## 

## **Phase-2**

**Student Name:** KIRUTHIKA.S

**Register Number:** 613023243029

**Institution:** Vivekanandha College Of Technology for women

**Department:**  B-Tech-AI&DS

**Date of Submission:** 03.05.2025

**GitHub Repository Link:**

### **1. Problem Statement**

The real estate market is characterized by its dynamic nature, with property prices

influenced by a multitude of factors such as location, market trends, economic

indicators, and property-specific attributes. Accurately predicting house prices

remains a critical challenge for stakeholders, including buyers, sellers, and

investors. Traditional forecasting methods often struggle to capture the complex,

non-linear relationships among these factors, leading to suboptimal predictions.

This project aims to leverage advanced regression techniques in data science, such

as multiple linear regression, ridge regression, LASSO, and ensemble methods, to

develop a robust and accurate model for house price forecasting. The objective is

to provide actionable insights and improve decision-making in the real estate

industry by employing intelligent algorithms and feature engineering to enhance

prediction accuracy.

**2. Project Objectives**

The primary goal is to accurately forecast house prices using advanced regression techniques in data science. This involves:

* Identifying key factors influencing house prices.
* Evaluating different regression models (e.g., linear regression, random forest, XGBoost, etc.).
* Improving prediction accuracy using feature engineering and hyperparameter tuning.
* Delivering actionable insights, trends, and recommendations based on data.
* Assessing whether initial objectives evolved after deeper data exploration.

**3. Flowchart of the Project Workflow**

### 

### **4. Data Description**

* Dataset Name & Source: The dataset used for house price prediction is often sourced from Kaggle competitions, such as the ["House Prices - Advanced Regression Techniques"]
* Data Type: Structured dataset containing numerical and categorical attributes.
* Number of Rows & Columns: Typically, datasets contain thousands of rows (e.g., 1,460 observations in the Kaggle dataset) and 80+ columns, covering various house features
* Static or Dynamic Dataset: Most datasets used for house price prediction are static meaning they do not update in real-time. However, dynamic datasets can be created by integrating live market data.
* Key Fields & Attributes:  
    Location-based attributes: Neighborhood, zip code  
    Structural features: Square footage, number of bedrooms/bathrooms, garage size  
    Quality indicators: Year built, renovation history, material quality  
    Market-related attributes: Sale price, tax assessment, inflation-adjusted values

**5. Data Preprocessing**

Steps include:

* Handling missing values using imputation techniques.
* Removing duplicate records.
* Formatting & parsing data (e.g., date conversions, numeric scaling).
* Encoding categorical variables (e.g., one-hot encoding, label encoding).
* Identifying & treating outliers using statistical or visualization methods.

**6. Exploratory Data Analysis (EDA)**

* Univariate Analysis: Histograms, boxplots to check distributions.
* Bivariate/Multivariate Analysis: Heatmaps to identify correlations, scatter plots, and grouped bar charts.
* Key Metrics Analysis: Average price trends, price variations based on features like location, house age, etc.
* Summary of Insights: Identifying patterns and relationships between variables.

**7. Feature Engineering**

* Handle Missing Data → Impute values using mean/median.
* Encode Categorical Variables → One-hot encoding for location, property type.
* Scale & Normalize → Min-Max Scaling for numerical features.
* Create New Features → Price per square foot, age of property, proximity score.
* Transform Data → Log transformation for skewed variables, polynomial features for nonlinear relationships.
* Remove Redundant Features → Drop highly correlated variables using correlation analysis.
* Feature Selection → Use RFE, SHAP values, or model-based importance metrics.

**8. Building Model**

1. Data Preparation

* Load and preprocess data (handle missing values, encode categorical features).
* Feature engineering (create new features, scale numerical values).

2. Splitting Data

* Divide dataset into training (80%) and testing (20%) sets.

3. Selecting Regression Models

* Linear Regression → Simple but interpretable.
* Random Forest → Handles non-linearity and feature importance.
* XGBoost → Advanced boosting method for accuracy.
* Polynomial Regression → Captures complex relationships.

4. Model Training & Evaluation

* models using the scikit-learn library in Python.
* Evaluate with Mean Absolute Error (MAE), R-Squared (R²), and Root Mean Train Square Error (RMSE)

5. Hyperparameter Tuning

* Use GridSearchCV or RandomizedSearchCV to optimize model parameters.

6. Prediction & Reporting

* Generate predictions for test data.
* Visualize results with Matplotlib & Seaborn (scatter plots, residual plots).

**9. Visualization Of Results & Model Insights**

1. Price Distribution (Histogram)

import matplotlib.pyplot as plt

import seaborn as sns

sns.histplot(data['price'], bins=30, kde=True)

plt.title("House Price Distribution")

plt.xlabel("Price")

plt.ylabel("Frequency")

plt.show()

1. Feature Importance

import pandas as pd

model = XGBRegressor()   
model.fit(X\_train, y\_train)

importance = pd.Series(model.feature\_importances\_, index=X\_train.columns)   
importance.nlargest(10).plot(kind='barh')   
plt.title("Top 10 Most Influential Features")   
plt.show()  
```

1. Correlation Heatmap

sns.heatmap(data.corr(), annot=True, cmap="coolwarm")   
plt.title("Feature Correlation Heatmap")   
plt.show()  
```

1. Actual vs Predicted Prices (Scatter Plot)

plt.scatter(y\_test, y\_pred, alpha=0.5)   
plt.xlabel("Actual Prices")   
plt.ylabel("Predicted Prices")   
plt.title("Actual vs Predicted House Prices")   
plt.show()

1. Residual Plot

sns.residplot(x=y\_test, y=y\_pred, lowess=True)   
plt.title("Residual Plot")   
plt.show()

**10. Tools and Technologies Used**

* Programming Language: Python
* Notebook/IDE: Google Colab, Jupyter Notebook
* Libraries: pandas, numpy, matplotlib, seaborn, plotly
* Optional Automation Tools: pandas-profiling for quick EDA

**11. Team Members and Contributions**

| **Name** | **Contribution** |
| --- | --- |
| Thasneen S | Problem Statement , Project objectives |
| Kiruthika R | Flowchart of the Project Workflow, Tools And Technology |
| Kiruthika S | Data Description, Feature Engineering |

Kowshika S Data Preprocessing, Building Model

Keerthana P EDA, Visulazation Of Results & Methods Insights