**PHASE 3 ASSIGNMENT**

**PROJECT TITLE: PREPROCESSING THE DATASET**

**PROBLEM DEFINITION:**The problem is to predict house prices using

machine learning techniques. The objective is to develop a model that accurately

predicts the prices of houses based on a set of features such as location, square

footage, number of bedrooms and bathrooms, and other relevant factors. This

project involves data preprocessing, feature engineering, model selection, training,

and evaluation.

**GITHUB LINK:**

[https://github.com/Sabitha78/predicting-house-prices-using machine-learning.git](https://github.com/Jayakrishnabharathi/predicting-house-prices-using%20machine-learning.git)

**DOCUMENT:**

**Building the project by preprocessing the data**

**DATASET LINK ON: Predicting House Prices**

<https://www.kaggle.com/datasets/vedavyasv/usa-housing>

Preprocessing a dataset is a crucial step in preparing data for machine learning models. The specific steps can vary depending on the nature of your data and the problem you're trying to solve. However, here's a general set of steps you might follow:

**1. \*\*Import Libraries:\*\***

- Import the necessary libraries for data manipulation and analysis such as Pandas, NumPy, and others.

```python

import pandas as pd

import numpy as np

```



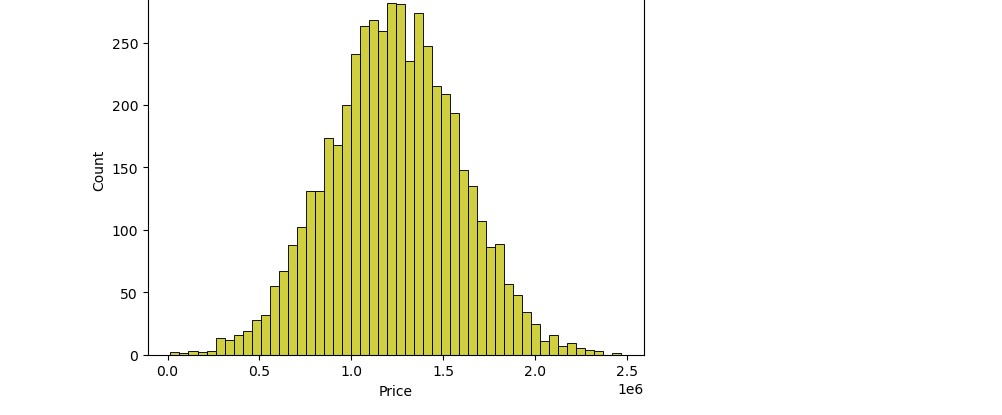
**2. \*\*Load the Dataset:\*\***

- Read the dataset into a Pandas DataFrame.

```python

data = pd.read\_csv('your\_dataset.csv')

```



**3. \*\*Explore the Data:\*\***

- Check for missing values, understand the structure of the data, and explore basic statistics.

```python

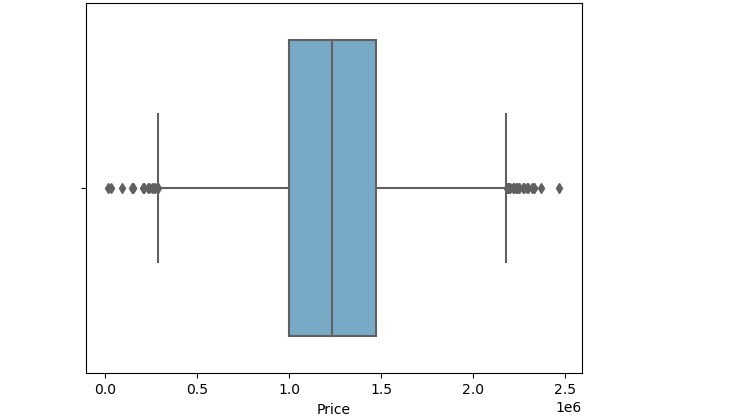
# Check for missing values

print(data.isnull().sum())

# Basic statistics

print(data.describe())

```



**4. \*\*Handle Missing Values:\*\***

- Decide on a strategy for handling missing data. Options include dropping missing values, filling them with mean or median, or using more advanced imputation techniques.

```python

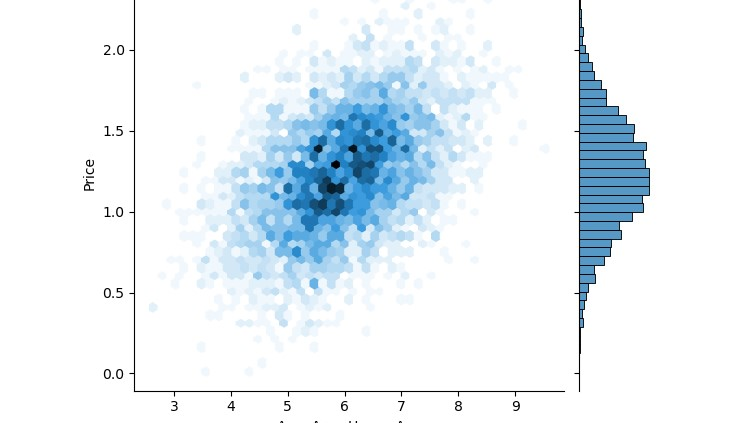
# Drop rows with missing values

data = data.dropna()

# Fill missing values with mean

data = data.fillna(data.mean())

```



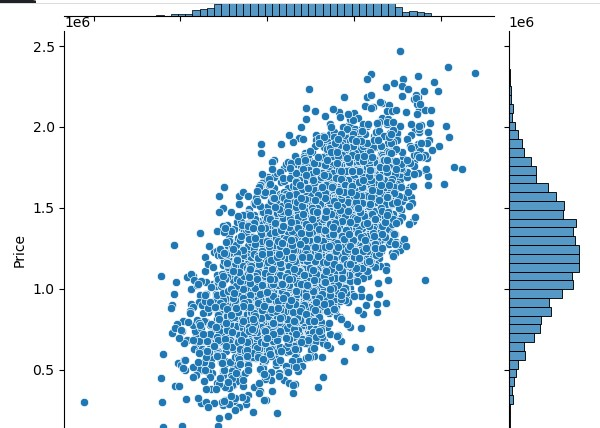
**5. \*\*Remove Duplicates:\*\***

- Check for and remove duplicate rows.

```python

data = data.drop\_duplicates()

```



**6. \*\*Handle Categorical Data:\*\***

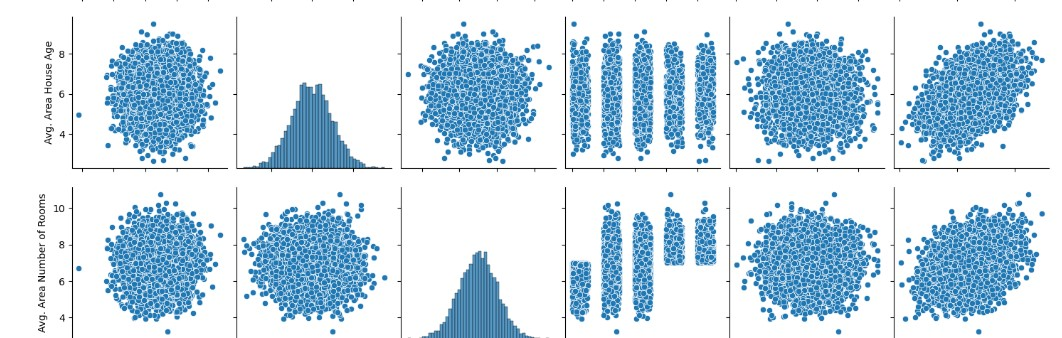
- Convert categorical variables into numerical format, using techniques like one-hot encoding or label encoding.

```python

# One-hot encoding

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

```



**7. \*\*Feature Scaling:\*\***

- Standardize or normalize numerical features to ensure they are on similar scales.

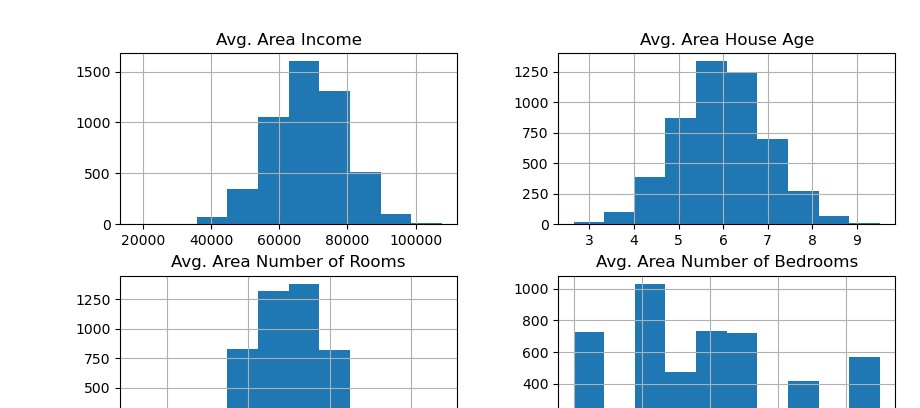
```python

from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()

data[['numerical\_column']] = scaler.fit\_transform(data[['numerical\_column']])

```



**8. \*\*Feature Engineering:\*\***

- Create new features or transform existing ones to better represent the underlying patterns in the data.

```python

# Example: Create a new feature

data['new\_feature'] = data['feature1'] \* data['feature2']

```



**9. \*\*Split the Dataset:\*\***

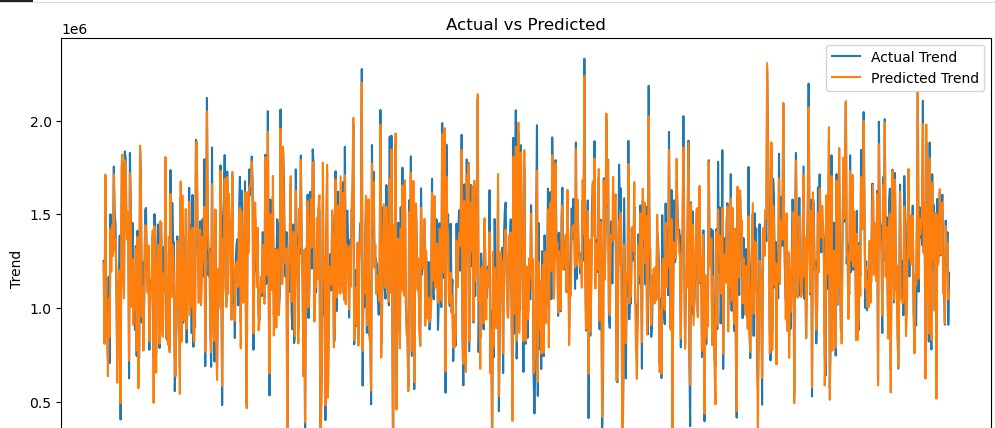
- Split the dataset into training and testing sets.

```python

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

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

```



**10. \*\*Save Preprocessed Data (Optional):\*\***

- Save the preprocessed data to a new file for future use.

```python

data.to\_csv('preprocessed\_data.csv', index=False)

**SUBMITTED BY:**

**STUDENT REG NO:** 711221104043

**NAAN MUDHALVAN:** au711221104043