Forecasting House Prices accurately Using Smart Regression

Techniques in Data Science

**Student Name:** S.Abinaya

**Register Number:** 422223243003

**Institution:** Surya group of Institutions

**Department:** B.Tech Artificial Intelligence and Data Science

**Date of Submission:** 14-05-2025

**Github Repository Link:** [**https://github.com/abinaya-209/abinaya-phase-3.git**](https://github.com/abinaya-209/abinaya-phase-3.git)

# Problem Statement

The objective is to accurately predict house prices based on various features like location, size, number of rooms, etc., using advanced regression techniques. This is a regression problem with real-world business value in real estate, helping buyers, sellers, and agencies make data-driven decisions.

# Abstract

This project aims to forecast house prices by applying smart regression techniques in data science. By using a rich dataset with housing attributes, we preprocess, analyze, and model the data using machine learning algorithms. Key techniques include feature engineering, model tuning, and performance evaluation. Models such as Linear Regression, Random Forest, and XGBoost are compared. The final deployed solution provides real-time predictions for user-input house features.

# System Requirements

**Hardware:** 8 GB RAM, Intel i5 processor or better

**Software:** Python 3.8+, Libraries - pandas, numpy, matplotlib, seaborn, scikit-learn, xgboost, streamlit **IDE:** Jupyter Notebook or Google Colab

# Objectives

The main objective of forecasting house prices using smart regression techniques is to develop an accurate and reliable

predictive model that can estimate property values based on various influencing factors. This involves utilizing advanced

data science methods to uncover patterns and relationships within housing data, enabling more informed decision-making

for buyers, sellers, and real estate professionals. Key goals include identifying the most relevant features that impact price,

applying effective regression algorithms to capture both linear and nonlinear trends, and ensuring the model is interpretable,

scalable, and performs well on unseen data. Ultimately, the aim is to provide a data-driven tool that enhances pricing

accuracy and supports strategic planning in the real estate market..

# Flowchart of Project Workflow

DATA COLLECTION

PREPROCESSING

EDA

FEATURE ENGINEERING

MODEL TRAINING

EVALUATION

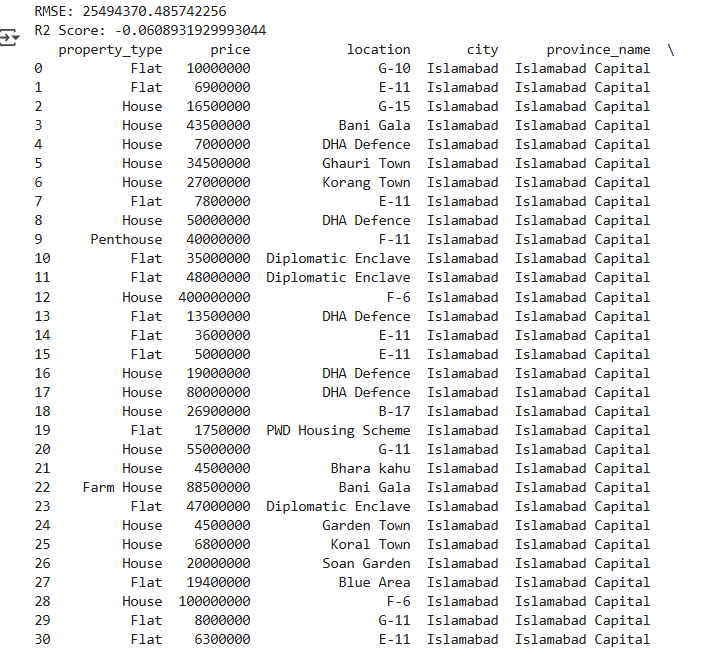
VISUALIZATION

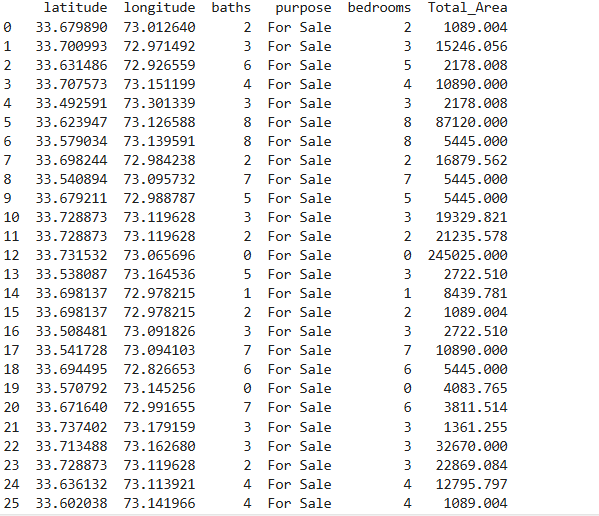
REPORTING

# Dataset Description

**Source:** Kaggle (e.g., Ames Housing Dataset) **Type:** Public

**Size:** ~2900 rows, ~80 columns Include a df.head() screenshot





# Data Preprocessing

Data processing for house price forecasting is a crucial step that ensures the dataset is clean, consistent, and suitable

for modeling. It begins with data collection, where historical housing data is gathered from reliable sources. The next step

is data cleaning, which involves handling missing values, correcting inconsistencies, and removing duplicates or irrelevant

records. Outliers are also detected and treated, as they can skew model performance. Categorical variables such as

location or property type are encoded into numerical formats using techniques like one-hot encoding or label encoding.

Numerical features are standardized or normalized to ensure uniformity and improve model efficiency. Additionally,

feature selection or dimensionality reduction techniques may be applied to retain only the most informative variables.

These preprocessing steps prepare the data for accurate and effective model building.

# Exploratory Data Analysis (EDA)

Exploratory Data Analysis (EDA) for house price forecasting begins with visualizing the distribution of the target

variable—house prices—to understand its spread, skewness, and potential anomalies. Next, correlation analysis

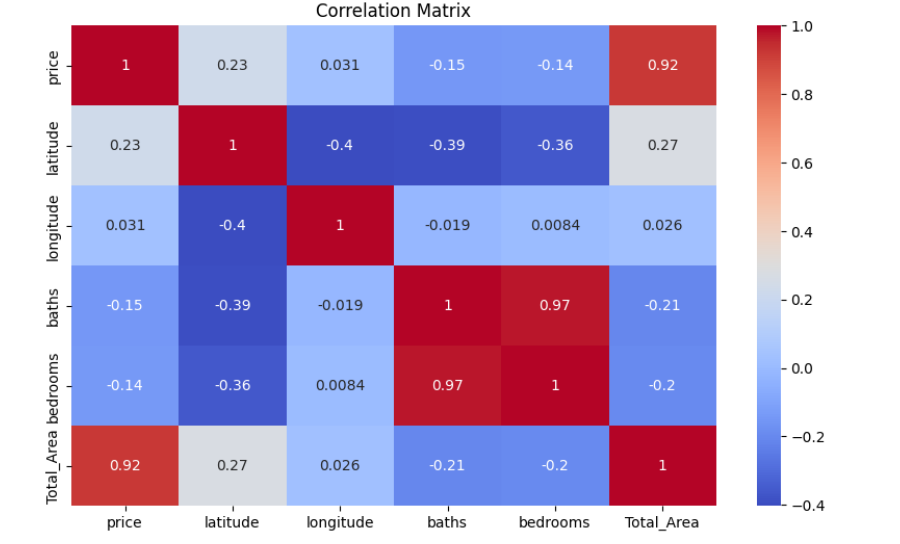
performed using heatmaps to identify features that have strong linear relationships with the target, helping prioritize

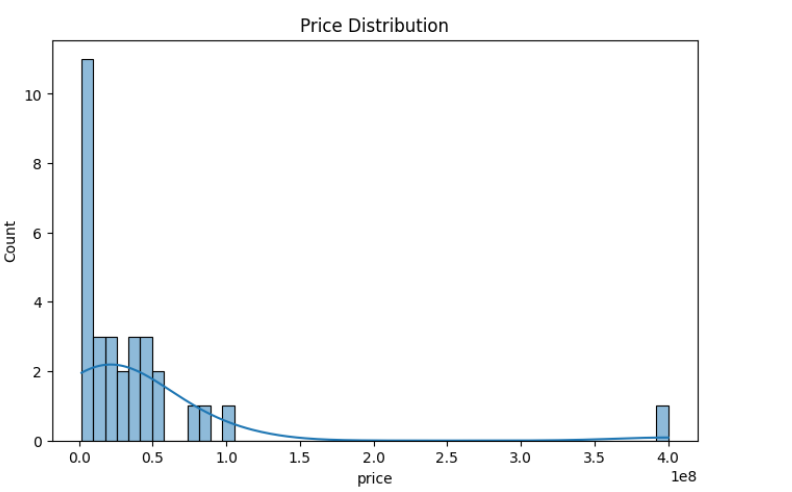
important variables. The distributions of both numerical and categorical features are then examined to uncover trends,

patterns, or irregularities within the data. Outlier detection follows, as extreme values in features like square footage or

price can significantly affect model accuracy. Lastly, missing data is analyzed to understand its extent and patterns,

guiding decisions on whether to impute, drop, or otherwise handle incomplete records.





# Feature Engineering

Feature engineering in house price forecasting involves creating, transforming, and selecting variables that

enhance the model’s ability to make accurate predictions. It starts with deriving new features such as the age of the

house (from the construction year), price per square foot, or indicators for recent renovations, which can provide deeper

insights into property value. Categorical features like location, building type, or neighborhood quality are converted into

numerical representations using encoding techniques. Interaction features, such as combining the number of bedrooms

with the size of the property, can capture complex relationships. Temporal features like the year of sale can also be

transformed to reveal market trends over time. Effective feature engineering ensures that the model captures the most

relevant information, ultimately improving prediction accuracy and model interpretability.

# Model Building

Data building for house price forecasting refers to the process of preparing and structuring the dataset to be

suitable for training machine learning models. It begins with collecting relevant data from sources such as real estate

listings, government records, or property databases, ensuring the inclusion of important variables like location, size

number of rooms, and year built. After gathering the data, it is organized into a structured format, typically a tabular

dataset where each row represents a house and each column represents a feature. This stage also includes integrating

multiple data sources if necessary and ensuring consistency across them. Once the data is compiled, it undergoes

preprocessing steps such as cleaning, encoding, and scaling. The final structured dataset is then ready to be used in the

model building phase for training regression models to forecast house prices accurately.

# Model Evaluation

Data modeling for house price forecasting involves selecting, training, and evaluating machine learning algorithms

to predict property values based on input features. The process starts by splitting the prepared dataset into training and

testing sets to ensure unbiased evaluation. Various regression models are then applied, such as Linear Regression,

Decision Trees, Random Forest, Gradient Boosting, and XGBoost, each chosen based on the complexity and nature of

the data. These models learn patterns and relationships between features like location, size, and condition of the house

and the target variable—price. During training, hyperparameter tuning and cross-validation are used to optimize

performance and prevent overfitting. Once trained, the models are evaluated using metrics like Mean Absolute Error

(MAE), Root Mean Squared Error (RMSE), and R-squared to determine accuracy.

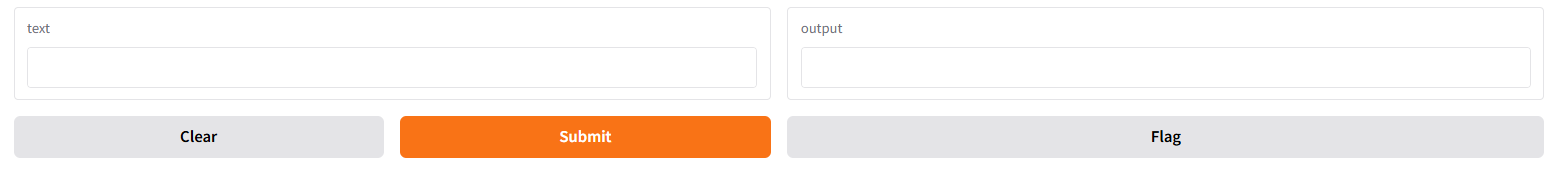
# Model Evaluation

* 1. Metrics: RMSE, R2 Score
  2. Visuals: Residual plot, Prediction vs Actual
  3. Best model: XGBoost with RMSE = XX
  4. Include evaluation metric tables & plots

# Deployment

* 1. Platform: Streamlit Cloud
  2. Public Link: [Insert your URL]
  3. Features: User inputs house data, real-time price prediction
  4. Include UI screenshot and sample output

Link: <https://68f91a4de0bcc81f2a.gradio.live/>



# Source Code

GitHub Repository: <https://github.com/abinaya-209/abinaya-phase-3.git>

# Future Scope

* 1. Integrate live real estate data via APIs
  2. Enhance model with image and text-based features
  3. Extend to rental price prediction

# Team Members and Roles

* 1. **Data Cleaning, EDA, Feature Engineering:** H. Aashra Begam
  2. **Model Building :** S. Abinaya
  3. **Evalution:** M. Heefa Sharin
     1. **Deployment and Documentation :** KS.Archana

