# PROJECT TITLE :**HOUSE PRICE PREDICTOR**

**PHASE 1 :**

* Problem Definition
* Design Thinking

**PROBLEM DEFINITION :**

* The goal of this project is to develop a machine learning model that accurately predicts the selling price of residential properties based on a set of relevant features.
* This predictive model aims to assist homebuyers, sellers, and real estate professionals in estimating property values, facilitating informed decisions, and improving the overall real estate experience.

**DESIGN THINKING :**

1. Data Source :KaggleUSA\_Housing
2. Data Preprocessing :

Preprocessing the data is a crucial step in building a house price predictor model. Proper preprocessing helps clean and prepare the data for machine learning.

Data Cleaning:

* + Handle missing values: Identify and decide how to handle missing data (e.g., imputation, removal, or a separate category).
  + Outlier detection: Detect and potentially remove extreme data points that could adversely affect the model's performance..

Handling Missing Values:

* + Identify columns with missing values. Missing data can affect the performance of your model.
  + Decide how to handle missing values based on the nature of the data and the extent of missingness:
  + Remove rows or columns with a high percentage of missing values if they are not critical.
  + Impute missing values using techniques like mean, median, mode, or more advanced imputation methods (e.g., regression imputation).

Encoding Categorical Variables:

* + Categorical variables need to be encoded into numerical format for machine learning models. Common encoding methods include:
    - * One-Hot Encoding: Create binary columns for each category.
      * Label Encoding: Assign unique integers to each category.
      * Target Encoding: Encode categorical variables based on the mean of the target variable for each category.

1. Feature Selection :

Feature selection is a crucial step in building an effective house price predictor model. It involves choosing the most relevant and informative features while eliminating irrelevant or redundant ones. Feature selection can improve model performance, reduce overfitting, and enhance model interpretability.

Tree-Based Models:

* + Tree-based models like Random Forest or Gradient Boosting provide feature importance scores. Features with higher importance scores are likely more relevant for prediction.
  + You can use these scores to select the top-n features.

1. Model Selection :

The choice of model depends on various factors, including the size and quality of your dataset, the complexity of the problem, and computational resources.

The model we have used here is Random Forest.

Random Forest:

* + Random Forest is an ensemble of decision trees that combines multiple weak learners to create a strong model.
  + It handles non-linear relationships and is less prone to overfitting compared to individual decision trees.

1. Model Training :

Training a house price predictor model using the Random Forest algorithm involves several steps. Random Forest is an ensemble learning method that combines multiple decision trees to make more accurate predictions.

1. Import Libraries:
   * Import the necessary Python libraries, including scikit-learn for machine learning and pandas for data manipulation.
2. Load and Prepare Data:
   * Load the preprocessed house price dataset.
   * Separate the features (independent variables) and the target variable (house prices).
3. Split the Data:
   * Split the data into training and test sets to evaluate the model's performance.
4. Initialize and Train the Random Forest Model:
   * Create an instance of the RandomForestRegressor and set hyperparameters (e.g., the number of trees, max depth).
   * Train the model on the training data.
5. Make Predictions:
   * Use the trained model to make predictions on the test set.
6. Evaluation :

Evaluating a house price predictor model built with the Random Forest algorithm is essential to assess its performance and reliability. Below are the steps to evaluate the model.

* Mean Absolute Error (MAE):
* MAE measures the average absolute difference between the predicted and actual house prices.

mae = mean\_absolute\_error(y\_test, y\_pred)

* Root Mean Squared Error (RMSE):
* RMSE is the square root of MSE and provides a measure of the average magnitude of errors in the same units as the target variable.

rmse = mean\_squared\_error(y\_test, y\_pred, squared=False)

* R-squared (R2) Score:
* R2 measures the proportion of the variance in the target variable that is explained by the model. A higher R2 value indicates a better fit.

r2 = r2\_score(y\_test, y\_pred)