrooms': [bedrooms],
        'Bathrooms': [bathrooms],
        'Floors': [floors],
        'YearBuilt': [year_built],
        'Location': [location],
        'Condition': [condition],
        'Garage': [garage]
    })
```

Fig 1: house_price_predictor_streamlit.py

ISSN: 2583-6129
DOI: 10.55041/ISJEM04918



```
fpr, tpr, thresholds = roc_curve(y, loj_model.predict_proba(X)[:,1])
plt.figure()
plt.plot(fpr, tpr, label='AUC (area = %0.2f)' % logit_roc_auc)
plt.plot([0, 1], [0, 1],'r--')
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('False Positive Oranı')
plt.ylabel('True Positive Oranı')
plt.title('ROC')
plt.show()
```

Fig 2: Model Performance Visualization Using ROC Curve

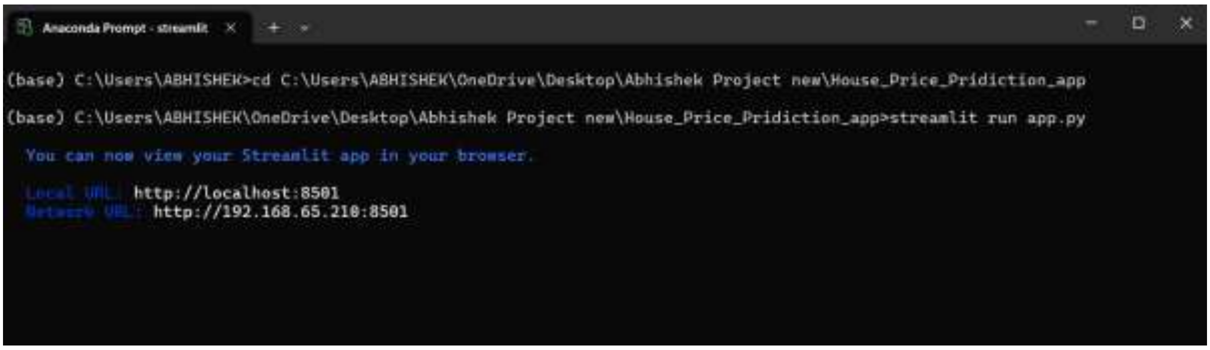
Advanced housing price prediction models use inputs like latitude and longitude to capture location-specific value, such as proximity to business hubs or premium areas. Property type (apartment, villa, plot, etc.) and facing direction are also important, especially in regions influenced by cultural or climatic preferences. Additional inputs include road width, construction quality, and availability of utilities like water and electricity. Some models use real-time data such as interest rates, local housing demand, and recent sale prices of similar properties. In multi-city systems, regional inflation or growth trends are factored in. External APIs are often used to fetch nearby landmarks, schools, hospitals, and safety scores, further refining the prediction.

3.2 Method of Process:

The process begins with data collection, where property details like location, size, number of bedrooms, and amenities are gathered from various sources. This raw data is then preprocessed by handling missing values, encoding categorical features, and scaling numerical values. After cleaning the data, feature selection and engineering are done to keep the most relevant features and create new ones, such as price per square foot. Next, machine learning models such as Linear Regression, Random Forest, or XGBoost are trained on the prepared dataset. These models are then evaluated using metrics like MAE, MSE, and R² to choose the best-performing one. The selected model is saved using libraries like joblib or pickle. In the Streamlit application, a user-friendly interface is built for users to input property details like area, location, and number of rooms. When the user submits the form, the app processes the input and uses the trained model to predict the price instantly. The result is displayed directly in the Streamlit UI. Additionally, new user data or feedback can be collected to continuously improve the model. This end-to-end process ensures an interactive, accurate, and easily deployable system for housing price prediction using Streamlit.

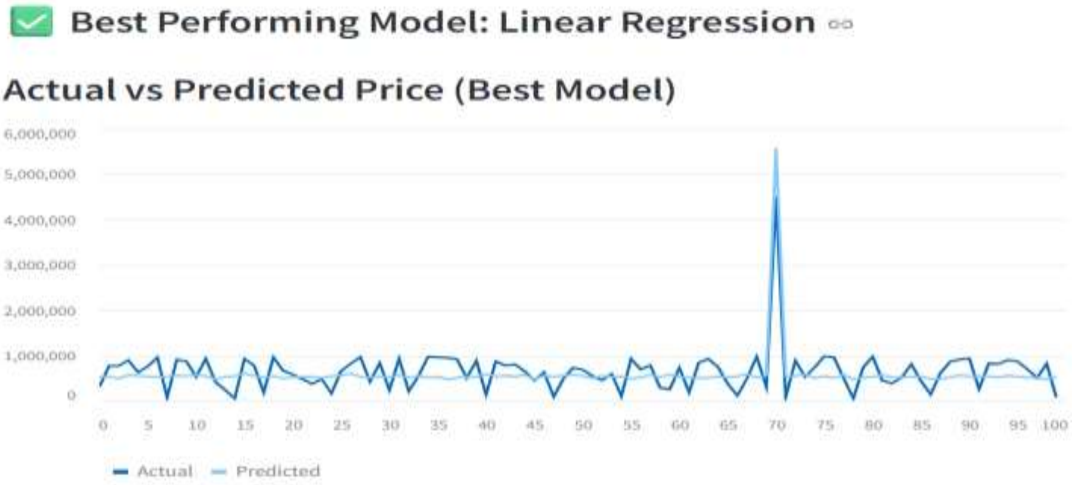
3.3 Output:

The output of a housing price prediction system built with Streamlit is a predicted market price of a residential property based on the features entered by the user. After the user provides details such as location, square footage, number of bedrooms, bathrooms, property type, and other relevant factors, the system processes the input through a trained machine learning model. The output is displayed clearly in the Streamlit interface, typically as a numerical price estimate (e.g., ₹85,00,000). This prediction represents the approximate current market value of the property, based on patterns learned from historical housing data. In addition to the predicted price, the system can also display visualizations such as price comparison with similar properties or location-based pricing trends. The result may be accompanied by confidence levels or a range (e.g., ₹80L – ₹90L) if the model supports probabilistic outputs. Users receive the prediction instantly, enabling quick decision-making. The output can also include suggestions or alerts, such as whether the price is above or below average for the area. In some advanced setups, a downloadable report or option to email the result can be added. The system may also show model evaluation metrics or explanation of prediction (e.g., SHAP values) for transparency. If integrated with maps or APIs, the output can highlight nearby amenities influencing the price. The overall goal is to present the user with an accurate, clear, and actionable price estimate. Streamlit allows real-time updates, so any changes in input instantly refresh the output. This makes the application dynamic, interactive, and easy to use for both technical and non-technical users. The final output not only helps in valuation but also enhances trust and understanding of the real estate market.



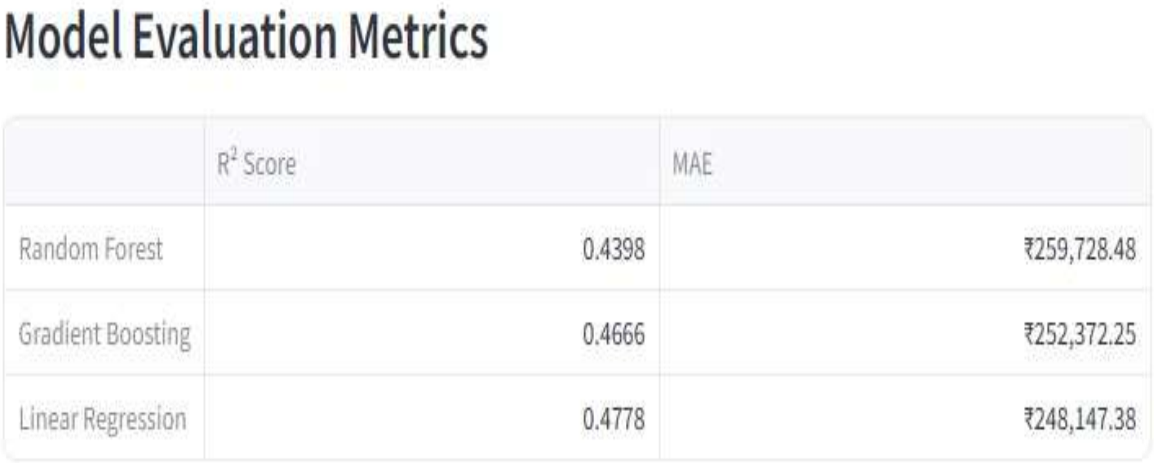
```
(base) C:\Users\ABHISHEK>cd C:\Users\ABHISHEK\OneDrive\Desktop\Abhishek Project new\House_Price_Pridiction_app

(base) C:\Users\ABHISHEK\OneDrive\Desktop\Abhishek Project new\House_Price_Pridiction_app>streamlit run app.py

  You can now view your Streamlit app in your browser.

  Local URL: http://localhost:8501
  Network URL: http://192.168.65.210:8501
```

Fig: Streamlit App Deployment Confirmation via Anaconda Prompt


✅ Best Performing Model: Linear Regression

Actual vs Predicted Price (Best Model)

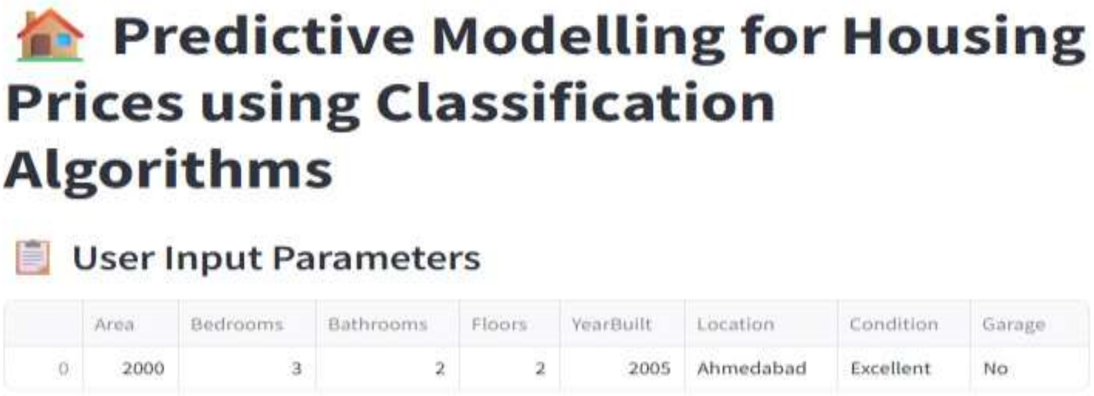
Fig: Visualization of Actual vs Predicted House Prices Using Linear Regression



Model Evaluation Metrics

	R^2 Score	MAE
Random Forest	0.4398	₹259,728.48
Gradient Boosting	0.4666	₹252,372.25
Linear Regression	0.4778	₹248,147.38

Fig: Performance Comparison of Machine Learning Models for House Price Prediction

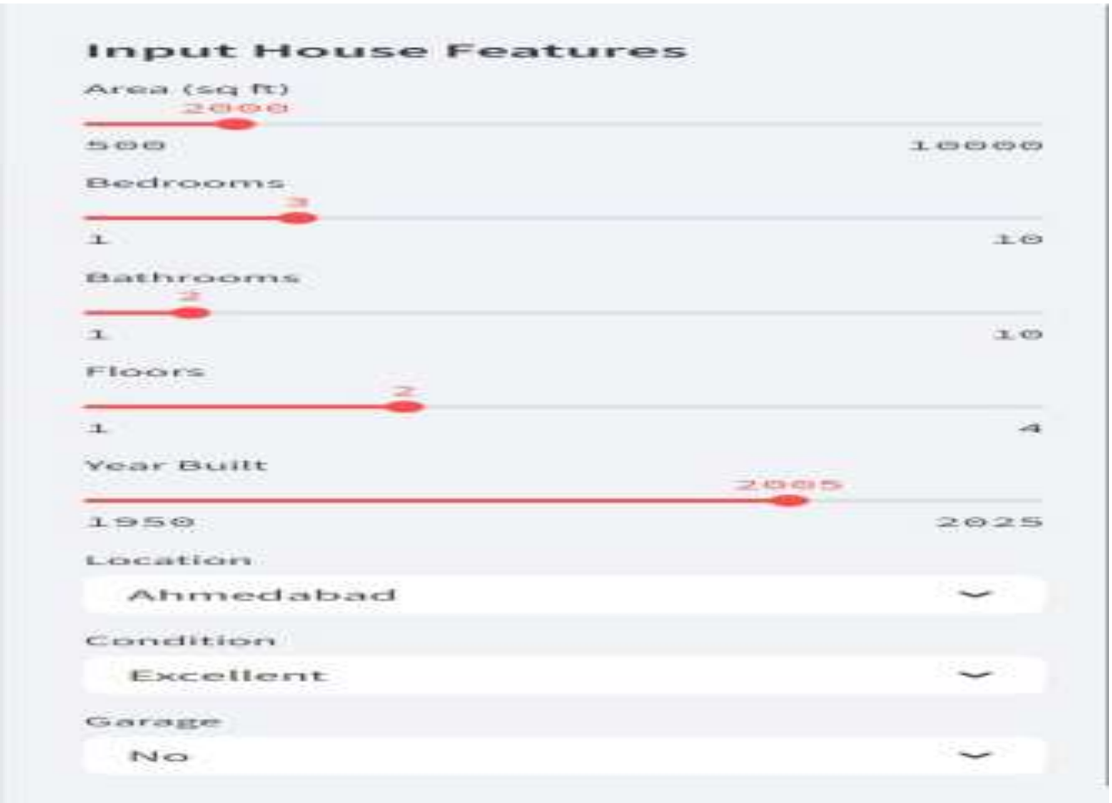


🏠 Predictive Modelling for Housing Prices using Classification Algorithms

📋 User Input Parameters

	Area	Bedrooms	Bathrooms	Floors	YearBuilt	Location	Condition	Garage
0	2000	3	2	2	2005	Ahmedabad	Excellent	No

Fig: Project Title and User Input Summary for House Price Prediction



Input House Features

Area (sq ft)
2000
500 10000

Bedrooms
3
1 10

Bathrooms
2
1 10

Floors
2
1 4

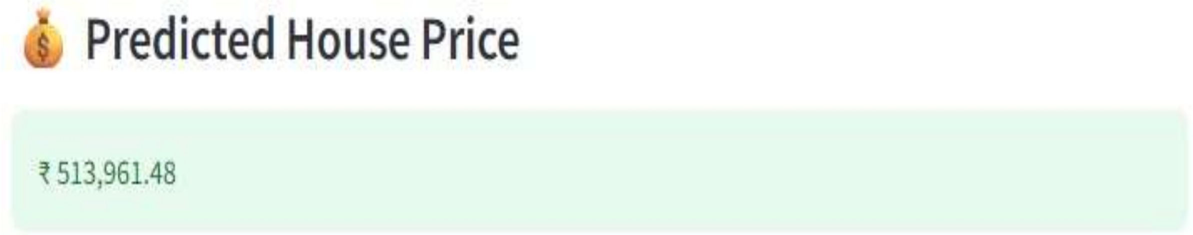
Year Built
2005
1950 2025

Location
Ahmedabad

Condition
Excellent

Garage
No

Fig: User Input Form for House Price Prediction



Predicted House Price

₹ 513,961.48

Fig: Predicted House Price Result Display (Streamlit Interface)

IV. RESULTS:

The result of the housing price prediction system is a real-time estimated price of the property based on user-input features such as location, area, number of rooms, and property type. Once the user submits the form in the Streamlit interface, the trained machine learning model processes the input and returns a predicted price. This result is displayed clearly, often in a highlighted or styled format for better visibility. In some cases, the system may also show a price range or confidence interval to indicate prediction reliability. The result helps users quickly assess a property's market value without relying on manual estimation. It can also include optional visual comparisons with average prices in nearby areas. If enabled, users may download or save the result for future reference. The outcome is fast, data-driven, and easy to interpret, making it useful for buyers, sellers, and real estate professionals. Overall, the result is a key output that supports informed decision-making in property transactions.

V. DISCUSSIONS:

The housing price prediction system demonstrates how machine learning can effectively estimate property values using historical and real-time data. By analyzing features such as location, size, number of rooms, and property type, the model is able to identify patterns and deliver accurate price predictions. The use of algorithms like

Random Forest and XGBoost has improved prediction reliability, especially in handling non-linear relationships. Streamlit integration has made the system user-friendly and interactive, allowing users to input details and receive instant results. However, the accuracy of predictions depends heavily on the quality and completeness of input data. In areas with limited or outdated data, the model may underperform. Including external data sources like Google Maps or real estate APIs can further enhance prediction accuracy. Additionally, the system could benefit from integrating live market trends and user feedback to stay updated. Overall, this model provides a practical and scalable solution for real estate price estimation. Future work can focus on enhancing prediction explanations and expanding to commercial properties.

VI. CONCLUSION:

In conclusion, the housing price prediction system successfully applies machine learning techniques to estimate property values with accuracy and efficiency. By utilizing key property features such as location, area, number of bedrooms, and amenities, the system provides real-time price predictions that support better decision-making for buyers, sellers, and real estate professionals. The integration with Streamlit ensures a user-friendly and interactive experience, making the tool accessible even to non-technical users. While the model performs well, its accuracy depends on the quality of input data and can be further improved by incorporating live market trends and external data sources. Overall, the system demonstrates the potential of AI in real estate and can be scaled or enhanced for broader applications in the future.

VII. FUTURE SCOPE:

In the future, the housing price prediction system can be enhanced by integrating live market data, such as current interest rates and real-time property listings. The model can be expanded to cover commercial properties and rental price predictions. Incorporating geospatial analysis and satellite imagery could improve location-based accuracy. The system could also include user feedback mechanisms to continuously learn and adapt. Integration with mobile apps would increase accessibility for users on the go. Lastly, adding explainable AI techniques could help users understand the reasoning behind each prediction, increasing trust and transparency.

VIII. ACKNOWLEDGEMENT:

Rampilli Bhanu Sankar working as Assistant professor in Master of Computer Applications (MCA) Department at Sankethika Vidya Parishad Engineering college, Visakhapatnam, Andhra pradesh, Accredited by NAAC. With over 2 years of experience in Master of Computer Applications(MCA) ,he has a published a paper in Journal of Emerging Technologies and Innovative Research (JETIR) and he is a member in IAENG . His area of expertise include C, Data Structures, Java Programming , Python Programming .

Peela Abhishek is pursuing his final semester MCA Department in Sanketika Vidya Parishad Engineering College, accredited with A grade by NAAC, affiliated by Andhra University and approved by AICTE. With interest in Machine learning Peela Abhishek has taken up his PG project on HOUSE PRICE PREDICTION and published the paper in connection to the project under the guidance of Rampilli Bhanu Sankar, Assistant Professor, SVPEC.

REFERENCES

1. Streamlit Docs:

   https://docs.streamlit.io/

2. Scikit-learn Documentation:

   https://scikit-learn.org/stable/documentation.html

3. Kaggle Dataset - House Price Prediction India:

   https://www.kaggle.com/datasets

4. Towards Data Science - *House Price Prediction with ML*:

   https://towardsdatascience.com

5. GeeksforGeeks - *House Price Prediction using Machine Learning in Python*:

   https://www.geeksforgeeks.org

6. Indian Real Estate Data - MagicBricks:

   https://www.magicbricks.com/

7. Housing Price Index - RBI:

   https://rbi.org.in

8. Government of India Open Data Portal:

   https://data.gov.in

9. IBM Developer – *Machine Learning Model Deployment Using Streamlit*:

   https://developer.ibm.com/

10. Google Cloud Blog – *Deploying ML Apps with Streamlit and GCP*:

    https://cloud.google.com/blog

11. **UCI Machine Learning Repository – Housing Data Set**
    https://archive.ics.uci.edu/ml/datasets/Housing

12. **Towards Data Science – Regression Models Explained**
    https://towardsdatascience.com/linear-vs-nonlinear-models

13. **Analytics Vidhya – Complete Guide to Regression Models**
    https://www.analyticsvidhya.com/blog/2021/07/a-complete-guide-to-regression-models/

14. **Stack Overflow – Streamlit Implementation Questions**
    https://stackoverflow.com/questions/tagged/streamlit

15. **Medium – End-to-End House Price Prediction using ML**
    https://medium.com/analytics-vidhya/end-to-end-house-price-prediction-using-machine-learning-

    bc3b4c5a2e10

16. **Open Government Data Platform India – Real Estate-related Datasets**
    https://data.gov.in/catalogs/real-estate

17. **Journal of Big Data – Predictive Modeling for Housing Prices**
    https://journalofbigdata.springeropen.com/

18. **ResearchGate – Machine Learning Models for Property Valuation**
https://www.researchgate.net/publication/344202955

19. **PythonAnywhere – Hosting Streamlit Web Apps**
https://help.pythonanywhere.com/pages/Streamlit/

20. **Microsoft Learn – ML Model Deployment with Azure & Streamlit**
https://learn.microsoft.com/en-us/azure/machine-learning/how-to-deploy-streamlit-app

21. Medium – Deploying Machine Learning Models with Streamlit
https://medium.com/swlh/deploying-machine-learning-models-using-streamlit-8b8b0b60a4a8