PREDICTING HOUSE PRICE USING MACHINE LEARNING

BATCH MEMBER

NAME: SHAKTHI.S

REGISTER NUMBER:

210221104031

Phase 3 submission document

Project Title: House Price Prediction

Phase 3: ***Development Part 1***

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

House Price Prediction

Introduction:

1. The is the introduction for house price prediction.Whether you're a homeowner looking to estimate the value of your house property, a real estate investor seeking profitable opportunities, or adata scientist aiming to build a predictive model,house price prediction can help developer to selling a price of house and can help customer to arrange the right time to purchase a house.

2. Building a house price prediction model is a data-driven process that involves harnessing the power of machine learning to analyse historical housing data and make informed price predictions. This journey begins with the fundamental steps of data loading and preprocessing.

3. 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 datafor further analysis. This includes handling missing values, encoding categorical variables, and ensuring that the data is appropriately scaled.

Necessary step to follow:

1.Import Libraries:

Start by importing the necessary libraries:

Program:

```
import pandas as pdimport

numpy as np

from sklearn.model_selection import train_test_splitfrom

sklearn.preprocessing import StandardScaler
```

2.Load the Dataset:

Load your dataset into a Pandas DataFrame. You can typically findhouse 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()
```

2. Exploratory Data Analysis (EDA):

Perform EDA to understand your data better. This includeschecking for missing values, exploring the data's statistics, and visualizing it to identify patterns.

Program:

```
# Check for missing values
print(df.isnull().sum())

# Explore statisticsprint(df.
describe())

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

3. Feature Engineering:

Depending on your dataset, you may need to create new features ortransform 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. AreaHouse Age '])
```

4. Split the Data:

Split your dataset into training and testing sets. This helps you evaluateyour model's performance later.

```
X = df.drop('price', axis=1) # Featuresy = df['price']   #

Target variable

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,random_state=42)
```

5. Feature Scaling:

Apply feature scaling to normalize your data, ensuring that all features have similar scales. Standardization (scaling to mean=0 andstd=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 inbuilding any machine learning model. However, it is especially important for house price prediction models..

By loading and preprocessing the dataset, we can ensure that the machine learning algorithm is able to learn from the data effectively andaccurately.

Challenges involved in loading and preprocessing a house pricedataset;

There are a number of challenges involved in loading and preprocessinga house price dataset, including:

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

7. 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.

8. 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.

How to overcome the challenges of loading and preprocessing ahouse 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:

1. Use a data preprocessing library:

There are a number of libraries available that can help with datapreprocessing tasks, such as handling missing values, encoding categorical variables, and scaling the features.

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

Loading the dataset:

9. Loading the dataset using machine learning is the process of bringingthe data into the machine learning environment so that it can be used to train and evaluate a model.

Specific steps in dataset:

10. The specific steps involved in loading the dataset will vary dependingon 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:

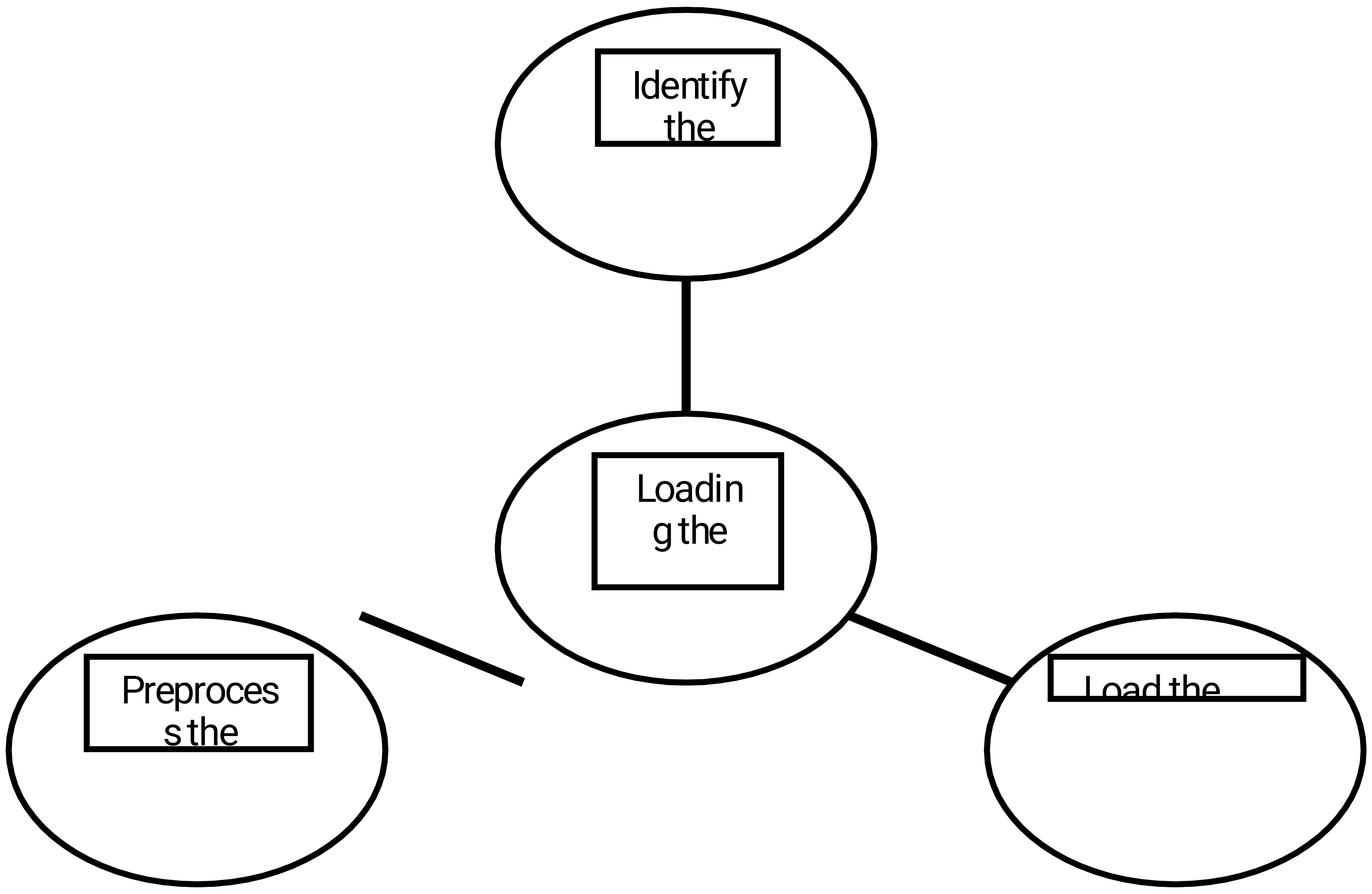
The first step is to identify the dataset that you want to load. Thisdataset may be stored in a local file, in a database, or in a cloud storageservice.

2.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 yourown 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 andtest 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_splitfrom

sklearn.preprocessing import StandardScaler

```
from sklearn.metrics import r2_score,
mean_absolute_error,mean_squared_error

from sklearn.linear_model import LinearRegressionfrom

sklearn.linear_model import Lasso

from sklearn.ensemble import RandomForestRegressorfrom sklearn.svm

import SVR

import xgboost as xg

%matplotlib inlineimport

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 forthis 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:

11. Preprocessing the dataset:

- Data preprocessing is the process of cleaning, transforming, andintegrating 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, andscaling 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]:

	Avg. Area Income	Avg. Area HouseAge	Avg. Area Number of Rooms	Avg. Area Number of Bedrooms	Area Population	Price
Avg. Area Income	1.000000	-0.002007	-0.011032	0.019788	-0.016234	0.639734
Avg. Area House Age	-0.002007	1.000000	-0.009428	0.006149	-0.018743	0.452543
Avg. Area Number ofRooms	-0.011032	-0.009428	1.000000	0.462695	0.002040	0.335664
Avg. Area Number of Bedrooms	0.019788	0.006149	0.462695	1.000000	-0.022168	0.171071
Area Population	-0.016234	-0.018743	0.002040	-0.022168	1.000000	0.408556

Price	0.639734	0.452543	0.335664	0.171071	0.408556	1.000000

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:

14. **Data cleaning:** This involves identifying and correcting errors andinconsistencies in the data. For example, this may involve removing duplicate records, correcting typos, and filling in missingvalues.

15. **Data transformation:** This involves converting the data into a format that is suitable for the analysis task. For example, this mayinvolve converting categorical data to numerical data, or scaling the data to a suitable range.
16. **Feature engineering:** This involves creating new features from the existing data. For example, this may involve creating featuresthat represent interactions between variables, or features that represent summary statistics of the data.
17. **Data integration:** This involves combining data from multiplesources into a single dataset. This may involve resolving inconsistencies in the data, such as different data formats or different variable names.

.

Program:

```
# Importing necessary librariesimport
pandas as pd

import numpy as np

from sklearn.model_selection import train_test_split

from sklearn.preprocessing import StandardScaler, OneHotEncoderfrom 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 variableX =

    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 Pipelinepreprocessor =

    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 dataX_train =
    model.fit_transform(X_train)

    # Transform the testing data using the fitted pipelineX_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


0Area Population



0
                                            0
Price                                 0
Address
```

2. Descriptive Statistics:

	Avg. Area Income	Avg. Area House Age	Avg. Area Number of Rooms	Avg. Area Number of Bedrooms
count	5000.000000	5000.000000	5000.000000	5000.000000

mean	62748.865	6.028323445	6.997892	4.25
std	2500.025031	3.934212	3.979123	1.462725
min	17796.63	2.644304186	3.236194	2
max	107701.7	9.519088066	10.75959	6.5

<u>Area</u>

<u>Population</u>	**<u>Price</u>**
5000.000000	5000.000000

34897.16035	20314.66
1.469203	50.50417 4
172.6107	15938.66
69621.71	2469066

Avg.Area House Age

Data Splitting;

X_train shape: (800, 7)

X_test shape: (200, 7)

y_train shape: (800,)

y_test shape: (200,)

Preprocessing Complete

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 facilitatedata manipulation and analysis.

- Understanding the data's structure, characteristics, and any potential issues through exploratory data analysis (EDA) isessential for informed decision-making.

- Data preprocessing emerged as a pivotal aspect of this process. It involves cleaning, transforming, and refining the dataset to ensurethat 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 houseprice prediction model.