

## **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/Jayakrishnabharathi/predicting-house-prices-using-machine-learning.git>

<https://github.com/Jayakrishnabharathi/innovation.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
```
```

## Data Exploration

In [3]:

```
dataset
```

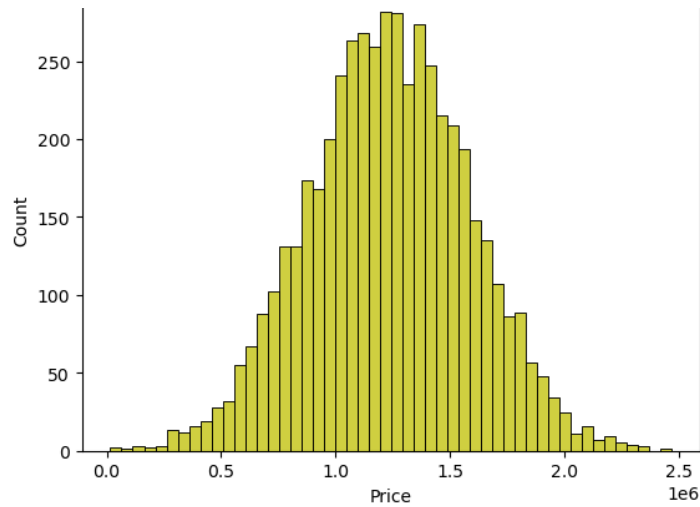
Out[3]:

|      | Avg. Area Income | Avg. Area House Age | Avg. Area Number of Rooms | Avg. Area Number of Bedrooms | Area Population | Price        | Address   |
|------|------------------|---------------------|---------------------------|------------------------------|-----------------|--------------|---|
| 0    | 79545.458574     | 5.682861            | 7.009188                  | 4.09                         | 23086.800503    | 1.059034e+06 | 208 Michael Ferry Apt. 674\nLaurabury, NE 3701... |
| 1    | 79248.642455     | 6.002900            | 6.730821                  | 3.09                         | 40173.072174    | 1.505891e+06 | 188 Johnson Views Suite 079\nLake Kathleen, CA... |
| 2    | 61287.067179     | 5.865890            | 8.512727                  | 5.13                         | 36882.159400    | 1.058988e+06 | 9127 Elizabeth Stravenue\nDanieltown, WI 06482... |
| 3    | 63345.240046     | 7.188236            | 5.586729                  | 3.26                         | 34310.242831    | 1.260617e+06 | USS Barnett\nFPO AP 44820                         |
| 4    | 59982.197226     | 5.040555            | 7.839388                  | 4.23                         | 26354.109472    | 6.309435e+05 | USNS Raymond\nFPO AE 09386                        |
| ...  | ...              | ...                 | ...                       | ...                          | ...             | ...          | ...   |
| 4995 | 60567.944140     | 7.830362            | 6.137356                  | 3.46                         | 22837.361035    | 1.060194e+06 | USNS Williams\nFPO AP 30153-7653                  |

## 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
```

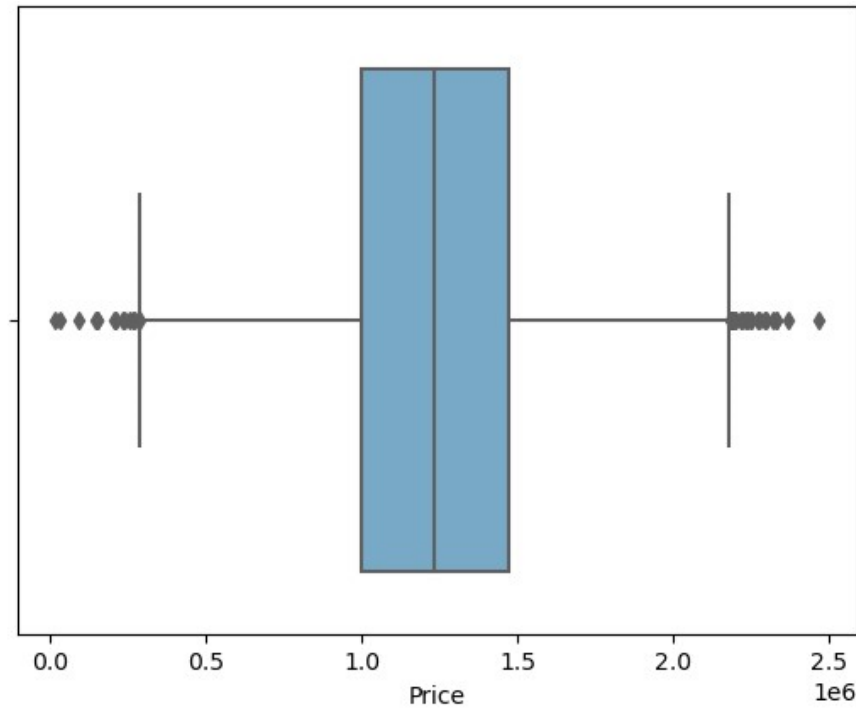
```
# Check for missing values
```

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

```
# Basic statistics
```

```
print(data.describe())
```

```
``
```

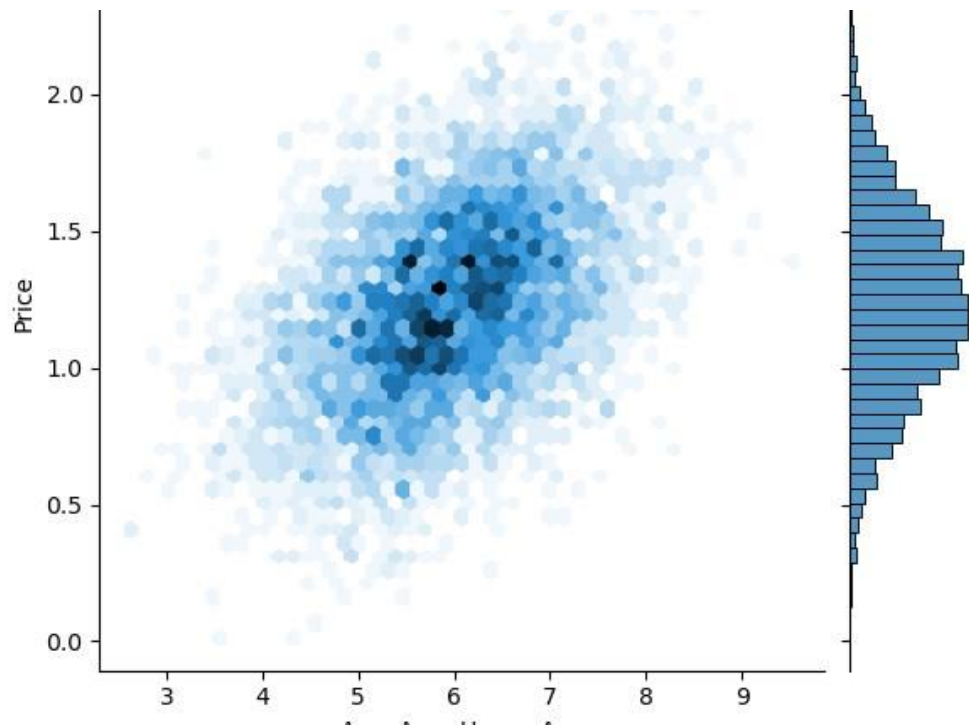


#### 4. **\*\*Handle Missing Values:\*\***

- 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())
```
```



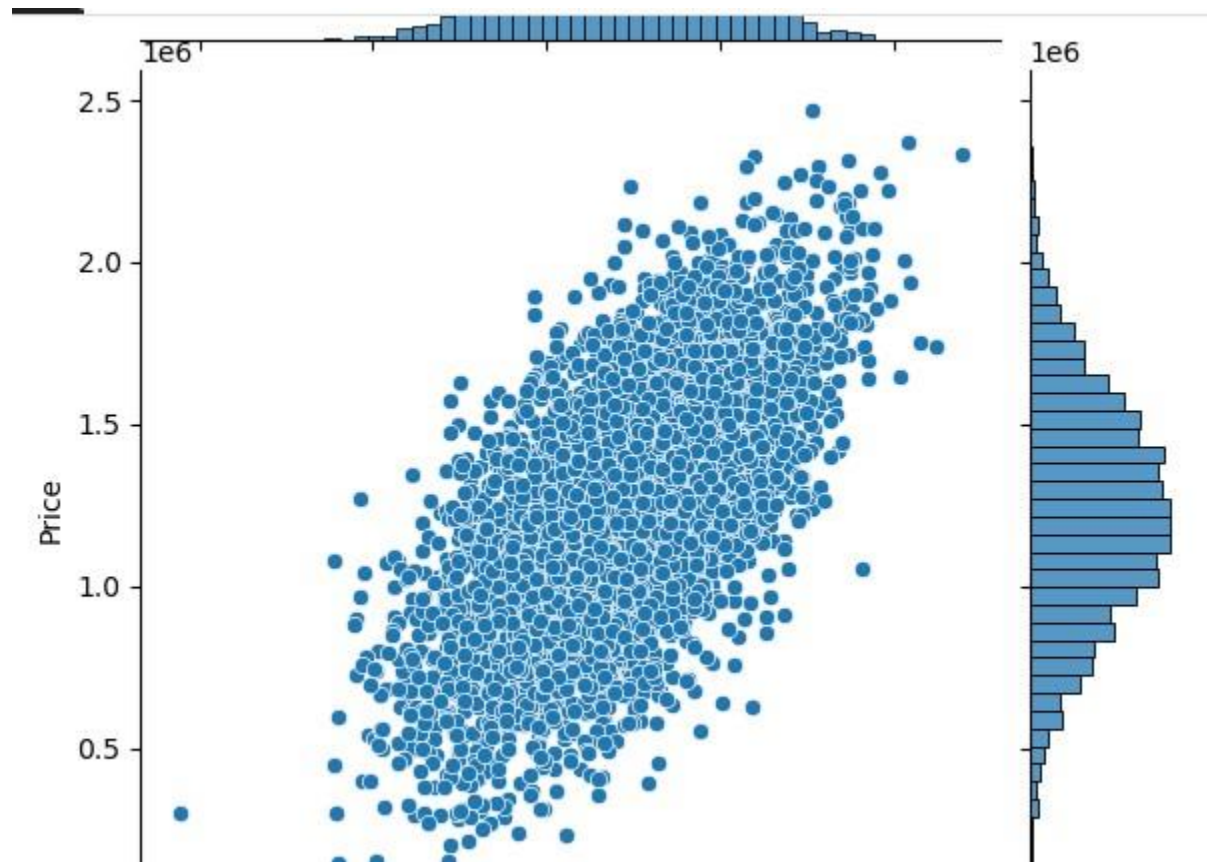
## 5. **\*\*Remove Duplicates:\*\***

- Check for and remove duplicate rows.

```
```python
```

```
data = data.drop_duplicates()
```

```
```
```



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

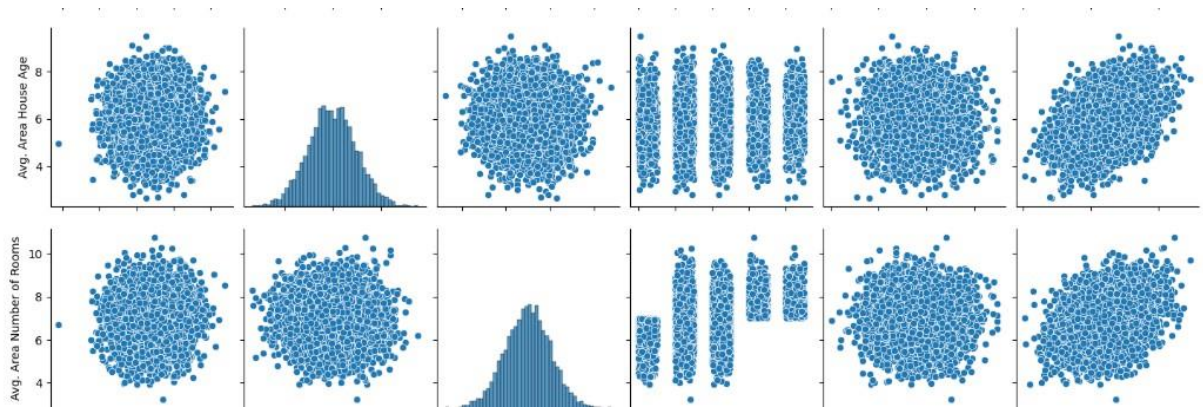
- Convert categorical variables into numerical format, using techniques like one-hot encoding or label encoding.

```
```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.

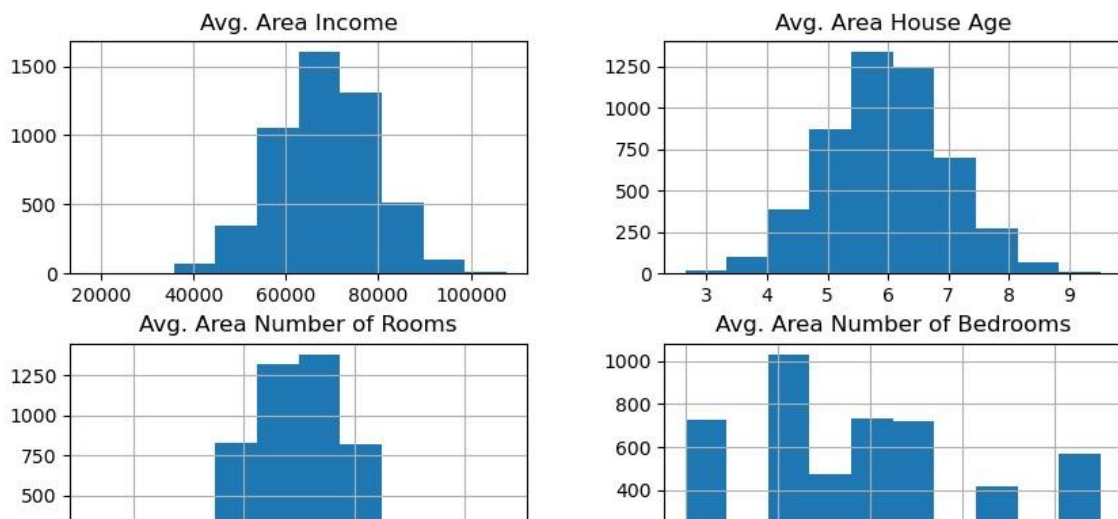
```
```python
```

```
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']
```
```



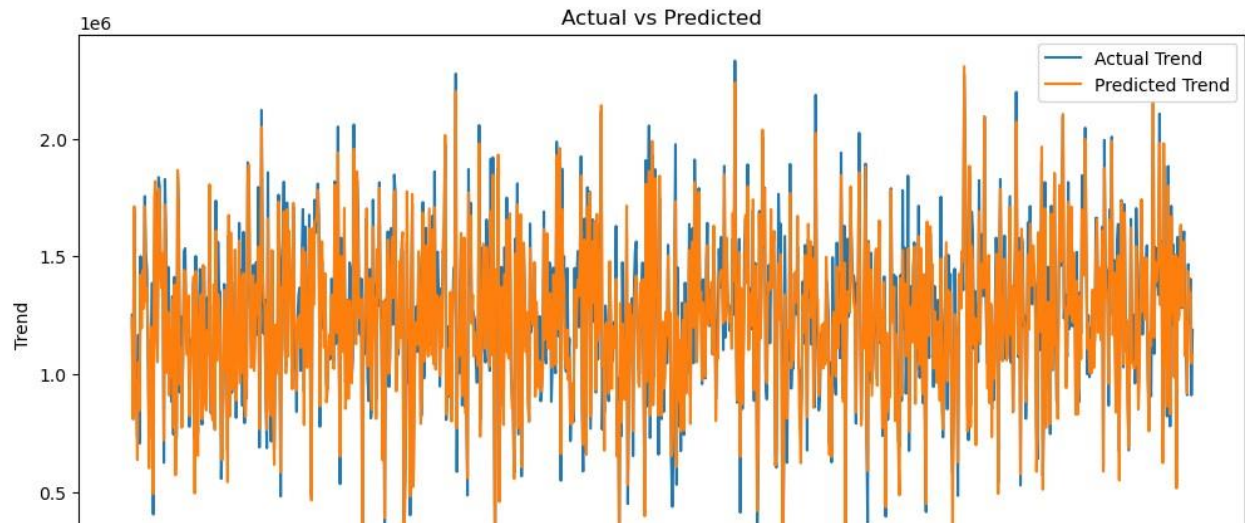
## 9. **\*\*Split the Dataset:\*\***

- 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:** 711221104022

**NAAN MUDHALVAN:** au711221104022