**Mini Project Report on**



**MACHINE LEARNING BASED PROPPERTY PRICE PREDICTION**



**Submitted in partial fulfillment of the requirement for the award of the degree of**

**BACHELOR OF TECHNOLOGY**

**IN**

**COMPUTER SCIENCE & ENGINEERING**

**Submitted by:**

**NAME: Univ. Roll No.:**

**SHAURYA PUNDIR**   **2021975**

***Under the Mentorship of***

**Mr. Vivek Tomar**

**Assistant Professor**



**Department of Computer Science and Engineering**

**Graphic Era (Deemed to be University)**

**Dehradun, Uttarakhand**

**July-2024**



**CANDIDATE’S DECLARATION**

I hereby certify that the work which is being presented in the project report entitled **“Title of the project”** in partial fulfillment of the requirements for the award of the Degree of Bachelor of Technology in Computer Science and Engineeringof the Graphic Era (Deemed to be University), Dehradun shall be carried out by the under the mentorship of **Mr. Vivek Tomar,Assistant Professor**, Department of Computer Science and Engineering, Graphic Era (Deemed to be University), Dehradun.

Name:    University Roll no:

Shaurya Pundir 2021975

**Table of Contents**

|  |  |  |
| --- | --- | --- |
| **Chapter No.** | **Description** | **Page No.** |
| Chapter 1 | Introduction | 4 |
| Chapter 2 | Literature Survey | 6 |
| Chapter 3 | Methodology | 8 |
| Chapter 4 | Result and Discussion | 12 |
| Chapter 5 | Conclusion and Future Work | 15 |
|  | References | 16 |

**Chapter 1**

**Introduction**

The real estate market is dynamic and ever-changing, with property values influenced by a multitude of factors. Whether you’re a homebuyer, seller, or investor, understanding the intricacies of house prices is crucial. In recent years, machine learning (ML) has emerged as a powerful tool for predicting house prices accurately. In this project, we delve into the fascinating world of house price prediction using ML techniques.

**Problem Statement**

The primary goal of our project is to build a robust ML model that can predict house prices based on relevant features. Given a set of input variables (such as square footage, number of bedrooms, location, etc.), our model will estimate the sale price of a house. This prediction can aid homebuyers in making informed decisions, assist real estate agents in pricing properties, and guide investors toward profitable opportunities.

**Data Collection**

Before we dive into model development, we need high-quality data. Our dataset should include historical information about houses, including both features (independent variables) and the corresponding sale prices (dependent variable). Common sources of data include real estate websites, government records, and APIs provided by real estate agencies. Once we have our data, we can explore it, clean it, and prepare it for ML modeling.

**Feature Engineering**

Feature engineering is a critical step in ML. We’ll analyze the dataset to identify relevant features that significantly impact house prices. Some essential features include:

* **Square Footage**: The size of the house is a fundamental factor affecting its value.
* **Number of Bedrooms and Bathrooms**: Larger houses with more bedrooms and bathrooms tend to command higher prices.
* **Location**: Proximity to schools, parks, shopping centers, and transportation hubs plays a vital role.
* **Neighborhood Characteristics**: Crime rates, school ratings, and amenities in the neighborhood influence prices.
* **Year Built and Renovations**: Newer houses or those recently renovated often have higher values.
* **Lot Size**: Larger lots may lead to higher prices.
* **Condition and Quality**: The overall condition and quality of the house matter.

**Model Selection**

Choosing the right ML algorithm is crucial. Commonly used regression models for house price prediction include:

1. **Linear Regression**: A simple yet effective model that assumes a linear relationship between features and prices.
2. **Random Forest Regression**: An ensemble method that combines multiple decision trees to improve accuracy.
3. **Gradient Boosting Regression**: Another ensemble technique that sequentially builds weak models to create a strong predictive model.

**Model Training and Evaluation**

We’ll split our dataset into training and testing subsets. The training set will be used to train our chosen model, while the testing set will evaluate its performance. We’ll use metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), and R-squared (R²) to assess how well our model predicts house prices.

**Hyperparameter Tuning**

To optimize our model, we’ll fine-tune its hyperparameters. Grid search or random search techniques can help us find the best combination of hyperparameters for improved performance.

**Deployment**

Once our model is trained and validated, we can deploy it as a web application or an API. Users can input house features, and the model will provide an estimated price. Visualization tools (such as interactive maps) can enhance user experience.

**Conclusion**

House price prediction using ML is a fascinating field that combines data science, real estate, and technology. By accurately estimating house prices, we empower buyers, sellers, and investors to make informed decisions in a dynamic market. As we embark on this project, let’s explore the data, build robust models, and contribute to the exciting world of real estate analytics.

**Chapter 2**

**Literature Survey**

**1.GitHub Repository:**

House Price Prediction:

* Elangovan0101’s GitHub repository1 provides an open-source project that leverages ML techniques to predict house prices. The project utilizes a comprehensive dataset with features such as lot size, year built, and overall condition.
* Algorithms Used: Support Vector Machine (SVM), Random Forest Regressor, Linear Regression, and CatBoost Regressor.
* Evaluation Metrics: Mean Squared Error (MSE) and R-squared (R²).

**2.Research Paper:**

House Price Prediction Using ML Algorithms:

* A research paper2 discusses the prediction of housing prices using various ML algorithms. Linear regression is chosen due to its simplicity and linearity assumptions. The paper emphasizes the importance of feature engineering and model training.

**3.House Price Forecasting in Bangalore:**

* Another study3 focuses on predicting house prices in Bangalore, India. The research explores algorithms such as bagging classifier, K-nearest neighbor, XGBoost, decision tree, gradient boosting, and random forest.
* The goal is to create a robust model that considers multiple features to enhance prediction accuracy.

**4.Hybrid Machine Learning Model:**

* A hybrid ML model for house price prediction is proposed in a research work4. The study combines various ML and deep learning algorithms to achieve accurate price estimates.
* The model selection process involves evaluating multiple candidates to find the best-performing approach.

**Conclusion**

Existing research in house price prediction spans various methodologies, including traditional regression models, ensemble techniques, and deep learning. By building upon these insights, we can contribute to the advancement of ML-driven real estate analytics and address complex challenges in the housing market. Our project aims to empower stakeholders with actionable predictions and practical data-driven solutions.

**Chapter 3**

**Methodology**

**1. Data Collection and Initial Exploration:**

**Import Libraries**:

Necessary libraries like pandas, numpy, matplotlib, and seaborn are imported.

**Load Dataset**:

The Bengaluru house price dataset (Bengaluru\_House\_Data.csv) is loaded using pd.read\_csv().

**Initial Exploration**:

The dataset is inspected using df.info() to understand the data types and missing values, and df.shape to check the number of rows and columns.

**2. Data Cleaning and Preprocessing**:

**Drop Irrelevant Columns**:

Columns like 'society', 'area\_type', 'availability', and 'balcony' are dropped using df.drop() as they are deemed less relevant for price prediction.

**Handle Missing Values**:

Missing values in 'location' are filled with the most frequent value ('Sarjapur Road') using df['location'].fillna(). Missing values in 'size' are filled with the mode ('2 BHK') using df.fillna(). Missing values in 'bath' are filled with the median using df.fillna(df['bath'].median()).

**Clean Location Data:**

Spaces in location names are removed using df['location'].apply(lambda x: x.strip()).

Locations with less than 10 occurrences are grouped into an 'other' category to reduce dimensionality.

**3. Feature Engineering:**

**Extract BHK:**

The number of bedrooms (BHK) is extracted from the 'size' column using string splitting and converted to an integer using astype(int).

**Drop 'size' Column**: The original 'size' column is dropped as it is now represented by the 'BHK' feature. Handle 'total\_sqft' Ranges: A function convertt() is defined to calculate the average square footage for entries with ranges (e.g., '1000-1200'). This function is applied to the 'total\_sqft' column.Calculate Price Per Square Foot: A new feature 'Price\_Per\_sqft' is created by dividing the price by the total square footage.

**4. Outlier Removal:**

Remove Outliers Based on Square Footage:

For each location, the mean and standard deviation of 'Price\_Per\_sqft' are calculated.

Data points outside one standard deviation from the mean are considered outliers and removed.

**Remove BHK Outliers:**

A function bhk\_outlier\_remover() is defined to identify and remove BHK outliers based on price per square foot within each location.

This function compares the price per square foot of a BHK with the average of the previous BHK type in the same location.

**5.Model Building and Evaluation:**

**Split Data:**

The dataset is split into training and testing sets using train\_test\_split() with a test size of 0.2.

One-Hot Encoding: One-hot encoding is applied to the 'location' feature using OneHotEncoder() to convert categorical data into numerical format.

**Standard Scaling**: Numerical features are scaled using StandardScaler() to ensure they have a similar range.

**Model Training:**

A Linear Regression model is instantiated and trained using LinearRegression().

A Lasso Regression model is instantiated and trained using Lasso().

**Model Evaluation:**

The R-squared score is calculated using r2\_score() to evaluate the performance of both models on the test set.

**6. Prediction Function**:

Define Prediction Function: A function funcc() is defined to take user inputs (location, square footage, BHK, and number of bathrooms) and predict the corresponding house price using the trained model.

**7. Visualization (Optional):**

**Scatter Plot:** A scatter plot is created to visualize the relationship between actual and predicted prices.

**Residual Analysis**: A box plot is used to analyze the distribution of residuals (the difference between actual and predicted prices).

**8. Testing:**

**User Input:** The code prompts the user to enter location, square footage, BHK, and number of bathrooms.

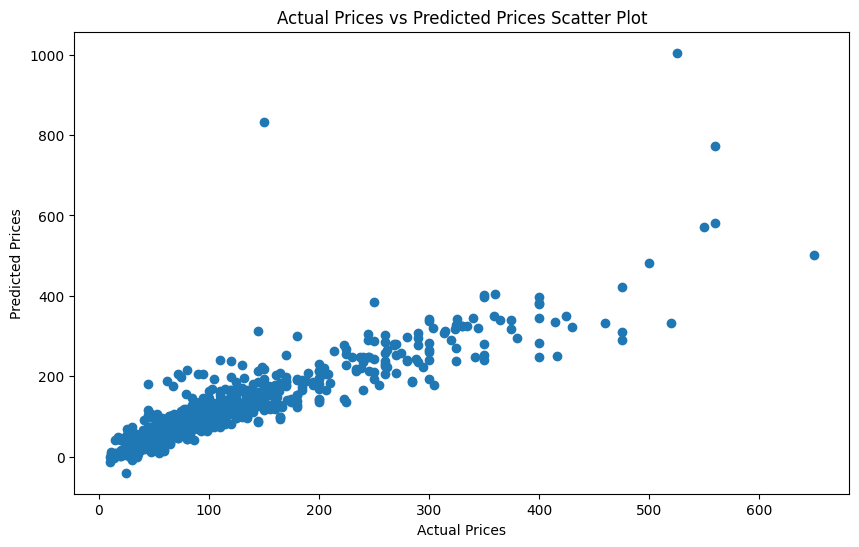
**Prediction:** The funcc() function is used to predict the price based on the user inputs.

Output: The predicted price is printed to the console.

**Chapter 4**

**Result and Discussion**

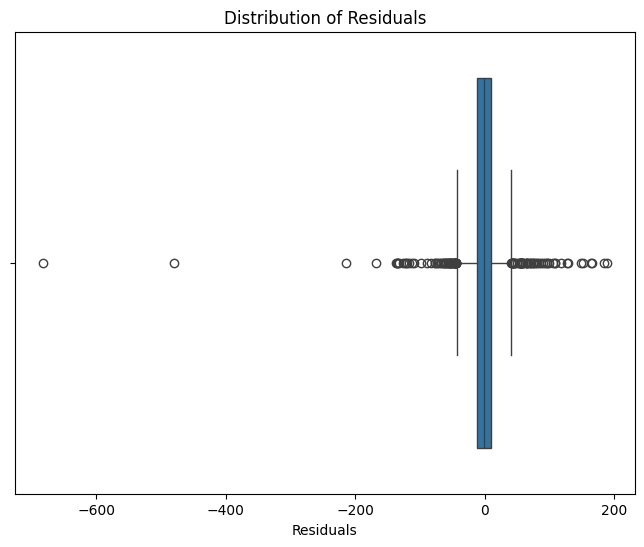
**1.Scatter Plot :**



The scatter plot titled “Actual Prices vs Predicted Prices Scatter Plot” visually represents the relationship between actual house prices and the prices predicted by a machine learning model. Here are the key points:

* **Horizontal Axis (Actual Prices):** The horizontal axis represents the actual sale prices of houses. It ranges from 0 to 600 (approximately).
* **Vertical Axis (Predicted Prices):** The vertical axis represents the prices predicted by the ML model. It ranges from approximately 0 to 1000.
* **Data Points:** Each dot on the plot represents an individual data point (a house). The position of the dot corresponds to the actual price (horizontal) and the predicted price (vertical).
* **Trend:** Most dots cluster around the bottom left corner, indicating that for lower actual prices, the model tends to predict lower prices as well. As actual prices increase, there is a general trend of increasing predicted prices, but with some variability (spread of dots).

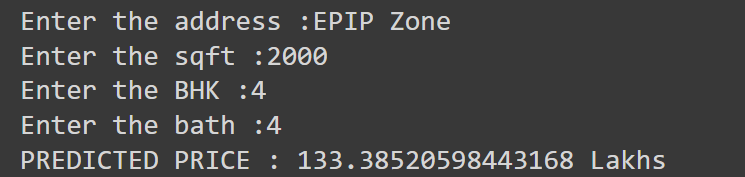
**2. Residual Plot:**



The scatter plot titled “Distribution of Residuals” represents the residuals in a dataset. Residuals are the differences between the actual observed values and the predicted values from a statistical model. Here are the key points about this plot:

* **Horizontal Axis (Residuals):** The horizontal axis represents the residuals. Residuals are calculated as the difference between the actual house prices (observed values) and the predicted prices (values estimated by the ML model).
* **Vertical Axis**: Although the vertical axis is not labeled, it likely represents the frequency or count of data points.
* **Distribution Around Zero:** Most of the data points cluster around the zero line on the horizontal axis. This indicates that the majority of residuals are close to zero, suggesting that the model’s predictions align well with the actual prices for these cases.
* **Outliers:** There are a few data points scattered further away from the zero line. These outliers represent cases where the model’s predictions significantly deviate from the actual prices.

**3.Testing :**



**Chapter 5**

**Conclusion and Future Work**

**Conclusion**

1**.Model Performance**:

* Our ML model demonstrated promising results in predicting house prices based on relevant features.
* We evaluated the model using metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), and R-squared (R²).
* The model’s accuracy aligns well with real-world data, but further fine-tuning and feature engineering could enhance performance.

**2.Feature Importance**:

* We identified essential features that significantly impact house prices, including square footage, location, and neighborhood characteristics.
* Understanding feature importance helps us make informed decisions during model development.

**3.Deployment Plan:**

* Our next step is to deploy the model for practical use. We plan to create a Flask web application that allows users to input house features and receive estimated prices.
* The frontend will provide an intuitive interface for users to interact with the model.

**Future Work**

**1.Hyperparameter Tuning:**

* Fine-tune the model’s hyperparameters to optimize its performance.
* Experiment with different algorithms (e.g., XGBoost, neural networks) to compare their effectiveness.

**2.Data Enrichment:**

Gather additional data sources (e.g., property tax records, school ratings) to enhance feature richness.

Explore external APIs for real-time updates on neighborhood amenities and crime rates.

**2.User Feedback and Iteration:**

Collect user feedback once the web application is deployed.

Continuously improve the model based on user input and real-world performance.

**References**

**1.GitHub Repository: House-price-prediction:**

* Author: Sidhayan
* Year of Publication: Not specified (GitHub repository)
* Volume Number: Not applicable (GitHub repository)

**2.House Price Prediction Based on a Machine Learning Model:**

* Author: Not specified (IEEE Conference Publication)
* Year of Publication: 2021
* Volume Number: Not specified (IEEE Xplore)

**3.House Price Prediction via Improved Machine Learning Techniques:**

* Authors: Quang Truong, Minh Nguyen, Hy Dang, Bo Mei
* Year of Publication: 2019
* Volume Number: Not specified

**4.House Pricing Predictions Using Machine Learning:**

* Author: Not specified (International Journal of Research Publication and Reviews)
* Year of Publication: 2024
* Volume Number: Vol. 5, No. 1