Phase-2

Student Name: ABINAYA S

Register Number: 510823205001

Institution: Ganadipathy Tulsi's Jain Engineering College

Department: Information Technology

Date of Submission: 08/05/2025

Github Repository Link: https://github.com/abzya26/House-price.git

---

1. Problem Statement

Accurately predicting house prices based on various property features is a key challenge in real estate analytics. This is a regression problem aiming to estimate continuous output (house price) using historical data. Solving this helps realtors, buyers, and policymakers make informed decisions, optimize investments, and understand market dynamics.

2. Project Objectives

- Build and evaluate multiple regression models to predict house prices.

- Compare performance metrics (MAE, RMSE, R^2) to select the best model.
- Ensure model interpretability and practical applicability.
- Improve prediction accuracy via feature engineering and model tuning.

3. Flowchart of the Project Workflow

4. Data Description

- **Source:** Kaggle's House Price Dataset.

- **Type:** Structured tabular data.
- **Features:** e.g., 80 variables (size, location, amenities).
- **Target:** SalePrice (continuous variable).
- **Dataset Type:** Static.

5. Data Preprocessing

- Handled missing values using median/mode imputation.
- Removed duplicates and irrelevant features.
- Outlier treatment via IQR method or log transformation.
- Encoded categorical variables using one-hot encoding.
- Normalized numerical features using MinMaxScaler/StandardScaler.

6. Exploratory Data Analysis (EDA)

- **Univariate:** Histograms and boxplots for numerical features.
- **Bivariate:** Heatmap of correlations to identify impactful predictors.

- **Insights:** Features like OverallQual, GrLivArea, and GarageCars show strong correlation with price.

7. Feature Engineering

- Created interaction terms (e.g., TotalBathrooms).
- Extracted date components from year-related features.
- Binned skewed variables to reduce variance.
- PCA for dimensionality reduction (optional).

8. Model Building

- **Models Used:** Linear Regression, Random Forest Regressor, XGBoost Regressor.
- **Train-Test Split:** 80-20 ratio with stratified sampling if needed.
- **Metrics:** MAE, RMSE, and R^2 score.
- **Best Model:** XGBoost with hyperparameter tuning showed lowest RMSE.

9. Visualization of Results & Model Insights

- Residual plots to check error distribution.

- Feature importance plot (from tree-based models).
- Comparison charts for MAE/RMSE across models.

10. Tools and Technologies Used

- **Language:** Python
- **IDE:** Jupyter Notebook / Google Colab
- **Libraries:** pandas, numpy, matplotlib, seaborn, scikit-learn, XGBoost

11. Team Members and Contributions

- **Abinaya S**– Data collection, Development.
- **Gokulavarshini P**-Model evaluation.
- **Madhan S**- Visualization, project co-ordination.
- **Jeevan R**-Documentation and Reporting,