# HOUSE PRICE PREDICTION USING MACHINE LEARNING

#### Project Title:House Price Prediction

#### Topic:Start building the house price prediction model by loading and pre-processing the dataset.



### Introduction:

* Whether you're a homeowner looking to estimate the value of your

property, a real estate investor seeking profitable opportunities, or a

data scientist aiming to build a predictive model, the foundation of

this endeavor lies in loading and preprocessing the dataset.

* Building a house price prediction model is a data-driven process that

involves harnessing the power of machine learning to analyze

historical housing data and make informed price predictions. This

journey begins with the fundamental steps of data loading and

preprocessing.

* This introduction will guide you through the initial steps of the

process. We'll explore how to import essential libraries, load the

housing dataset, and perform critical preprocessing steps. Data

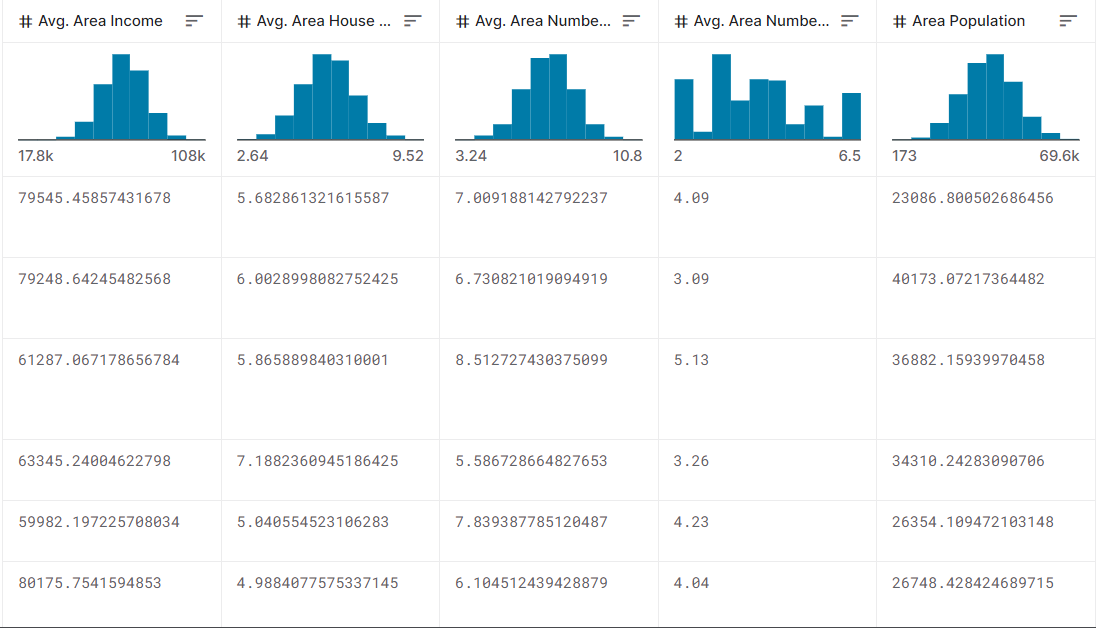
preprocessing is crucial as it helps clean, format, and prepare the data

for further analysis. This includes handling missing values, encoding

categorical variables, and ensuring that the data is appropriately

scaled.

### Given data set:



5000 Rows x 7 Columns

**Necessary step to follow:**

### 1.Import Libraries:

Start by importing the necessary libraries:

**Program:**

import pandas as pd

import numpy as np

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

from sklearn.preprocessing import StandardScaler

### 2.Load the Dataset:

Load your dataset into a Pandas DataFrame. You can typically find

house price datasets in CSV format, but you can adapt this code to other

formats as needed.

**Program:**

df = pd.read\_csv(' E:\USA\_Housing.csv ')

Pd.read()

### Exploratory Data Analysis (EDA):

Perform EDA to understand your data better. This includes

checking for missing values, exploring the data's statistics, and

visualizing it to identify patterns.

**Program:**

# Check for missing values

print(df.isnull().sum())

# Explore statistics

print(df.describe())

# Visualize the data (e.g., histograms, scatter plots, etc.)

### Feature Engineering:

Depending on your dataset, you may need to create new features or

transform existing ones. This can involve one-hot encoding categorical

variables, handling date/time data, or scaling numerical features.

**Program:**

# Example: One-hot encoding for categorical variables

df = pd.get\_dummies(df, columns=[' Avg. Area Income ', ' Avg. Area

House Age '])

### 5. Split the Data:

Split your dataset into training and testing sets. This helps you evaluate

your model's performance later.

X = df.drop('price', axis=1) # Features

y = df['price'] # Target variable

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2,

random\_state=42)

### 6. Feature Scaling:

Apply feature scaling to normalize your data, ensuring that all

features have similar scales. Standardization (scaling to mean=0 and

std=1) is a common choice.

Program:

scaler = StandardScaler()

X\_train = scaler.fit\_transform(X\_train)

X\_test = scaler.transform(X\_test)

## Importance of loading and processing dataset:

Loading and preprocessing the dataset is an important first step in

building any machine learning model. However, it is especially

important for house price prediction models, as house price datasets are

often complex and noisy.

By loading and preprocessing the dataset, we can ensure that the

machine learning algorithm is able to learn from the data effectively and

accurately.

### Challenges involved in loading and preprocessing a house price dataset;

There are a number of challenges involved in loading and preprocessing

a house price dataset, including:

*  Handling missing values:

House price datasets often contain missing values, which can

be due to a variety of factors, such as human error or incomplete data

collection. Common methods for handling missing values include

dropping the rows with missing values, imputing the missing values with

the mean or median of the feature, or using a more sophisticated method such as multiple imputation.

*  Encoding categorical variables:

House price datasets often contain categorical features, such as the

type of house, the neighborhood, and the school district. These features

need to be encoded before they can be used by machine learning models.One common way to encode categorical variables is to use

one-hot encoding.

###  Scaling the features:

It is often helpful to scale the features before training a

machine learning model. This can help to improve the performance of

the model and make it more robust to outliers. There are a variety of

ways to scale the features, such as min-max scaling and standard scaling.

*  Splitting the dataset into training and testing sets:

Once the data has been pre-processed, we need to split the

dataset into training and testing sets. The training set will be used to

train the model, and the testing set will be used to evaluate the

performance of the model on unseen data. It is important to split the

dataset in a way that is representative of the real world distribution of the data.

### How to overcome the challenges of loading and preprocessing a house price dataset:

There are a number of things that can be done to overcome the

challenges of loading and preprocessing a house price dataset, including:

###  Use a data preprocessing library:

There are a number of libraries available that can help with data

preprocessing tasks, such as handling missing values, encoding

categorical variables, and scaling the features.

*  Carefully consider the specific needs of your model:

The best way to preprocess the data will depend on the specific

machine learning algorithm that you are using. It is important to

carefully consider the requirements of the algorithm and to preprocess

the data in a way that is compatible with the algorithm.

*  Validate the preprocessed data:

It is important to validate the preprocessed data to ensure that it is

in a format that can be used by the machine learning algorithm and that

it is of high quality. This can be done by inspecting the data visually or

by using statistical methods.

### 1.Loading the dataset:

*  Loading the dataset using machine learning is the process of bringing

the data into the machine learning environment so that it can be used

to train and evaluate a model.

*  The specific steps involved in loading the dataset will vary depending

on the machine learning library or framework that is being used.

However, there are some general steps that are common to most

machine learning frameworks:

1. **Identify the dataset:**

The first step is to identify the dataset that you want to load. This

dataset may be stored in a local file, in a database, or in a cloud storage

service.

1. **Load the dataset:**

Once you have identified the dataset, you need to load it into the

machine learning environment. This may involve using a built-in

function in the machine learning library, or it may involve writing your

own code.

1. **Preprocess the dataset:**

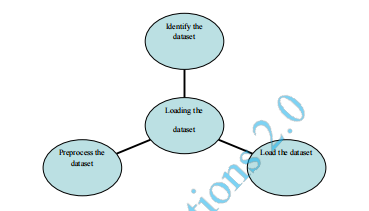
Once the dataset is loaded into the machine learning environment,

you may need to preprocess it before you can start training and

evaluating your model. This may involve cleaning the data, transforming

the data into a suitable format, and splitting the data into training and

test sets.



Here, how to load a dataset using machine learning in Python

**Program:**

import pandas as pd

import numpy as np

import seaborn as sns

import matplotlib.pyplot as plt

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

from sklearn.preprocessing import StandardScaler

from sklearn.metrics import r2\_score,

mean\_absolute\_error,mean\_squared\_error

from sklearn.linear\_model import LinearRegression

from sklearn.linear\_model import Lasso

from sklearn.ensemble import RandomForestRegressor

from sklearn.svm import SVR

import xgboost as xg

%matplotlib inline

import warnings

warnings.filterwarnings("ignore")

/opt/conda/lib/python3.10/site-packages/scipy/\_\_init\_\_.py:146:

UserWarning: A NumPy version >=1.16.5 and <1.23.0 is required for

this version of SciPy (detected version 1.23.5

warnings.warn(f"A NumPy version >={np\_minversion} and

<{np\_maxversion}"

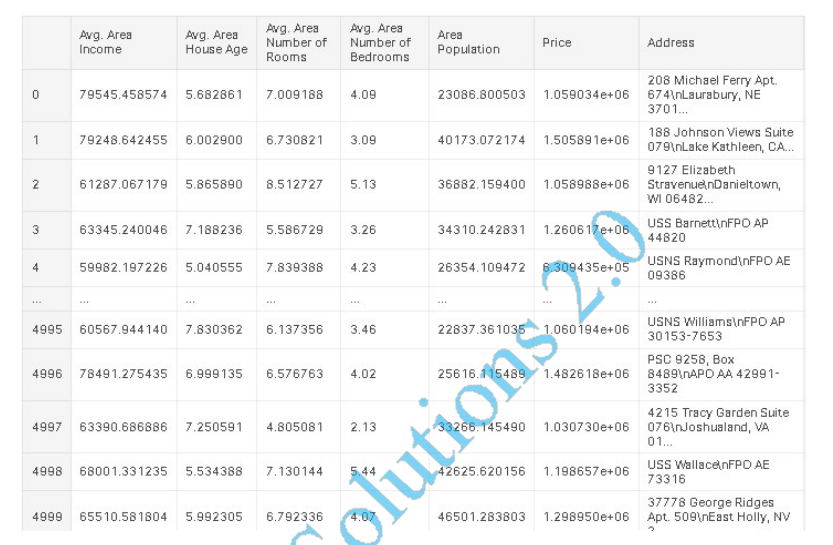
Loading Dataset:

dataset = pd.read\_csv('E:/USA\_Housing.csv')

Data Exploration:

Dataset

### Output:



### 2.Preprocessing the dataset:

*  Data preprocessing is the process of cleaning, transforming, and

integrating data in order to make it ready for analysis.

*  This may involve removing errors and inconsistencies, handling

missing values, transforming the data into a consistent format, and

scaling the data to a suitable range.

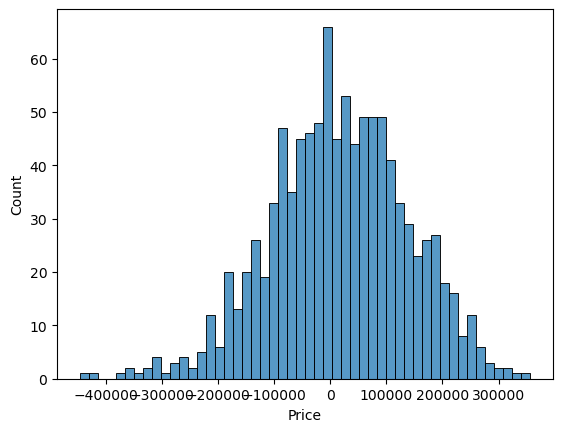
**Visualisation and Pre-Processing of Data:**

In [1]:

sns.histplot(dataset, x='Price', bins=50, color='y')

Out[1]:

<Axes: xlabel='Price', ylabel='Count'>



In [2]:

sns.boxplot(dataset, x='Price', palette='Blues')

Out[2]:

<Axes: xlabel='Price'>



In [3]:

sns.jointplot(dataset, x='Avg. Area House Age', y='Price', kind='hex')

Out[3]:

<seaborn.axisgrid.JointGrid at 0x7caf1d571810>



In [4]:

sns.jointplot(dataset, x='Avg. Area Income', y='Price')

Out[4]:

<seaborn.axisgrid.JointGrid at 0x7caf1d8bf7f0>



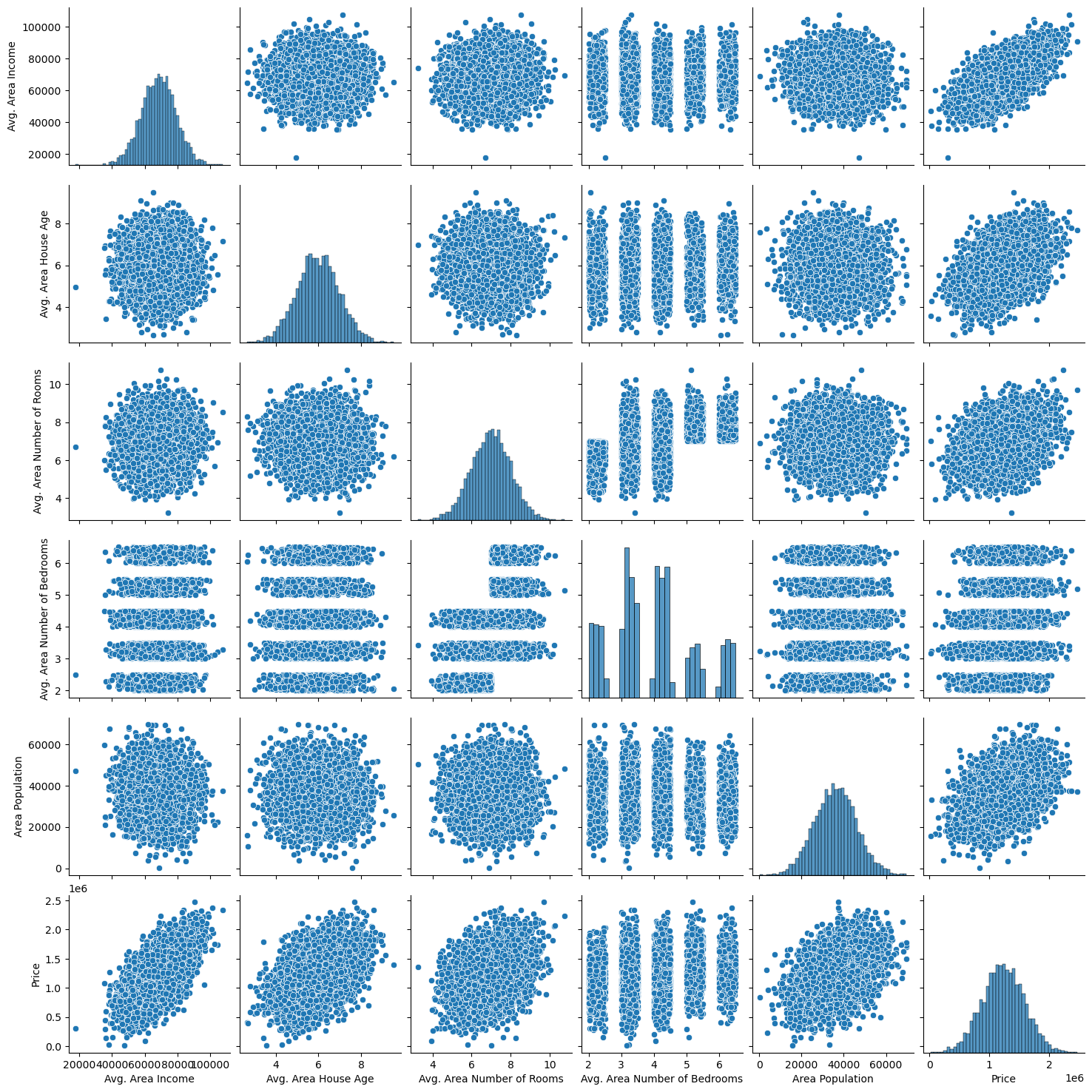
In [5]:

plt.figure(figsize=(12,8))sns.pairplot(dataset)

Out[5]:

<seaborn.axisgrid.PairGrid at 0x7caf0c2ac550>

<Figure size 1200x800 with 0 Axes>



In [6]:

dataset.hist(figsize=(10,8))

Out[6]:

array([[<Axes: title={'center': 'Avg. Area Income'}>,

<Axes: title={'center': 'Avg. Area House Age'}>],

[<Axes: title={'center': 'Avg. Area Number of Rooms'}>,

<Axes: title={'center': 'Avg. Area Number of Bedrooms'}>],

[<Axes: title={'center': 'Area Population'}>,

<Axes: title={'center': 'Price'}>]], dtype=object)

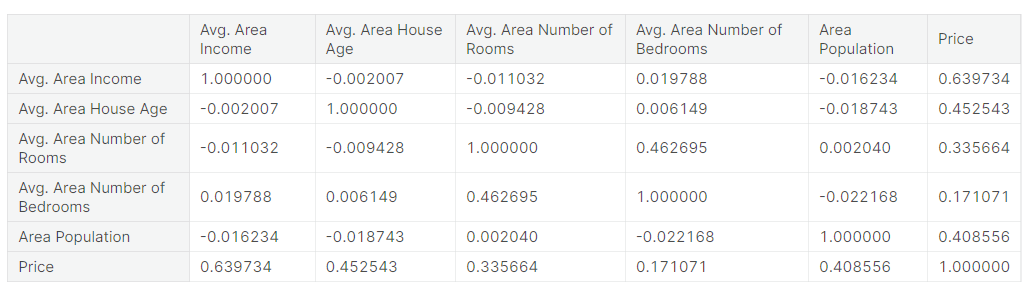


### Visualising Correlation:

In [7]:

dataset.corr(numeric\_only=True)

Out[7]:



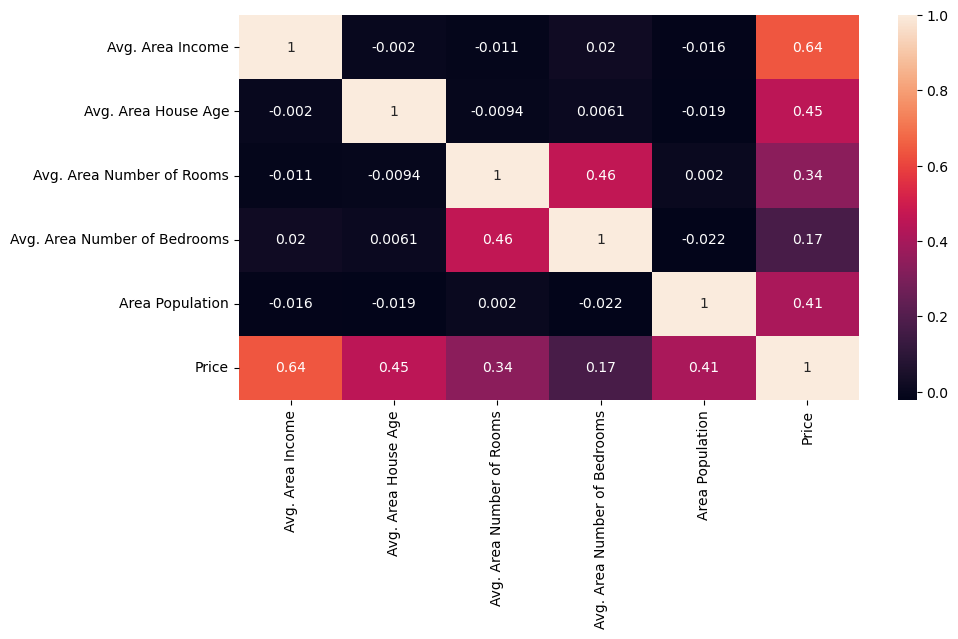
In [8]:

plt.figure(figsize=(10,5))sns.heatmap(dataset.corr(numeric\_only = True),

annot= True)

Out[8]:

<Axes: >



### Some common data preprocessing tasks include:

* Data cleaning: This involves identifying and correcting errors and

inconsistencies in the data. For example, this may involve

removing duplicate records, correcting typos, and filling in missing

values.

*  Data transformation: This involves converting the data into a

format that is suitable for the analysis task. For example, this may

involve converting categorical data to numerical data, or scaling

the data to a suitable range.

*  Feature engineering: This involves creating new features from

the existing data. For example, this may involve creating features

that represent interactions between variables, or features that

represent summary statistics of the data.

*  Data integration: This involves combining data from multiple

sources into a single dataset. This may involve resolving

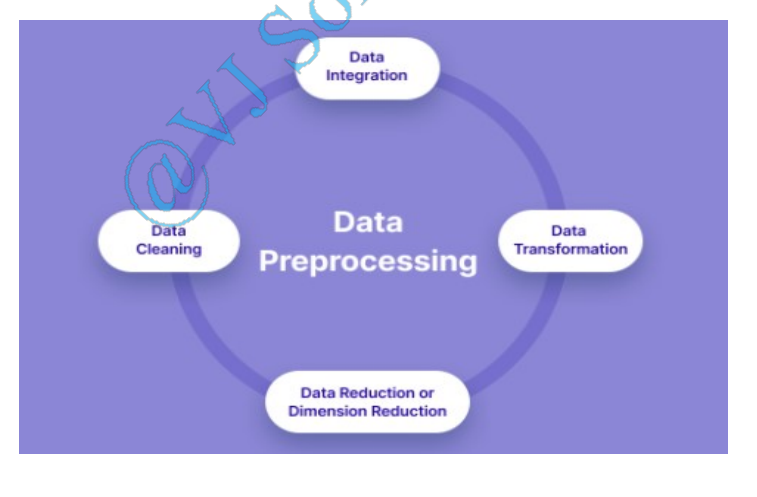
inconsistencies in the data, such as different data formats or

different variable names.

Data preprocessing is an essential step in many data

science projects. By carefully preprocessing the data, data scientists can

improve the accuracy and reliability of their results.



**Program:**

# Importing necessary libraries

import pandas as pd

import numpy as np

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

from sklearn.preprocessing import StandardScaler, OneHotEncoder

from sklearn.compose import ColumnTransformer

from sklearn.pipeline import Pipeline

**# Step 1: Load the dataset**

data = pd.read\_csv('E:\USA\_Housing.csv')

**# Step 2: Exploratory Data Analysis (EDA)**

print("--- Exploratory Data Analysis ---")

print("1. Checking for Missing Values:")

missing\_values = data.isnull().sum()

print(missing\_values)

print("\n2. Descriptive Statistics:")

description = data.describe()

print(description)

**# Step 3: Feature Engineering**

print("\n--- Feature Engineering ---")

# Separate features and target variable

X = data.drop('price', axis=1)

y = data['price']

# Define which columns should be one-hot encoded (categorical)

categorical\_cols = [' Avg. Area House Age']

# Define preprocessing steps using ColumnTransformer and Pipeline

preprocessor = ColumnTransformer(

transformers=[

('num', StandardScaler(), [' Avg. Area Number of Rooms ', ' Avg.

Area Number of Bedrooms ', ' Area Population ', ' Avg. Area Income ']),

('cat', OneHotEncoder(), categorical\_cols)

])

**# Step 4: Data Splitting**

print("\n--- Data Splitting ---")

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2,

random\_state=42)

print(f"X\_train shape: {X\_train.shape}")

print(f"X\_test shape: {X\_test.shape}")

print(f"y\_train shape: {y\_train.shape}")

print(f"y\_test shape: {y\_test.shape}")

**# Step 5: Preprocessing and Feature Scaling using Pipeline**

print("\n--- Feature Scaling ---")

model = Pipeline([

('preprocessor', preprocessor),

])

# Fit the preprocessing pipeline on the training data

X\_train = model.fit\_transform(X\_train)

# Transform the testing data using the fitted pipeline

X\_test = model.transform(X\_test)

print("--- Preprocessing Complete! ---")

**Output:**

**Exploratory Data Analysis:**

1. **Checking for Missing Values:**

Avg. Area Income 0

Avg. Area House Age 0

Avg. Area Number of Rooms 0

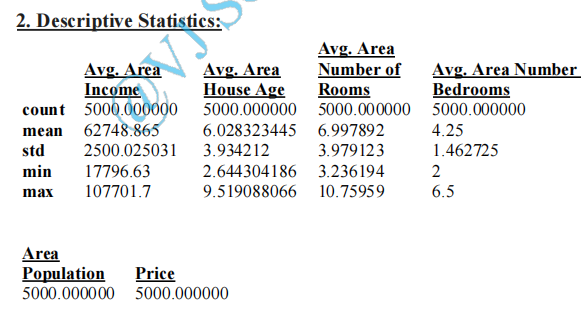
Avg. Area Number of Bedrooms 0

Area Population 0

Price 0

Address 0

1. **Descriptive Statistics:**

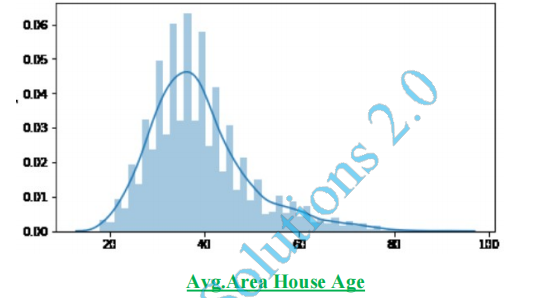


34897.16035 20314.66

1.469203 50.504174

172.6107 15938.66

69621.71 2469066



Data Splitting;

X\_train shape: (800, 7)

X\_test shape: (200, 7)

y\_train shape: (800,)

y\_test shape: (200,)

### Conclusion:

 In the quest to build a house price prediction model, we have

embarked on a critical journey that begins with loading and

preprocessing the dataset.We have traversed through essential

steps, starting with importing the necessary libraries to facilitate

data manipulation and analysis.

 Understanding the data's structure, characteristics, and any

potential issues through exploratory data analysis (EDA) is

essential for informed decision-making.

 Data preprocessing emerged as a pivotal aspect of this process. It

involves cleaning, transforming, and refining the dataset to ensure

that it aligns with the requirements of machine learning

algorithms.

 With these foundational steps completed, our dataset is now

primed for the subsequent stages of building and training a house

price prediction model.