**Phase-2 Submission Template**

**Student Name:** Harikrishna S

**Register Number:** 732123205029

**Institution:** Nandha College of Technology

**Department:** Information Technology

**Date of Submission:** [Insert Date]

**Github Repository Link:** https://github.com/harikrishna2705/Harikrishna-S-EBPL.git

### **1. Problem Statement**

* In today's rapidly evolving real estate markets, accurately forecasting house prices is critical for homeowners, investors, financial institutions, and policy-makers. This project aims to develop a robust predictive model to estimate house prices using advanced regression techniques based on historical housing data. The goal is to uncover the complex relationships between various property features (e.g., location, size, number of rooms, amenities, year built) and their corresponding market prices.

### Problem Type:

This is a regression problem, where the target variable is continuous — specifically, the sale price of a house. Unlike classification, which predicts discrete labels, regression is used to forecast numerical outcomes, making it well-suited for property valuation.

### Why This Problem Matters:

* Financial Planning: Accurate price prediction helps buyers and sellers make informed decisions.
* Banking and Insurance: Lenders rely on valuations for mortgage approvals; insurers assess property value to calculate premiums.
* Urban Development and Policy: Governments and urban planners use housing price trends to plan infrastructure and allocate resources.
* Investment Strategy: Real estate investors use price forecasts to identify undervalued or overvalued properties, optimizing portfolio returns.

### **2. Project Objectives**

### Key Technical Objectives:

1. Build an Accurate Predictive Model:
   * Develop regression models (e.g., Linear Regression, Lasso/Ridge, Random Forest, XG Boost) to predict house prices based on available features.
   * Use cross-validation to ensure model generalization and avoid overfitting.
2. Perform Feature Engineering and Selection:
   * Create new features (e.g., total square footage, age of property, renovation status).
   * Handle missing data, encode categorical variables, and scale numerical features.
   * Apply feature importance metrics to retain only the most relevant predictors.
3. Optimize Model Performance:
   * Tune hyper parameters using grid search or Bayesian optimization.
   * Evaluate models using performance metrics such as RMSE, MAE, and R².
4. Ensure Model Interpretability:
   * Use SHAP (Shapley Additive ex Planations) or permutation importance to explain the influence of features on price predictions.
5. Deployability and Real-World Readiness:
   * Design the solution so it can be integrated into web apps or tools for users like real estate agents or property investors.
   * Prepare a model pipeline that includes preprocessing, training, and prediction stages for easy deployment.

### **3. Flowchart of the Project Workflow**

### House Price Prediction by Machine ...

### **4. Data Description** *.*

* Dataset name and origin is Ames Housing Dataset, sourced from Kaggle.
* Type of data:
  + Structured data — each row represents a single house, and columns capture various attributes (numerical and categorical) of the properties.
* Number of records and features.

 Approximately 2,930 records (rows), each representing a residential property.

 Around 80 features (columns), including:

Numerical features (e.g., Lot Area, Gr Liv Area, Year Built)

Categorical features (e.g., Neighborhood, House Style, Garage Type)

* Static or dynamic dataset.

The dataset is static — it represents a snapshot in time of home sales in Ames, Iowa, and does not update in real time.

* Target variable supervised

The target variable is Sale Price, a continuous numeric value representing the final sale price of each house (in USD).

### **5. Data Preprocessing**

Handle missing values .  
 Upon inspection, several features in the dataset contained missing values. These were treated based on the nature of the variable:

* Categorical features (e.g., PoolQC, FireplaceQu) were imputed with 'None', indicating absence.
* Numerical features (e.g., LotFrontage, GarageYrBlt) were imputed using the median of the respective columns to avoid bias from extreme values.
* Remove or justify duplicate records.
  + Duplicate rows were checked using .duplicated(). Any records found to be exact duplicates were removed to avoid redundant data influencing the model.
* Detect and treat outliers.

Outliers in key numerical columns (such as Gr Liv Area and Sale Price) were detected using the Z-score method. Records with Z-scores greater than 3 were considered outliers and removed, as they could skew model predictions.

* Convert data types and ensure consistency.

Data types were reviewed and corrected:

Features like MS Sub Class, which are numerical but categorical in nature, were converted to the category data type.

This helps with correct encoding and interpretation during modeling.

* Encode categorical variables (label encoding, one-hot encoding).
  + Categorical variables were transformed using One-Hot Encoding, which creates binary columns for each category. This method is suitable for nominal variables and helps in providing numerical input to machine learning algorithms.
* Normalize or standardize features where required.

To ensure all numerical features are on the same scale, especially for models sensitive to feature magnitude:

* + Numerical features were standardized using Standard Scaler, which transforms them to have a mean of 0 and a standard deviation of 1.
  + This ensures fair treatment of all features during model training.
* Document and explain each transformation step clearly in code and markdown.]

After preprocessing:

* All missing values were handled.
* Data types were consistent and appropriate.
* The dataset was free of duplicates and major outliers.
* Categorical variables were encoded.
* Numerical features were scaled.

### 

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

* Univariate Analysis:  
   Histograms were used to visualize the distribution of continuous variables like Gr LivArea, Lot Area, Sale Price, and Year Built.

Boxplots helped identify outliers and spread for variables such as Garage Area, Total BsmtSF, and 1stFlrSF.

Sale Price is right-skewed, indicating the need for log transformation.

Gr Liv Area and Total BsmtSF contain some extreme values (outliers).

Most properties were built after 1950.

* *Bivariate/Multivariate Analysis:*
* Correlation Matrix (Heat map):
  + Revealed strong positive correlations between Sale Price and:
    - Overall Qual (quality of materials)
    - Gr Liv Area (above-ground living area)
    - Total BsmtSF (basement area)
    - Garage Cars and Garage Area

Scatterplots:

* + Showed linear trends between Gr Liv Area, Garage Area, and Sale Price.
  + Helped confirm the log-normal nature of Sale Price.

Boxplots (Grouped):

* + Overall Qual vs Sale Price: Showed a clear increase in median house price with higher quality.
  + Neighborhood vs Sale Price: Indicated strong geographical influence on price variation.

Pair plots:

* + Used on selected features to visualize pairwise relationships and spot multi collinearity.
* *Insights Summary:* Key Influencers of Price:
* Over all Qual Gr Liv Area, Garage Cars, SF, and Year Built are highly influential.
* Neighborhood significantly impacts house value, suggesting location-based price modeling is essential.

Skewed Distributions:

* Sale Price and several numeric features are skewed. Log transformation may improve model performance.

### **7. Feature Engineering***.*

* *Using new feature such as Total SF(total Square Food)/House Age and Remodel*
* Sale Month and Sale Year
* PCA (Principal Component Analysis) was considered to reduce dimensionality after encoding categorical variables.Justify each feature added or removed.]

### 

### **8. Model Building**

### **1. Model Selection**

We selected the following two models:

* Linear Regression: A simple and interpretable model that works well when the relationship between variables is mostly linear.
* Random Forest Regressor: A powerful ensemble method that captures complex patterns and non-linear relationships in the data.
* The dataset was split into two parts:
* 80% training data – used to train the models.
* data – used to evaluate the models' performance.20% testing

Divide data in training and test sets, accuracy, accuracy, recall, F1 score,

MAE, RMSE and R. Evaluate the screen using the score.

### **9. Visualization of Results & Model Insights***.*

* A residual plot shows the difference between actual and predicted values.
* A scatter plot where the x-axis represents the actual house prices, and the y-axis represents the predicted prices
* A bar chart showing the importance of each feature in predicting the house price.
* A bar chart comparing different models based on their performance (e.g., error rate or accuracy).

### **10. Tools and Technologies Used**

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

### **11. Team Members and Contributions**

1. Harikrishna S [Team Lead] – Data Collection and Exploration.

2. Ashik T – Model Developing and Evaluation.

3. Ganesh ram M– Testing.

4. Ahamed faaiz S -Deployin.