**Phase-2 Submission Template**

**Student Name:** HARSHITHA B

**Register Number:** 511923106301

**Institution:** PRIYADARSHINI ENGINEERING COLLEGE

**Department:** ELECTRONICS AND COMMUNICATION ENGINEERING **Date of Submission:** 05/05/2025

**Github Repository Link:**

# 1.Problem Statement

The escalating cost of housing and the complexity of factors influencing property values make it challenging for individuals, investors, and real estate professionals to accurately estimate house prices. This uncertainty can lead to inefficient market transactions, investment risks, and difficulties in property valuation.

**Type of Problem:** This is a **regression** problem. We aim to predict a continuous numerical value, which is the sale price of a house.

**Why Solving This Problem Matters:** Accurate house price prediction has significant real-world impact. It can:

* **Empower homebuyers and sellers:** Providing realistic price expectations for informed decision-making.
* **Aid investors:** Identifying undervalued or overvalued properties for strategic investment.
* **Assist real estate agencies:** Offering data-driven valuation services to clients.
* **Inform urban planning and policy:** Understanding factors driving property values can help in developing effective housing policies.
* **Improve financial risk assessment:** Banks and mortgage lenders can use these predictions for more accurate loan evaluations.

# 2. Project Objectives

Our primary objective is to build a robust and reliable machine learning

model capable of accurately predicting the sale price of houses based on their features.

**Key Technical Objectives:**

* Develop at least two different regression models.
* Achieve a Root Mean Squared Error (RMSE) below a certain threshold (to be determined after initial model evaluation) on the test dataset.
* Identify the key features that significantly influence house prices.
* Ensure the model is reasonably interpretable, allowing us to understand the relationship between features and the predicted price.

**Evolution After Data Exploration:** After exploring the dataset (as we'll discuss in EDA), our objectives might evolve. For instance, we might discover highly skewed features requiring specific transformations or identify strong multicollinearity that necessitates feature selection techniques. We might also adjust our target RMSE based on the inherent variability in the data.

# 3. Flowchart of the Project Workflow

**Data Cleaning**

**EDA**

**Feature Engineering**

**Model Building**

**Evaluation**

**Visualization**

**Recommendations**

**Data Collection**

# 4. Data Description

We will be using the **Ames Housing dataset**, a widely used dataset for regression tasks, originally compiled by Dean De Cock for educational purposes.

* **Dataset Origin:** Kaggle (often found in introductory machine learning competitions and datasets)
* **Type of Data:** Structured, tabular data.
* **Number of Records and Features:** Approximately 1460 records (houses) and over 80 features describing various aspects of the properties (e.g., size, location, quality, age).
* **Static or Dynamic Dataset:** Static, meaning the data represents a snapshot in time.
* **Target Variable:** SalePrice (the price the house sold for).

# 5. Data Preprocessing

This stage involves cleaning and preparing the data for analysis and model building.

* **Handle Missing Values:**
  + We will first identify features with missing values and analyze the pattern of missingness. oFor categorical features with many missing values, we might impute with a new category like 'Missing' or consider dropping the feature if it seems uninformative. oFor numerical features, imputation strategies could include using the mean, median, or more advanced techniques based on feature relationships. We will document the chosen method for each feature.

|  |  |
| --- | --- |
| • | **Remove or Justify Duplicate Records:** We will check for and remove any duplicate rows, ensuring each data point represents a unique house sale. If duplicates are found, we'll investigate the reason before removal. |
| • | **Detect and Treat Outliers:** We will use visual methods (boxplots, scatterplots) and statistical methods (e.g., IQR rule, Z-score) to identify potential outliers in numerical features, especially the target variable. We will then decide whether to remove, cap, or transform these outliers based on their potential impact on the model. |
| • | **Convert Data Types and Ensure Consistency:** We will ensure that the data types of each feature are appropriate (e.g., numerical features are not stored as strings). We'll also check for inconsistencies in categorical values (e.g., different spellings for the same category) and standardize them. |
| • | **Encode Categorical Variables:** Since most machine learning models work with numerical input, we will encode categorical features:  o**Label Encoding:** May be used for ordinal categorical features (e.g., quality ratings like 'Fair', 'Good', 'Excellent'). o**One-Hot Encoding:** Will be applied to nominal categorical features (e.g., neighborhood, house style) to create binary dummy variables for each category. We need to be mindful of the curse of dimensionality if there are many unique categories. |

|  |
| --- |
| * **Normalize or Standardize Features:** For models sensitive to feature scaling (e.g., linear regression with regularization, KNN, neural networks), we will normalize (scaling to a range of [0, 1]) or standardize (scaling to have zero mean and unit variance) the numerical features. The choice will depend on the distribution of the features and the model being used. * **Documentation:** All preprocessing steps will be clearly documented in the code with comments and further explained in markdown cells, justifying the choices made.   **6. Exploratory Data Analysis (EDA)**  This phase aims to understand the data better through statistical and visual exploration.   * **Univariate Analysis:**      * + **Distribution of numerical features:** Histograms and boxplots will reveal the distribution, skewness, and presence of outliers for each numerical feature.   + **Distribution of categorical features:** Countplots will show the frequency of each category within each categorical feature.      * **Bivariate/Multivariate Analysis:**      * + **Correlation matrix:** A heatmap of the correlation matrix will show the linear relationships between numerical features and with the target variable. |

* + **Pairplots:** Scatterplots of all pairs of numerical features can reveal non-linear relationships and potential multicollinearity.
  + **Scatterplots:** We will specifically examine the relationship between key numerical features (e.g., living area, lot size) and the SalePrice.
  + **Grouped bar plots/Boxplots:** We will analyze how the SalePrice varies across different categories of important categorical features (e.g., neighborhood, overall quality).
* **Insights Summary:**

* + We will highlight any observed patterns, trends, and interesting relationships in the data. For example, we might find a strong positive correlation between living area and sale price. oWe will identify features that appear to have a strong influence on the SalePrice based on correlations and visual analysis. For instance, features related to size, quality, and location are likely to be important. We will explain the reasoning behind these observations.

# 7. Feature Engineering

This step involves creating new features or transforming existing ones to potentially improve model performance.

•**Create new features based on domain knowledge or EDA insights:**

oWe might create a 'Total Square Footage' feature by combining basement, first floor, and second floor areas.

|  |  |
| --- | --- |
|  | oAn 'Age of House' feature can be derived from the year built and year sold. oWe could create interaction terms between features that seem to have a combined effect on price (e.g., overall quality \* living area). |
| • | **Combine or split columns:** As mentioned above, combining square footage columns is an example. We might also extract month and year from a date feature if present (though not prominent in this dataset). |
| • | **Use techniques like binning, polynomial features, ratios:**   * We might bin numerical features like lot size into different categories. * Introducing polynomial features (e.g., square of living area) can capture non-linear relationships. * We could calculate ratios like the area of the porch relative to the total living area. |
| • | **Apply dimensionality reduction (optional, e.g., PCA):** If we end up with a very high number of features after one-hot encoding, we might consider using Principal Component Analysis (PCA) to reduce the dimensionality while retaining most of the variance. |
| • | **Justify each feature added or removed:** Every feature  engineering step will be clearly documented with an explanation of |

the rationale behind it, whether it's based on domain knowledge, EDA findings, or an attempt to capture non-linearities or interactions.

# 8. Model Building

We will select and implement at least two different regression models.

* **Selected Models (Justification):**
  + **Linear Regression (with Regularization - Lasso or Ridge):** This is a fundamental linear model and serves as a good baseline. Regularization techniques (L1 or L2) can help prevent overfitting, especially with a potentially large number of features after encoding. It also provides insights into feature importance through coefficient magnitudes.
  + **Random Forest Regressor:** This is a powerful non-linear ensemble method that can capture complex relationships between features and the target variable. It is generally robust to outliers and can provide a measure of feature importance.

* **Split Data:** We will split the dataset into training and testing sets (e.g., 80% for training and 20% for testing) using a random split with a fixed random\_state for reproducibility. Stratification might not be strictly necessary for a purely numerical target variable, but we'll ensure a representative split of the target variable's distribution if needed.

* **Train Models and Evaluate Initial Performance:**

* + We will train each selected model on the training data. oWe will then evaluate their performance on the unseen test data using appropriate regression metrics:

* + - **Mean Absolute Error (MAE):** Average absolute difference between predicted and actual prices.
    - **Root Mean Squared Error (RMSE):** Square root of the average squared difference, penalizes larger errors more. This is a common metric for house price prediction.
    - **R-squared (R²):** Proportion of the variance in the dependent variable that is predictable from the independent variables.

# 9. Visualization of Results & Model Insights

We will use visualizations to understand model behavior and results.

* **Residual Plots:** Scatter plots of the predicted values versus the residuals (actual - predicted). These plots help assess the model's assumptions (e.g., homoscedasticity) and identify any systematic biases.

* **Scatter Plot of Predictions vs. Actual Values:** Plotting the predicted sale prices against the actual sale prices on the test set will visually show how well the model is performing. A perfect model would have all points lying on a straight diagonal line.

* **Feature Importance Plot (for Random Forest):** Bar plots showing the relative importance of each feature as determined by the Random Forest model. This helps identify the most influential factors in predicting house prices.

* **Coefficient Plots (for Linear Regression):** Bar plots showing the coefficients of the features in the linear regression model. The magnitude and sign of the coefficients indicate the strength and direction of the relationship with the sale price.

* **Visual Comparisons of Model Performance:** We might use bar charts to compare the MAE, RMSE, and R² scores of the different models on the test set for easy comparison.

* **Interpretation of Top Features:** We will clearly explain which features the models identify as most important and how they influence the predicted sale price based on the visualizations (e.g., a high positive coefficient for 'Living Area' in linear regression suggests that larger houses tend to have higher sale prices).

# 10. Tools and Technologies Used

* **Programming Language:** Python
* **IDE/Notebook:** Google Colab or Jupyter Notebook
* **Libraries:**
  + pandas for data manipulation and analysis. o numpy for numerical operations.
  + seaborn and matplotlib for data visualization.
  + scikit-learn 1 for machine learning algorithms (linear regression, random forest, train-test split, metrics, preprocessing).



# Thank you