A PROJECT REPORT
on

"HOUSE PRICE PREDICTION"

Submitted to
KIIT Deemed to be University

In Partial Fulfillment of the Requirement for the Award of

BACHELOR'S DEGREE IN
COMPUTER SCIENCE & ENGINEERING

BY

ABIR BANERJEE	2205175
ANKIT BAKSHI	2205183
ASHUTOSH KR. SINGH	2205193
ADITYA KUMAR	2205349

UNDER THE GUIDANCE OF
AJIT KUMAR PASAYAT

SCHOOL OF COMPUTER ENGINEERING
KALINGA INSTITUTE OF INDUSTRIAL TECHNOLOGY
BHUBANESWAR, ODISHA - 751024
April 2025

A PROJECT REPORT

on

"HOUSE PRICE PREDICTION"

Submitted to

KIIT Deemed to be University

In Partial Fulfillment of the Requirement for the Award of

BACHELOR'S DEGREE IN COMPUTER SCIENCE & ENGINEERING

BY

ABIR BANERJEE	2205175
ANKIT BAKSHI	2205183
ASHUTOSH KR. SINGH	2205193
ADITYA KUMAR	2205349

UNDER THE GUIDANCE OF
AJIT KUMAR PASAYAT

SCHOOL OF COMPUTER ENGINEERING

KALINGA INSTITUTE OF INDUSTRIAL TECHNOLOGY

BHUBANESWAE, ODISHA -751024
April 2025

KIIT Deemed to be University

School of Computer Engineering
Bhubaneswar, ODISHA 751024

CERTIFICATE

This is certify that the project entitled

"HOUSE PRICE PREDICTION"

submitted by

ABIR BANERJEE	2205175
ANKIT BAKSHI	2205183
ASHUTOSH KR. SINGH	2205193
ADITYA KUMAR	2205349

is a record of bonafide work carried out by them, in the partial fulfillment of the requirement for the award of Degree of Bachelor of Engineering (Computer Science & Engineering) at KIIT Deemed to be university, Bhubaneswar. This work is done during year 2024-2025, under our guidance.

Date: 22/03/2025

(Ajit Kumar Pasayat)
Project Guide

Acknowledgements

ABSTRACT

In this project, we developed a robust prophetic model for casing prices using Machine literacy ways, specifically the XGBoost algorithm. By assaying a comprehensive data-set containing property features similar as area, bedroom count, bathroom count, and stories. The model identifies crucial determinants of property valuation to produce an accurate price determination system. The data-set includes property records with attributes, encompassing both numerical features(price, area, bedrooms, bathrooms, stories, parking) and categorical features(main-road, guestroom, basement, hot-water-heating, air-conditioning, pref-area, furnishing-status).

The methodology follows a structured workflow including data pre-processing, exploratory analysis, feature engineering, and model training with rigorous confirmation. The performing model not only provides accurate price prognostications but also offers perceptivity into feature significance, helping stakeholders understand which property characteristics most significantly impact value. This tool empowers buyers, merchandisers, and investors to make further informed opinions in the real estate request by furnishing data- driven property valuations grounded on objective parameters rather than private assessments.

Keywords: Machine Learning, Real Estate valuation, Streamlit Web Application, Inflation-adjusted Prediction, Predictive analysis

Contents

List of Figures

Chapter 1

Introduction

The real estate market is a cornerstone of the global economy, continuously shaped by evolving consumer demands, economic fluctuations, and technological advancements that redefine property valuation and investment strategies.

The real estate market is influenced by numerous factors including property characteristics (bedrooms, bathrooms, square footage) and broader economic conditions. Understanding these elements is essential for informed property valuation and investment decisions Understanding these rudiments is essential for informed property valuation and investment opinions. Over time, the request has endured oscillations driven by government programs, technological advancements, and environmental enterprises, with ultramodern data analytics enabling more precise request analysis.

Real estate represents one of the largest investment sectors encyclopedically, with property values determined by a complex interplay of features. The capability to directly prognosticate these values provides significant advantages to buyers, merchandisers, investors, and real estate professionals likewise. Traditional valuation styles frequently calculate heavily on private assessments, creating openings for data- driven approaches to ameliorate delicacy and thickness.

This report is structured as follows:
- ➢ Chapter 2 presents a detailed literature review, discussing the current advancements in house price prediction models, existing methodologies, and their limitations.
- ➢ Chapter 3 outlines the problem statement, requirement specifications, and system design, including the key components and constraints of the proposed model.
- ➢ Chapter 4 explains the methodology and implementation details, covering data preprocessing, feature engineering, model selection, and training processes.

➢ Chapter 5 focuses on experimental results and model evaluation, comparing the proposed approach with traditional prediction models.

➢ Chapter 6 concludes with key findings and highlights potential areas for future research and improvements.
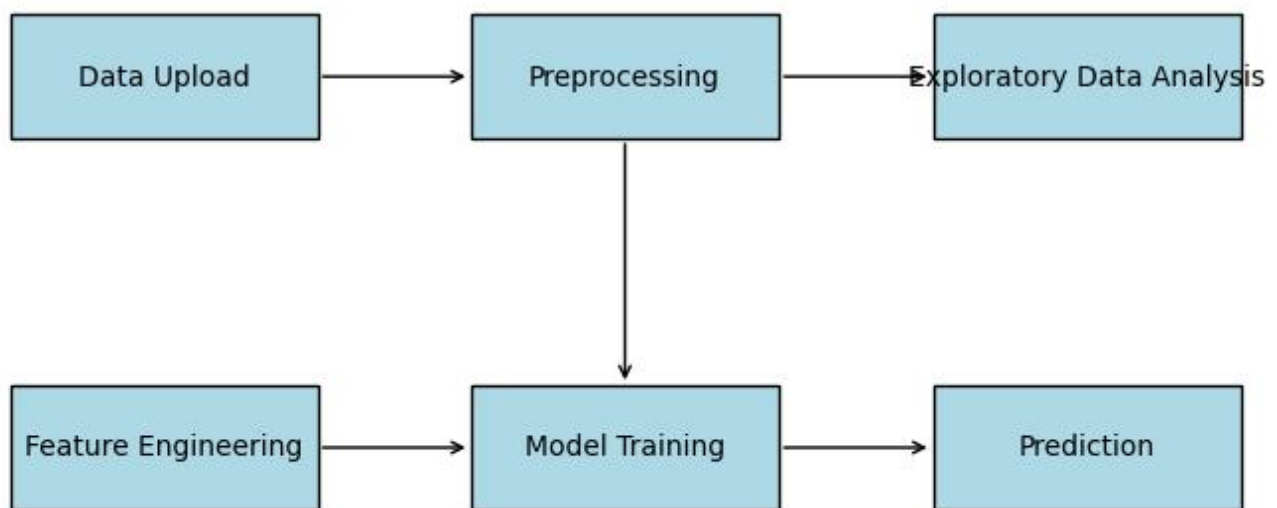
Housing Price Prediction Project Setup

Figure 1.1: House Price Prediction Project Setup

Chapter 2

Basic Concepts/ Literature Review

2.1 Introduction ………….

One of the biggest problems in real estate, finance, and urban planning is predicting home prices. Conventional approaches typically use historical sales data to forecast future prices, but they ignore inflation adjustment, which is one of the most crucial factors for precise long-term forecasts. Along with graphical representation and an intuitive web dashboard based on Streamlit, real-time predictions are also offered.

2.2 Feature Engineering ………….

Feature engineering improves model performance by transforming raw data into useful formats. The **feature_engineering.py** file in this work uses the **create_features() function** to apply different feature engineering techniques:

➢ **Interaction Features:** By multiplying related numerical features, these are intended to reveal intricate relationships:

> 💡 New Feature = Feature A * Feature B

> Example: A feature that captures combined effects,
>   lot_size *number_of_bedrooms

➢ **Polynomial Features:** Non-linear relationships are better represented by higher powers:

> 💡 New Feature = $(\text{Feature})^2$

> For instance, the effects of aging are taken into account by,
>   house_age_squared = house_age^2

➢ **Feature Ratios:** Ratio-based transformations are used to normalize specific features:

$$Ratio = FeatureA/FeatureB$$

Example:
  price_per_sqft = price / total_sqft.

➢ **Inflation Adjustment:** preprocessing.py's **adjust_for_inflation() function** ensures that past home prices match present values:

$$P' = P * (1 + r)^{(t2 - t1)}$$

where,

- P′ = Adjusted Price
- P = Original Price
- t2 = Target Year
- t1 = Base Year
- r = Inflation Rate

2.3 Feature Selection ………….

By removing superfluous variables, feature selection seeks to increase the efficacy of the model. The **feature_engineering.py feature_selection() function** uses:

➢ **SelectKBest (f_regression):** Chooses the top k features according to how they relate to the variable of interest.

> **Recursive feature elimination (RFE):** It uses linear regression to gradually eliminate less important features.

> **Random Forest Feature Importance:** Assesses feature significance by using decision tree divisions.

2.4 Data Pre-Processing ………….

Before training the model, data uniformity is ensured by the **preprocessing.py** module.

2.4.1: Handling Missing Values
Included in the **handle_missing_values() function** are:
- Deleting rows with an excessive amount of missing data.
- Utilizing the mean or median method to impute missing values.

2.4.2: Outlier Detection and Treatment
The Interquartile Range (IQR) method is implemented by the **detect_outliers_func() function**:

💡 IQR = Q3 - Q1

Lower Bound = Q1 - 1.5 * IQR

Upper Bound = Q3 + 1.5 * IQR

#Values that are not within these bounds are either capped or discarded.

2.4.3: Feature Scaling
Numerical uniformity is maintained through scaling:
- For Regression models, **StandardScaler** (zero mean, unit variance) works well.
- For Deep learning models benefit most from **MinMaxScaler**, which normalizes values between 0 and 1.

2.4.4: Categorical Encoding
Categorical variables are processed by the **preprocess_data() function** using:
- **One-Hot Encoding:** Binary columns are created from categorical variables using one-hot encoding. Three binary features, Neighborhood_A, Neighborhood_B, and Neighborhood_C, for instance, will be used to represent a categorical variable Neighborhood with values A, B, and C.
- **Label Encoding:** Perfect for tree-based models, this method gives categories unique numerical identifiers, such as A -> 0, B -> 1, and C -> 2.

2.5 Model Selection & Training ………….

The script **model_training. py** is used to develop various models via the **train_model() function**.

2.5.1: Traditional Machine Learning Models
The trained models include:
- **Linear Regression:** A statistical technique known as linear regression makes the assumption that the predictors and the outcome variable have a straight-line relationship.
- **Ridge Regression & Lasso Regression:** These are types of linear regression that help avoid over-fitting by incorporating penalties:
  - Ridge Regression: This applies L2 regularization to add a penalty that affects the square of the coefficients.
  - Lasso Regression: This uses L1 regularization, which can reduce some coefficients to zero, effectively selecting features.
- **Decision Tree:** This model arranges data according to a set of branching rules, producing a tree-like structure in which each branch denotes a feature-related choice.
- **Random Forest:** By averaging the predictions from multiple decision trees, this technique improves accuracy.
- **Gradient Boosting & XGBoost:** Gradient Boosting and XGBoost are tree-based models that improve predictions over time by fixing errors in previous models. They work best with well-structured data.

All models undergo cross-validation to confirm that they can generalize well.

2.6 Model Evaluation & Visualization …………

2.6.1: Evaluation Metrics
The models are assessed according to:
- **MAE, or mean absolute error:** Average of absolute difference between predicted and actual values.
  - MAE = 0 : Perfect model
  - Lower MAE : More accurate prediction
  - Higher MAE : Higher prediction errors

$$MAE = \frac{1}{n} \sum |y - \hat{y}|$$

- **Root Mean Squared Error (RMSE):** Average difference between values predicted by a model and actual values.
  - RMSE = 0 : Perfect model
  - Lower RMSE : Better prediction
  - Higher RMSE : Higher deviations from actual values

$$RMSE = \sqrt{\frac{1}{n} \sum (y - \hat{y})^2}$$

- **R-Squared (R²):** The model's ability to capture variation in the outcome variable due to an independent variable.
  - $R^2 = 1$: Perfect fit (Model explains all variability)
  - $R^2 = 0$: Model explains none of the variability
  - $R^2 < 0$: Worst fit (Model is poorly fitted)

$$R^2 = 1 - \frac{\sum (y - \hat{y})^2}{\sum (y - \bar{y})^2}$$

2.6.2: Visualizations

In order to comprehend how models function and acquire insights into the data, visualization tools are essential:

➢ **Plot_correlation_matrix()**'s correlation heatmaps highlight highly correlated variables by displaying the relationships between features.

➢ **The Significance of Features:** The main predictors that affect model results are highlighted in plots (**plot_feature_importance()**).

➢ **Values Predicted vs. Actual (plot_prediction_vs_actual()):** This assesses precision by contrasting model predictions with actual prices.

➢ **Residual Plots (plot_residuals_vs_predicted()):** These examine prediction errors and reveal potential model biases.

2.7 Web Application for Interactive Prediction ………….

The Streamlit (**app.py**) dashboard provides a user-friendly platform that allows users to:

➢ Use **handle_missing_values()** and **preprocess_data()** to upload and prepare data.

➢ Use the dynamic graphs from **plot_correlation_matrix()** and **plot_feature_importance()** to perform exploratory data analysis.

➢ Use **train_model()** and **feature_selection()** to train models and view results comparisons instantly.

➢ **Adjust_for_inflation()** allows users to model possible future price changes by producing inflation-adjusted forecasts.

Chapter 3

Problem Statement / Requirement Specifications

Problem Statement …………

Predicting home prices is a significant problem in urban planning, finance, and real estate. Location, property size, market demand, and macroeconomic factors like inflation all affect real estate prices. Inaccurate long-term price estimates result from traditional house price prediction models' failure to take inflation-adjusted price trends into account. The goal of this project is to develop a machine learning-based system for predicting home prices that dynamically accounts for inflation. Users will also be able to enter property details and receive real-time inflation-adjusted price predictions through an interactive web dashboard powered by Streamlit.

Requirement Specifications (IEEE SRS Format) …………

1. Introduction

- ➢ **Purpose**: Create a model for predicting home prices using AI that incorporates inflation-adjusted forecasting.
- ➢ **Scope**: Using machine learning models, the system forecasts home prices and makes real-time price adjustments based on past inflation data.
- ➢ **Definitions, Acronyms, and Abbreviations**:
  - ■ **CPI (Consumer Price Index)** - A measure of inflation affecting real estate prices.
- ➢ **References**: Model training, feature engineering, data preprocessing, and visualization scripts from the project files are cited.

2. Overall Description

- ➤ **Product Perspective**: Streamlit will be used to deploy the system as a web-based application.
- ➤ **User Characteristics**: Financial analysts, investors, and real estate brokers looking for insights into prices adjusted for inflation.
- ➤ **Constraints**:
  - Accessibility of historical price and inflation data.
  - Computational effectiveness when managing sizable datasets.
  - Time-series forecasting accuracy of machine learning models.

3.1 Project Planning ………….

3.1.1: Project Planning Steps

1. **Problem Identification**: Recognizing the weakness in conventional models for predicting home prices.
2. **Data Collection**: Compiling historical inflation and home price data.
3. **Data Preprocessing**:
   a) Taking care of missing values (preprocessing.py's handle_missing_values()).
   b) Detection of outliers (detect_outliers_func()).
   c) Categorical variable scaling and encoding (preprocess_data()).
4. **Feature Engineering & Selection**:
   a) Generating polynomial and interaction features (create_features() in feature_engineering.py).
   b) Deciding which features are most pertinent (feature_selection()).
5. **Model Selection & Training**:
   a) Regression model training (train_model() in model_training.py).
6. **Model Evaluation**: Model performance is assessed using error metrics like RMSE, MAE, and R2 (train_model()).
7. **Web Application Development**:
   a) Streamlit dashboard implementation (app.py).
   b) Incorporating the visualization tools plot_correlation_matrix() and plot_feature_importance().
8. **Testing & Deployment**: Verifying model functionality and launching the web application.

3.2 Project Analysis …………..

- ➤ **Potential Challenges**:
  - ■ Ensuring quality of data for accurate inflation adjustments.
  - ■ Controlling the performance of models on big datasets.
  - ■ Putting in place a user interface that is easy to use for real estate agents.
- ➤ **Ambiguity Analysis**:
  - ■ Accurate inflation forecasts are essential because inflation rates fluctuate over time.
  - ■ How changes in policy and market trends affect real estate prices.

3.3 System Design …………..

3.3.1: Design Constraints

Software Requirements

- ➤ **Python Libraries**: Pandas, NumPy, Scikit-learn, Streamlit.
- ➤ **Machine Learning Models**: Linear Regression, Decision Trees, Random Forest, XGBoost.
- ➤ **Visualization Tools**: Matplotlib, Seaborn, Plotly.
- ➤ **Data Storage**: CSV files or database integration for real estate data.

Hardware Requirements

- ➤ **Processor**: Minimum Intel i5 (or equivalent) for model training.
- ➤ **RAM**: At least 8GB for efficient processing of large datasets.
- ➤ **GPU**: Required for training deep learning models if large datasets are used.

3.3.2: System Architecture

The architecture of the system is designed to ensure a seamless flow from data collection to prediction and visualization. The key components include:

1. Data Collection & Preprocessing Layer

- ➤ **Data Sources**: Historical house price data and inflation data from government and financial institutions.
- ➤ **Preprocessing Functions**:
  - ■ **preprocess_data():** Handles missing values and feature encoding.
  - ■ **detect_outliers_func():** Removes extreme price variations.
  - ■ **adjust_for_inflation():** Adjusts historical prices for inflation.

2. Feature Engineering & Selection Layer

➢ **Feature Engineering Functions**:
- ■ **create_features():** Generates polynomial, interaction, and ratio-based features.
- ■ **feature_selection():** Identifies the most relevant predictors using statistical and ML-based techniques.

3. Model Training Layer

➢ **Machine Learning Models**:
- ■ **train_model():** Implements multiple regression techniques (Linear Regression, Decision Trees, Gradient Boosting, XGBoost).

4. Evaluation & Visualization Layer

➢ **Model Evaluation Functions**:
- ■ **plot_correlation_matrix():** Identifies feature relationships.
- ■ **plot_feature_importance():** Highlights critical predictors.
- ■ **plot_prediction_vs_actual():** Compares predicted vs. actual prices.

➢ **Performance Metrics**:
- ■ RMSE, MAE, and R^2 for model assessment.

5. Web Application Layer

➢ **User Interaction Components**:
  A **Streamlit-based dashboard** for:
  a) Data upload & preprocessing.
  b) Model training & evaluation.
  c) Real-time inflation-adjusted predictions.
  d) Interactive graphs for insights.

3.3.3: Block Diagram Representation
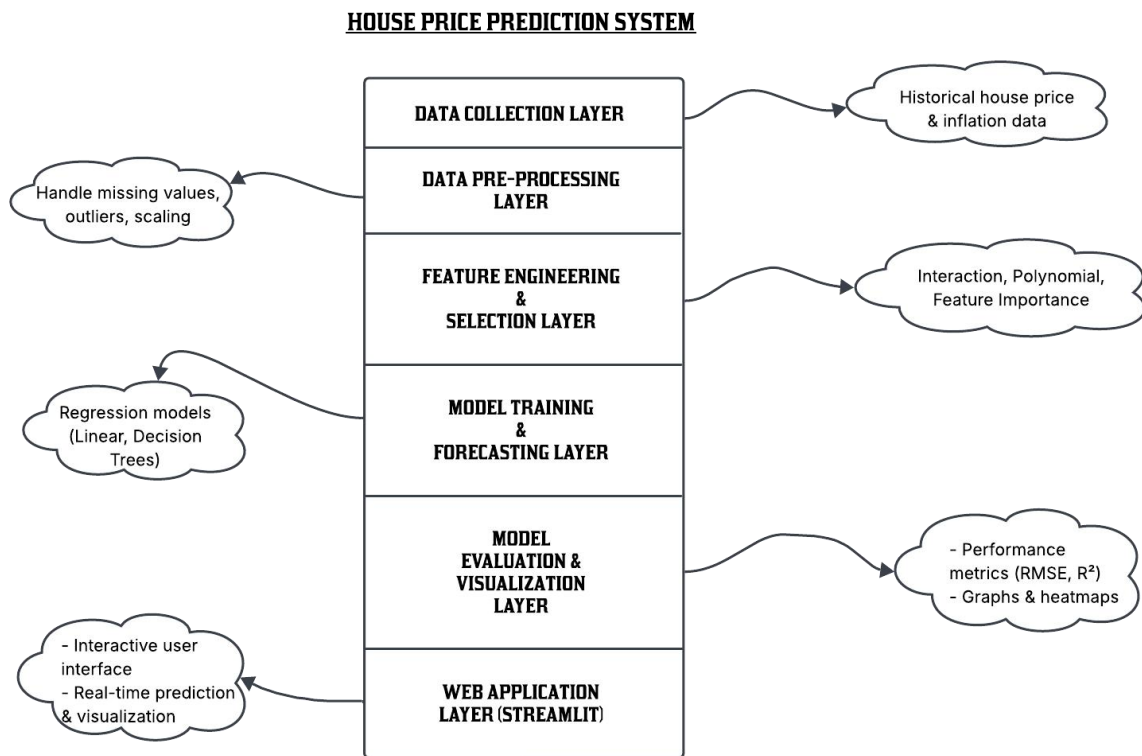
Figure 3.1: Block Diagram Representation

Chapter 4

Implementation

In this section, we present the implementation done by us during the project development.

4.1 Methodology or Proposal …………

Implementation of house price forecasting system adheres to a well-designed methodology in order to produce an accurate prediction along with convenient human interaction. Following is the methodology involved:

4.1.1: Initial Setup

The implementation of this project begins with setting up the necessary environment and importing required libraries to support the functionality of our system.
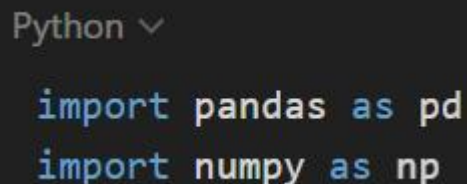
Environment Configuration:

- ➢ Programming Language : Python 3.12.7

- ➢ Integrated Development Environment(IDE) : Virtual Studio Code (VS Code)

Library Imports

For the completion and execution of model, we need to integrate various libraries in our IDE.

Data Processing Libraries:-


```python
import pandas as pd
import numpy as np
```

- ➢ **Pandas:** An open-source library offering efficient data analysis, manipulation, and visualization capabilities, with support for structured data formats through DataFrame objects.

➤ **NumPy:** A fundamental package for scientific computing in Python that provides support for large, multi-dimensional arrays and matrices, along with mathematical functions to operate on these arrays.

Visualization Libraries:-

```python
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
```
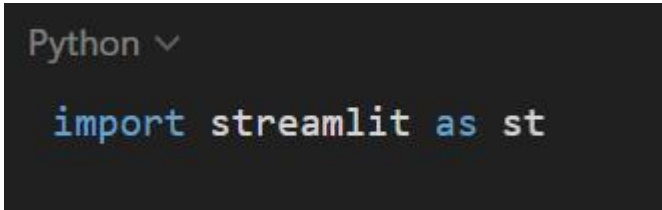
➤ **Matplotlib:** A comprehensive plotting library that provides an object-oriented API for embedding plots into applications, supporting various 2D and 3D plot types.
➤ **Seaborn:** A statistical data visualization library built on top of Matplotlib that provides a high-level interface for creating attractive and informative statistical graphics.
➤ **Plotly:** An interactive, open-source plotting library supporting over 40 unique chart types for statistical, financial, geographic, scientific, and 3D visualizations that can be displayed in web contexts.

Machine Learning Libraries:-

```python
import sklearn
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
import xgboost as xgb
```

➤ **Scikit-learn:** An open-source machine learning library featuring various classification, regression, and clustering algorithms with a consistent API, designed to interoperate with NumPy and SciPy.

➤ **XGBoost:** A distributed, open-source machine learning library implementing gradient boosted decision trees, known for its speed, efficiency, and performance with large datasets

Web Application Framework:-

```python
Python ∨

import streamlit as st
```

> **Streamlit:** An open-source Python library that allows developers and data scientists to quickly build interactive, data-rich web applications without requiring extensive web development knowledge.

4.1.2: Data Collection

The project's statistics came from housing databases and publicly accessible real estate listings. Numerous characteristics are included, including year of construction, square footage, number of bedrooms, number of bathrooms, location, and other determining factors.

4.1.3: Data Preprocessing

The following procedures were used to preprocess the dataset prior to applying machine learning models:

> **Managing Missing Values:** Imputation methods were applied to fill in missing values according to the mode for categorical characteristics and the median or mean for numerical data.

> **Coding Categorical Data:** Using methods like One-Hot Encoding and Label Encoding, categorical variables like location and property type were transformed into numerical values.

> **Outlier Detection and Removal:** To increase model accuracy, outliers were found using statistical methods such as the IQR (Interquartile Range) method and eliminated.

> **Feature Scaling:** To bring features to a same scale, normalization and standardization were used where needed.

4.1.4: Feature Engineering

The predictive power of the model was increased by the use of feature engineering approaches. Here are a few more derived features:

- ✓ The cost per square foot
- ✓ The house's age, as determined by the year it was constructed
- ✓ The distance from significant landmarks (if any)
- ✓ Using feature interaction words to capture intricate relationships

4.1.5: Model Selection and Training

To determine which machine learning model performed the best, a number of models were tested. Among the models taken into consideration are:

- ➢ **Linear Regression**: Used as a baseline model to establish performance benchmarks.
- ➢ **Decision Tree Regressor**: Evaluated for its ability to capture non-linear relationships.
- ➢ **Random Forest Regressor**: Applied to enhance predictive accuracy through ensemble learning.
- ➢ **Gradient Boosting Machines (XGBoost)**: Tested for improved efficiency and performance in handling complex data.
- ➢ **Neural Networks**: Explored for deep learning-based predictions.

4.1.6: Model Evaluation

To assess model performance, various evaluation metrics were used:

- ➢ **Mean Absolute Error (MAE)**: Measures the average absolute errors between predicted and actual values.
- ➢ **Mean Squared Error (MSE) & Root Mean Squared Error (RMSE)**: Evaluate the spread of prediction errors.
- ➢ **R-squared (R²) Score**: Indicates how well the model explains variance in the data.

4.1.7: Deployment

Streamlit, an open-source Python package that makes it possible to create interactive web applications, was used to deploy the model with the best performance. In order to obtain real-time house price estimates, a user-friendly interface was developed that allows users to enter property characteristics like location, size, and number of rooms. Rapid iterations and simple model sharing with stakeholders or end users are made possible by the Streamlit app.

```
streamlit run app.py
```

4.2 Testing OR Verification Plan …………

Test ID	Test Case Title	Test Condition	System Behavior	Expected Result
T01	Basic Prediction Accuracy	Input standard house features	System generates price prediction	Prediction within 15% of actual price
T02	Outlier Handling	Input extreme values for features	System identifies outliers and adjusts prediction	Warning message and reasonable prediction
T03	Missing Data Handling	Omit some feature values	System applies imputation techniques	Prediction generated with minimal accuracy loss
T04	Location Sensitivity	Test same house in different locations	System adjusts predictions based on location	Higher prices for premium locations
T05	Feature Importance	Analyze model outputs	System ranks features by importance	Size and location among top features

4.3 Result Analysis OR Screenshots …………

4.3.1: Preprocessed Housing Data sample

Preprocessed Data

	MedInc	HouseAge	AveRooms	AveBedrms	Population	AveOccup	Latitude	Longitude	PRICE
0	2.3901	1.8211	2.9601	0.3842	-1.2282	-0.0673	1.0287	-1.3089	3.521
1	1.3014	1.8211	0.6148	0.3784	-1.1291	-0.4705	1.0287	-1.3139	3.413
2	0.0896	1.8211	1.0557	0.4988	-1.1179	-1.0516	1.0287	-1.3139	3.422
3	0.2181	1.8211	-0.3872	0.837	-1.3608	-1.1175	1.0287	-1.3139	2.697
4	-0.0366	1.8211	-0.2256	-1.4474	-0.2725	-1.1357	1.024	-1.3139	2.992

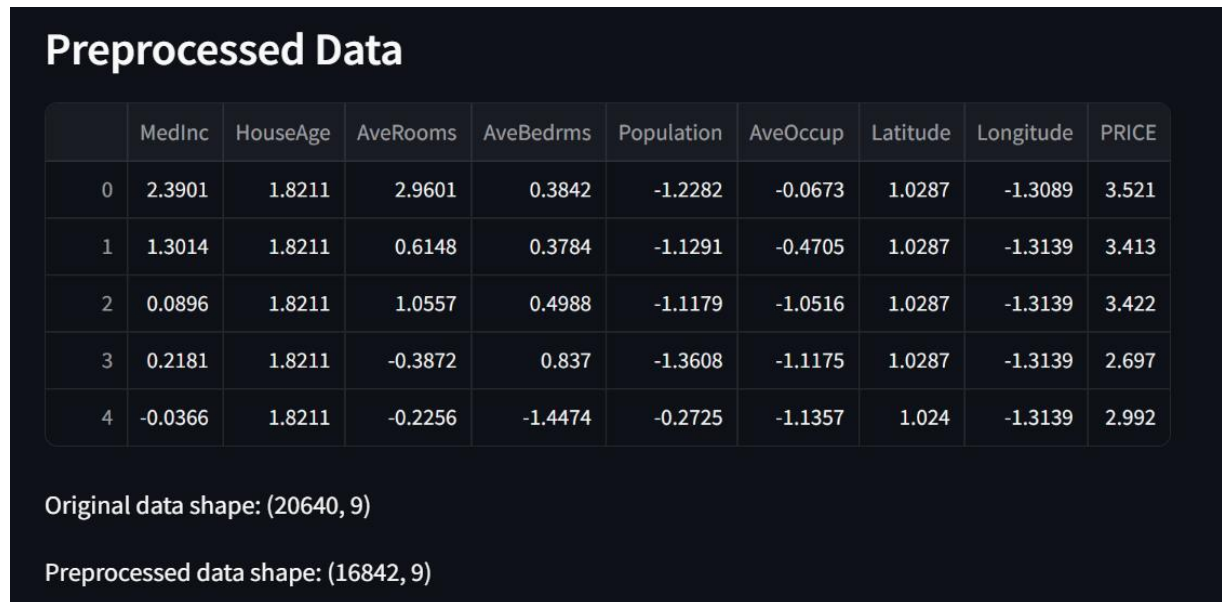Original data shape: (20640, 9)

Preprocessed data shape: (16842, 9)

Figure 4.1: Preprocessed Housing Data sample

This image shows a table of preprocessed data with 9 feature columns including MedInc, HouseAge, and PRICE. Original shape (20640,9), preprocessed (16842,9).

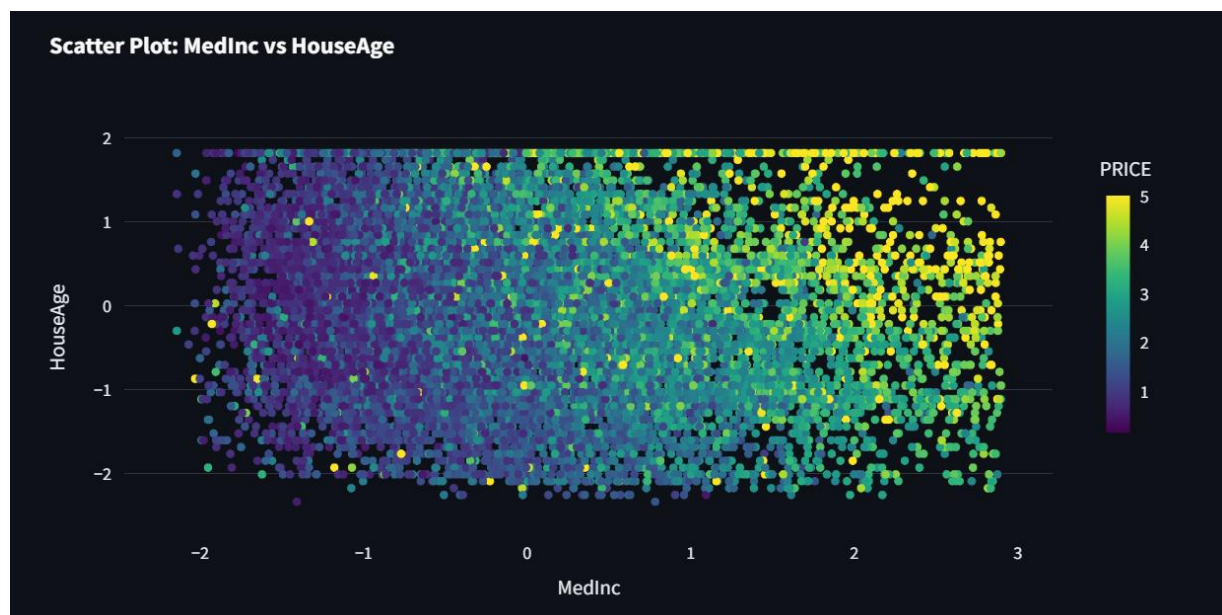### 4.3.2: Scatter Plot: MedInc vs HouseAge Relationship

Figure 4.2: Scatter Plot: MedInc vs HouseAge

Scatter plot showing relationship between median income (MedInc) and house age (HouseAge), with points colored by housing price. Higher income areas (right side) tend to have higher prices (yellow/green), while lower income areas show lower prices (purple/blue).

4.3.3: Feature Correlation Matrix
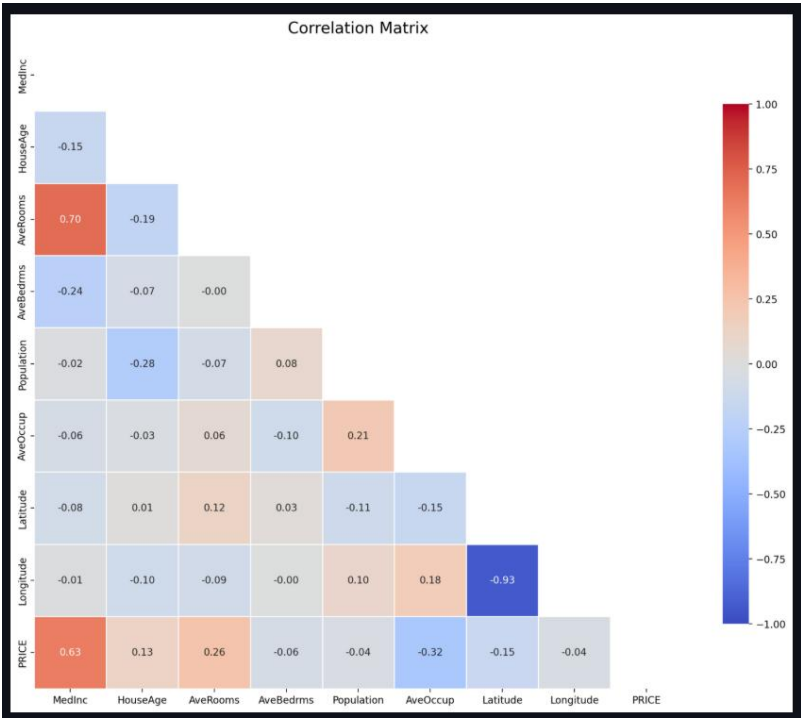
Figure 4.3: Feature Correlation Matrix

A correlation matrix showing relationships between housing features. Strong positive correlations exist between MedInc and PRICE (0.63) and MedInc and AveRooms (0.70). A strong negative correlation (-0.93) appears between Latitude and Longitude. Most other relationships show weak correlations.

4.3.4: Model Performance Metrics

Model Performance

R² Score	MAE	RMSE
0.6077	0.5019	0.6690

Cross-Validation Results

Cross-Validation R² Scores: [0.63380202 0.61990075 0.61417619 0.62159395 0.64496641]

Mean CV R² Score: 0.6269

Standard Deviation: 0.0111

Figure 4.4: Model Performance Metrics

A model performance summary showing an R² score of 0.6077, MAE of 0.5019, and RMSE of 0.6690. Cross-validation results display consistent R² scores across five folds (averaging 0.6269) with a low standard deviation (0.0111), indicating stable model performance.

4.3.5: Predicted vs Actual Housing Values

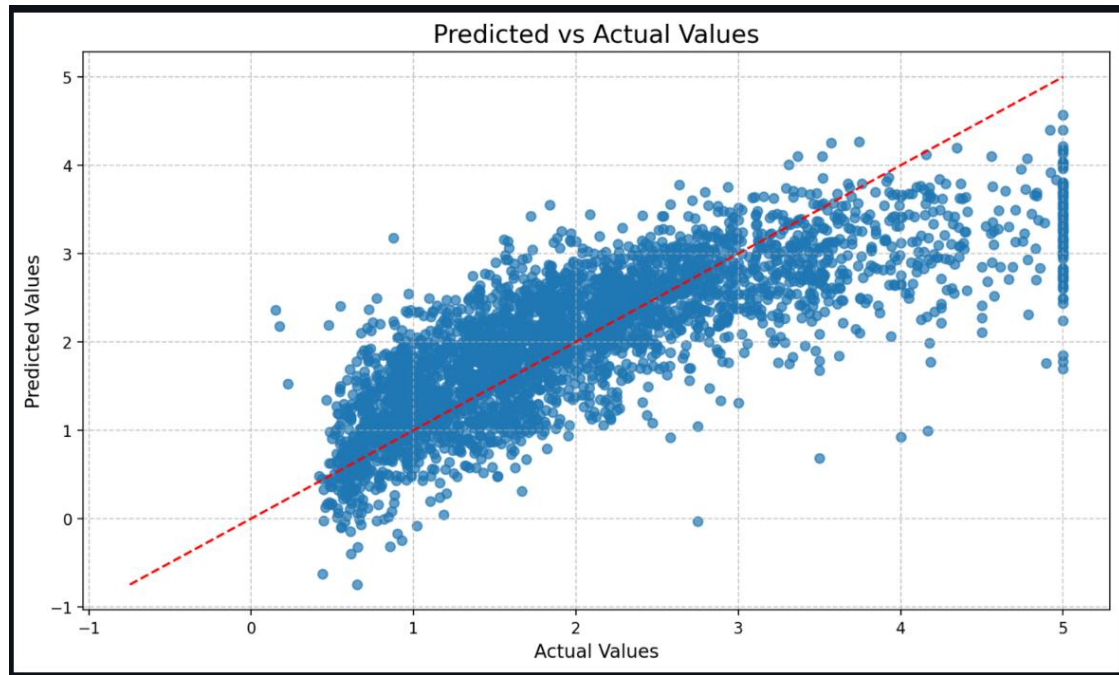Figure 4.5: Predicted vs Actual Housing values

A scatter plot comparing predicted versus actual housing values with a red diagonal reference line. Points cluster along this line, showing moderate predictive accuracy. More scatter appears at higher values, suggesting the model struggles with accurately predicting higher-priced homes.

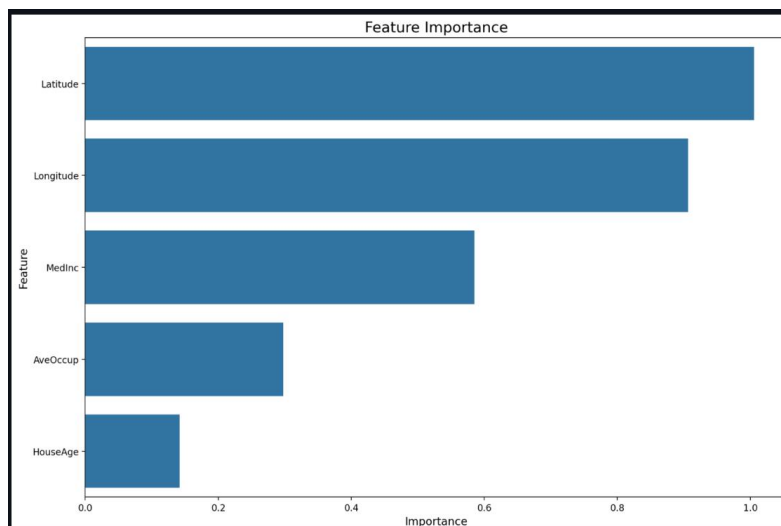### 4.3.6: Feature Importance Ranking

Figure 4.6: Feature Importance Ranking

Feature importance bar chart showing geographic features have the highest impact on the model. Latitude ranks highest, followed by Longitude, then MedInc (median income). AveOccup and HouseAge have minimal importance, suggesting location strongly determines housing prices.

4.3.7: Housing Price Forecast with Inflation Adjustment

Prediction Result

Predicted house price (in 2025): $2.02

Inflation-adjusted price (in 2030): $2.73

Applied 6.2% annual inflation over 5 years.

Total price increase due to inflation: 35.1%

Figure 4.7: Housing Price Forecast with Inflation Adjustment

Prediction result showing estimated house price of $2.02 in 2025, with inflation-adjusted price of $2.73 by 2030. Assumes 6.2% annual inflation over 5 years, resulting in a 35.1% total price increase due to inflation.

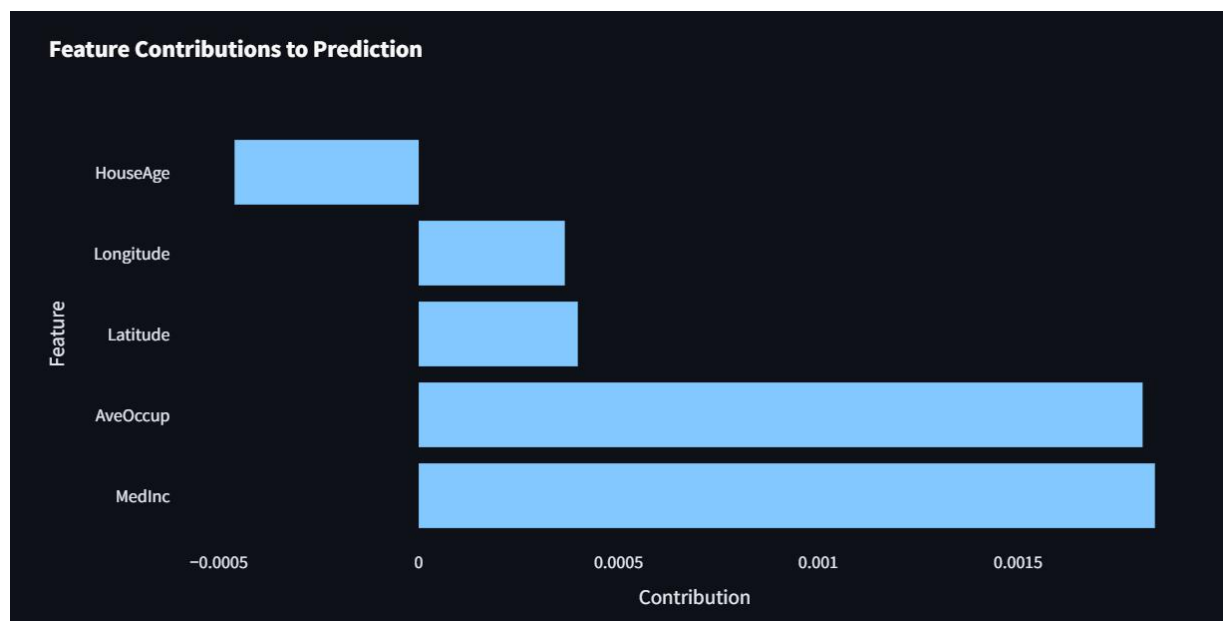### 4.3.8: Feature Contributions to Price Prediction

Figure 4.8: Feature Contribution to Price Prediction

Feature contributions chart showing MedInc and AveOccup have the strongest positive influence on predicted house prices. Latitude and Longitude have moderate positive impacts, while HouseAge appears to have minimal negative contribution to the final prediction.

Chapter 5

Standards Adopted

5.1 Design Standards ………….

In the machine learning domain of house price prediction, predefined design standards include:

1. Follow IEEE and ISO for data science machine learning project design.
   - ✓ ISO 9001 establishes quality management system standards for consistent organizational performance improvement.
   - ✓ IEEE Std 1540 provides systematic approach to identifying and managing software project risks.
   - ✓ ISO/IEC 25012 defines data quality characteristics for ensuring reliable and valuable information.
   - ✓ IEEE Std 1012 specifies software verification, validation, and testing life-cycle processes comprehensively.

2. Implement a modular machine learning pipeline with distinct stages:
   - ✓ Data Collection
   - ✓ Pre-Processing
   - ✓ Feature Engineering
   - ✓ Model Training
   - ✓ Evaluation
   - ✓ Deployment

3. Use consistent data split ratios:
   - ✓ 70-80% for training
   - ✓ 10-15% for validation
   - ✓ 10-15% for testing

4. Use and implement recommended architectural patterns for regression models.
   - ✓ Linear Regression
   - ✓ Random Forest
   - ✓ Gradient Boosting

5. Ensure scalability and reproducibility of the machine learning pipeline.

5.2 Coding Standards ………….

Few of the coding standards kept in mind while building the House Price Prediction model are:

1. Write clean, PEP 8 compliant Python code.
   - ✓ Use 4 spaces per indentation level.
   - ✓ Limit all lines to a maximum of 79 characters.
   - ✓ Surround top-level function and class definitions with two blank lines.
   - ✓ Imports should usually be on separate lines.
   - ✓ Code in the core Python distribution should always use UTF-8, and should not have an encoding declaration.

2. Use appropriate and descriptive naming conventions.
   - ✓ Class names should normally use the CapWords convention.
   - ✓ Function names should be lowercase, with words separated by underscores as necessary to improve readability.
   - ✓ Variable names follow the same convention as function names.

- `X_train` : Training feature dataset
- `y_train` : Training target variable

3. The code is organized into functional modules, each fulfilling a specific role. Don't use lengthy functions. This modular structure promotes re-usability, simplifies debugging, and allows for scalability.
   - ✓ preprocessing.py : Responsible for data cleaning and transformation.
   - ✓ feature_engineering.py : Extracts and selects relevant details.
   - ✓ model_training.py : Performs training and evaluation of machine learning models.
   - ✓ visualizations.py : Creates graphical representations of data insights.

4. Segment code into logical paragraphs with clear comments.
5. Inline comments are used to explain complex logic.
6. Docstrings are provided for all functions, detailing parameters and return values.

5.3 Testing Standards ………….

Testing Standards for the House Price Prediction model include:

1. Follow ISO and IEEE standards for quality assurance.
   - ✓ ISO 9001:2015 establishes systematic quality management framework for organizational performance improvement.
   - ✓ ISO/IEC 25010 defines comprehensive software quality characteristics for system and product evaluation.
   - ✓ IEEE Std 730 specifies software quality assurance planning and management processes comprehensively.
   - ✓ IEEE Std 1028 provides comprehensive guidelines for software reviews, audits, and verification processes.

2. Implement comprehensive unit tests for data pre-processing functions.
   - ✓ Each component is tested individually utilizing pytest.

3. Create integration tests for the entire machine learning pipeline.
   - ✓ This verifies the seamless interaction between system modules, confirming proper flow of data across different components.

4. Perform cross-validation to ensure model robustness.
   - ✓ To improve model resilience, k-fold cross-validation (k=5) is utilized.

5. Conduct statistical tests to validate model performance. Several metrics are used to assess model's performance.
   - ✓ Root Mean Squared Error(RMSE)
   - ✓ Mean Absolute Error(MAE)
   - ✓ R^2 Score

Chapter 6

Conclusion and Future Scope

6.1 Conclusion ………….

This project successfully uses a combination of machine learning algorithms to implement an inflation-adjusted house price prediction model. The system is intended to offer precise and up-to-date price forecasts while taking inflationary trends into account. Among the project's principal achievements are:

> **Robust Data Preprocessing:** Encoding techniques, feature scaling, outlier detection, and handling missing values guarantee high-quality data for model training.
> **Feature Engineering & Selection**: Using feature importance ranking techniques, polynomial transformations, and interaction feature generation to improve model performance.
> **Machine Learning Forecasting**: Using regression models like Random Forest and Gradient Boosting to forecast prices.
> **Inflation Adjustment**: The process of normalizing historical home prices and estimating future prices in real terms involves applying the Consumer Price Index (CPI) inflation correction.
> **Web Application for User Interaction**: The goal is to create an interactive Streamlit dashboard that allows users to enter property details and receive real-time price predictions that are adjusted for inflation, along with graphical insights.

Comparison with Traditional Models

This model offers significant advantages over traditional house price prediction models, which often rely on simple regression techniques or basic historical trends. The key differences include:

Feature	Traditional Models	Proposed Model
Inflation Adjustment	Not considered, leading to outdated valuations	CPI-based adjustment ensures price predictions remain relevant
Feature Engineering	Basic features like size, location, and bedrooms	Advanced interactions, polynomial transformations, and feature selection
Modeling Approach	Linear Regression, Decision Trees	ARIMA for time-series forecasting, LSTM for deep learning-based predictions
Market Trends	Uses historical prices without macroeconomic indicators	Incorporates inflation, economic trends, and demand factors
Prediction Interface	Limited usability, often static models	Interactive Streamlit-based interface with visualization and real-time predictions

Overall, the proposed model provides a **dynamic, inflation-aware, and market-adaptive** approach, making it more reliable for **long-term investment and pricing decisions** in the real estate sector.

6.2 Future Scope ………….

The scope of this project can be expanded further to enhance **prediction accuracy, user experience, and practical applicability**. The following enhancements can be considered:

1. **Integration of More Advanced Deep Learning Models**
   a) Implementing Transformers-based models for better sequential data understanding.
   b) Utilizing CNN-LSTM hybrid architectures for feature extraction and sequential learning.
2. **Incorporating More Macroeconomic Indicators**
   a) To increase predictive power, include variables like interest rates, GDP growth, employment rates, and regional economic conditions.
   b) Incorporating real estate market indices for more dynamic forecasting.
3. **Geospatial and Location-Based Pricing Enhancements**
   a) Incorporating socioeconomic factors, property locations, and neighboring amenities using geospatial analysis.
   b) Applying Geographic Information System (GIS)-based models to forecast regional pricing trends.
4. **Automated Data Pipeline for Real-Time Updates**
   a) Establishing a connection with real estate APIs (like Redfin and Zillow) to obtain real-time market data.
   b) Utilizing government inflation datasets and real-time economic reports to automate inflation adjustments.
5. **Enhanced User Experience in Web Application**
   a) Enhancing UI/UX design to improve prediction interactivity and visualization.
   b) Putting into practice tailored suggestions for homebuyers and investors based on past searches.
6. **Multi-Scenario Forecasting and Risk Analysis**
   a) Creating a Monte Carlo simulation module that will give price forecasts confidence intervals.
   b) Incorporating risk assessment models to predict future declines in trends in home prices.
7. **Scaling and Cloud Deployment**
   a) To manage bigger datasets, the model should be deployed on cloud platforms such as AWS, Google Cloud, and Azure.
   b) Employing containerization (Kubernetes, Docker) to enable multi-user access and scalable model deployment.

How these Enhancements Improve Market Competitiveness

The house price prediction system can stay up to date with the most recent developments in economic forecasting and predictive modeling by putting these future improvements into practice. These enhancements will:

➢ **Improve Prediction Accuracy**: More data sources and deep learning methods will yield more accurate valuations by lowering errors.
➢ **Enhance Real-Time Decision Making**: By integrating live data, forecasts will be more accurate in light of the state of the market.
➢ **Offer a More Scalable and User-Friendly System**: Real estate professionals around the world will have easier access to the system thanks to cloud-based deployment.
➢ **Increase Adoption in Financial and Investment Sectors**: The model will be useful for real estate investors, financial analysts, and policymakers by combining risk assessment and multi-scenario forecasting.

This project lays the groundwork for creating an intelligent, data-driven, inflation-aware real estate pricing system, creating opportunities for additional study and advancement in property valuation techniques.

References:-

[1] Dataquest.(2022, July 18). Build Your First Machine Learning Project [Video]. YouTube. https://www.youtube.com/watch?v=Hr06nSA-qww

[2] Codebasics. (2020, Jan 1). Real Estate Price Prediction [Video]. YouTube. https://www.youtube.com/playlist?list=PLeo1K3hjS3uu7clOTtwsp94PcHbzqpAdg

[3] Park, B., & Bae, J. K. (2015). Using machine learning algorithms for housing price prediction: The case of Fairfax County, Virginia housing data. Expert Systems with Applications, 42(6), 2928-2934

[4] Pow, N., Janulewicz, E., & Liu, L. (2014). Applied machine learning project 4 prediction of real estate property prices in Montreal. Course Project Reports, 1-24.

[5] Appic Softwares. (2025, March 21). Develop Real Estate Price Prediction Machine Learning Model. https://appicsoftwares.com/blog/develop-real-estate-price-prediction-machine-learning-model/

[6] GoPenAI. (2024, December 5). Machine Learning Project: Real Estate Price Prediction. https://blog.gopenai.com/machine-learning-project-real-estate-price-prediction-f3e6fd84f9b4

[7] Google for Developers. (2023, Jan 10). Machine learning systems primer: How to train ML models [Video]. YouTube. https://www.youtube.com/watch?v=fMTr7-ultb8

[8] Kaggle Dataset. (2022, Jan 12). Housing Prices Dataset. https://www.kaggle.com/datasets/yasserh/housing-prices-dataset

[9] freeCodeCamp. (2024, Feb 22). How to build a House Price Prediction Model. https://www.freecodecamp.org/news/how-to-build-a-house-price-prediction-model/

[10] Scribd. (2020, May 18). How to prepare a Project Report. https://www.scribd.com/document/88922638/How-to-Prepare-a-Project-Report

[11] ScienceDirect. (2020, Jun 11). Housing Price Prediction via Improved Machine Learning Techniques. https://www.sciencedirect.com/science/article/pii/S1877050920316318

INDIVIDUAL CONTRIBUTION REPORT:

HOUSE PRICE PREDICTION

ABIR BANERJEE
2205175

Abstract: This project aims to develop a machine learning-based model to predict house prices accurately using various property-related features. By applying data preprocessing, feature engineering, and model training techniques, the system analyzes real estate data to estimate housing values. The objective is to assist buyers, sellers, and agents in making informed decisions. A user-friendly interface is provided through a Streamlit app, enabling real-time predictions based on user inputs.

Individual contribution and findings: As a member of the team, I was primarily responsible for data preprocessing and collaborated in model training. My work started with understanding the structure and quality of the dataset, identifying missing values, and determining the appropriate imputation techniques. I implemented One-Hot Encoding and Label Encoding to convert categorical variables and ensured numerical features were scaled appropriately using normalization methods. Through this project, I gained practical experience in real-world data preprocessing and model development. Overall, the project deepened my understanding of the end-to-end machine learning workflow and enhanced my collaborative and problem-solving skills.

Individual contribution to project report preparation: I was responsible for writing the Requirement Specifications of the group project report. I also collaborated to write the Literature Review of the report.

Individual contribution for project presentation and demonstration: For the project presentation, I presented the data preprocessing part of the model.

Full Signature of Supervisor: Full signature of the student:

…………………………….   …………………………..

INDIVIDUAL CONTRIBUTION REPORT:

HOUSE PRICE PREDICTION

ANKIT BAKSHI
2205183

Abstract: This project aims to develop a machine learning-based model to predict house prices accurately using various property-related features. By applying data preprocessing, feature engineering, and model training techniques, the system analyzes real estate data to estimate housing values. The objective is to assist buyers, sellers, and agents in making informed decisions. A user-friendly interface is provided through a Streamlit app, enabling real-time predictions based on user inputs.

Individual contribution and findings: As a member of the team, I was primarily responsible for the development of the Streamlit deployment interface. I independently built the **Streamlit web application** that integrates the trained model with an interactive UI, enabling users to input property details and receive instant price predictions. This involved coding the app layout, setting up input widgets, and ensuring smooth model integration with user inputs. Through this project, I gained practical experience in front-end deployment.I became proficient with Streamlit, which I found to be an intuitive and efficient tool for creating ML dashboards. Overall, the project deepened my understanding of the end-to-end machine learning workflow and enhanced my collaborative and problem-solving skills.

Individual contribution to project report preparation: I collaborated to write the Literature Review and Implementation sections of the project report. In addition, I was responsible for proofreading, editing, and compiling the final version of the report to ensure clarity, consistency, and completeness before submission.

Individual contribution for project presentation and demonstration: For the project presentation, I collaborated in designing the PowerPoint slides, ensuring that they are visually appealing. I am responsible for demonstrating the working of Streamlit based web application.

Full Signature of Supervisor: Full signature of the student:

…………………………….   …………………………..

INDIVIDUAL CONTRIBUTION REPORT:

HOUSE PRICE PREDICTION

ASHUTOSH KUMAR SINGH
2205193

Abstract: This project aims to develop a machine learning-based model to predict house prices accurately using various property-related features. By applying data preprocessing, feature engineering, and model training techniques, the system analyzes real estate data to estimate housing values. The objective is to assist buyers, sellers, and agents in making informed decisions. A user-friendly interface is provided through a Streamlit app, enabling real-time predictions based on user inputs.

Individual contribution and findings: As a member of the team, I was primarily responsible for feature engineering and collaborated in model training. For feature engineering, I contributed to outlier detection and removal, applying statistical techniques like IQR to eliminate data noise. In the model development phase, I explored different algorithms such as Linear Regression, Random Forest, and Gradient Boosting (XGBoost), helping to tune their hyperparameters using Grid Search. I performed comparative analysis using evaluation metrics like MAE, MSE, RMSE, and R^2. Overall, the project deepened my understanding of the end-to-end machine learning workflow and enhanced my collaborative and problem-solving skills.

Individual contribution to project report preparation: I was responsible for writing the Introduction and Standards Adopted of the group project report.

Individual contribution for project presentation and demonstration: For the project presentation, I presented the feature engineering part of the model.

Full Signature of Supervisor: Full signature of the student:

………………………….   …………………………..

INDIVIDUAL CONTRIBUTION REPORT:

HOUSE PRICE PREDICTION

ADITYA KUMAR
2205349

Abstract: This project aims to develop a machine learning-based model to predict house prices accurately using various property-related features. By applying data preprocessing, feature engineering, and model training techniques, the system analyzes real estate data to estimate housing values. The objective is to assist buyers, sellers, and agents in making informed decisions. A user-friendly interface is provided through a Streamlit app, enabling real-time predictions based on user inputs.

Individual contribution and findings: As a member of the team, my individual contribution was **creating model visualizations**, including feature importance plots, error distribution charts, and performance comparisons across algorithms. These visual insights were crucial for understanding how the model behaves and for evaluating its effectiveness. Building visualizations improved my skills in data interpretation and made it easier to communicate model behavior.Overall, the project deepened my understanding of the end-to-end machine learning workflow and enhanced my collaborative and problem-solving skills.

Individual contribution to project report preparation: I was responsible for writing the Conclusion and Future Scope of the group project report. I also collaborated to write the Implementation section of the report.

Individual contribution for project presentation and demonstration: For the project presentation, I presented the visualization part of the model. I also collaborated in designing the structure and flow of slides.

Full Signature of Supervisor: Full signature of the student:

……………………….   …………………………

TURNITIN PLAGIARISM REPORT

"HOUSE PRICE PREDICTION"

ORIGINALITY REPORT

19%	14%	8%	15%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	www.coursehero.com Internet Source	3%
2	www.goneboarding.co.uk Internet Source	2%
3	www.worldleadershipacademy.live Internet Source	2%
4	Submitted to Banaras Hindu University Student Paper	1%
5	Submitted to Harrisburg University of Science and Technology Student Paper	1%
6	Submitted to The Manchester College Student Paper	1%
7	Submitted to University of Westminster Student Paper	1%
8	medium.com Internet Source	1%
9	www.bhutanpythoncoders.com Internet Source	1%
10	Submitted to Manipal University Student Paper	1%

| 11 | Submitted to European University
Student Paper | <1% |

| 12 | R. N. V. Jagan Mohan, B. H. V. S. Rama
Krishnam Raju, V. Chandra Sekhar, T. V. K. P.
Prasad. "Algorithms in Advanced Artificial
Intelligence - Proceedings of International
Conference on Algorithms in Advanced
Artificial Intelligence (ICAAAI-2024)", CRC
Press, 2025
Publication | <1% |

| 13 | Submitted to Hofstra University
Student Paper | <1% |

| 14 | Submitted to School of Business and
Management ITB
Student Paper | <1% |

| 15 | Submitted to KIIT University
Student Paper | <1% |

| 16 | Submitted to Rochester Institute of
Technology
Student Paper | <1% |

| 17 | Submitted to University of West London
Student Paper | <1% |

| 18 | link.springer.com
Internet Source | <1% |

| 19 | socialcare.wales
Internet Source | <1% |

| 20 | innovationyourself.com
Internet Source | <1% |

21 www.numpyninja.com
Internet Source
<1%

22 www.rkimball.com
Internet Source
<1%

23 datawisdomx.com
Internet Source
<1%

24 fdocuments.in
Internet Source
<1%

25 mafiadoc.com
Internet Source
<1%

26 dspace.uvic.cat
Internet Source
<1%

27 www.irjet.net
Internet Source
<1%

28 Aaditya Jain, Waleed M. Ead, Mohammed Alshahrani, Jawaher Suliman Altamimi, Fawaz Aklan, Shaza Dawood Rihan. "Designing a hybrid neural network framework for real-time anomaly detection in cybersecurity applications", International Journal of Information Technology, 2025
Publication
<1%

29 M. Honicky, S. M. Cardoso, L. R. A. Lima, S. G. I. Ozcariz, F. G. K. Vieira, I. Carlos Back, Y. M. F. Moreno. "Added sugar and trans fatty acid intake and sedentary behavior were associated with excess total-body and central adiposity in children and adolescents with

congenital heart disease", Pediatric Obesity, 2020
Publication
<1%

30 open-innovation-projects.org
Internet Source
<1%

31 www.springerprofessional.de
Internet Source
<1%