**PREDICTING HOUSE PRICE USING MACHINE LEARNING**

## Phase 3 Submission Document

**Project:** House Price Prediction



## Introduction:

* The real estate market is one of the most dynamic and lucrative sectors, with house prices constantly fluctuating based on various factors such as location, size, amenities, and economic conditions. Accurately predicting house prices is crucial for both buyers and sellers, as it can help make informed decisions regarding buying, selling, or investing in properties.
* Traditional linear regression models are often employed for house price prediction. However, they may not capture complex relationships between predictors and the target variable, leading to suboptimal predictions. In this project, we will explore advanced regression techniques to enhance the accuracy and robustness of house price prediction models.
* Emphasize the need for advanced regression techniques like Gradient Boosting and XGBoost to enhance prediction accuracy.

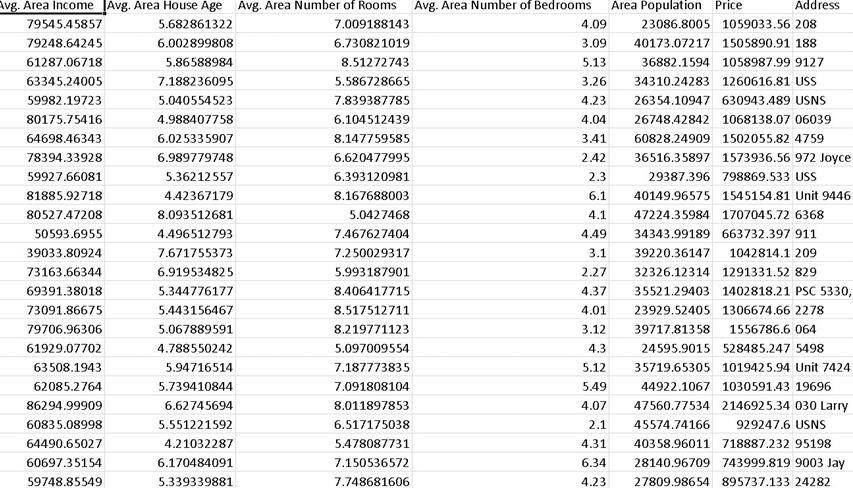
## Phase 3: Development Part 1

In this part you will begin building your project by loading and preprocessing the dataset. Start building the house price prediction model by loading and preprocessing the dataset. Load the housing dataset and preprocess the data.

## Data Source

A good data source for house price prediction using machine learning should be Accurate, Complete, Covering the geographic area of interest, Accessible.

Dataset Link: (https://[www.kaggle.com/datasets/vedavyasv/usa-housing)](http://www.kaggle.com/datasets/vedavyasv/usa-housing))



#### Import Libraries:

Import the necessary libraries for your project. You'll likely need libraries such as Pandas, NumPy, and Scikit-Learn

Source Code:

import pandas as pd import numpy as np

from sklearn.model\_selection import train\_test\_split

#### Load the dataset:

Load your housing dataset into a Pandas DataFrame. You can typically load data from a CSV file using the pd.read\_csv() function

Source Code:

# Load the dataset

data = pd.read\_csv('housing\_data.csv') # Replace 'housing\_data.csv' with the actual filename.

#### Explore the data:

Take a look at the data to understand its structure and content. You can use functions like data.head(), data.info(), and data.describe() to get an overview.

Source Code:

# Display the first few rows of the dataset print(data.head())

# Get information about the dataset print(data.info())

# Summary statistics of the data print(data.describe())

# Data Preprocessing:

### Handling Missing Data:

Check for missing values in the dataset and decide how to handle them (e.g., by filling missing values with the mean, median, or using other techniques).

Source Code:

# Handle missing values (if any)

data.fillna(method='ffill', inplace=True) # Example: Forward fill missing values

### Feature Engineering:

Create new features or transform existing ones if necessary. For

example, you might want to create features like the age of the house, the price per square foot, etc.

### Categorical Data:

If your dataset contains categorical variables, you may need to encode them using techniques like one-hot encoding.

Source Code:

# Example one-hot encoding

data = pd.get\_dummies(data, columns=['categorical\_column'])

### Scaling:

Normalize or standardize numerical features to ensure that they are on a similar scale.

Source Code:

from sklearn.preprocessing import StandardScaler scaler = StandardScaler() data[['numerical\_column1', 'numerical\_column2']] =

scaler.fit\_transform(data[['numerical\_column1', 'numerical\_column2']])

# Split the data:

Split your dataset into training and testing sets. This is typically done to assess the model's performance on unseen data.

Source Code:

# Define the target variable (house prices) and features (predictors) X = data.drop('price', axis=1)

y = data['price']

# Split the data into training and testing sets

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

## Conclusion and Future Work:

We have loaded and preprocessed the dataset. Next step we can proceed to build and train the house price prediction model in the next phase of the project.

#### Project Conclusion:

In Phase 3 of the project, the housing dataset was loaded and preprocessed. Key steps included handling missing data, feature engineering, categorical data encoding, and feature scaling. Exploratory analysis was performed to gain insights into the data's structure and quality. The dataset was then split into training and testing sets for model evaluation.

These essential preprocessing steps have paved the way for subsequent phases, where the house price prediction model will be developed and assessed.

Thank You

Done by –

Rajnish Kumar