Aim - Predict the lifetime value of customers for a business based on their historical interactions.

Import Statements and Their Purposes:

- import numpy as np

Purpose: Provides support for large, multi-dimensional arrays and matrices, along with a collection of mathematical functions to operate on these arrays.

- import pandas as pd

Purpose: Used for data manipulation and analysis, including reading data from files like CSV and handling DataFrame operations.

- import matplotlib.pyplot as plt

Purpose: Essential for creating visualizations such as line charts, scatter plots, and histograms.

- from sklearn.preprocessing import OneHotEncoder, StandardScaler

Purpose:

- **OneHotEncoder**: Encodes categorical features as binary vectors (useful for machine learning models).
- **StandardScaler**: Normalizes numerical features to have a mean of 0 and a standard deviation of 1, which is necessary for models sensitive to feature scales.

- from sklearn.compose import ColumnTransformer

Purpose: Used to apply transformations like one-hot encoding to specific columns while leaving other columns unchanged.

- from sklearn.model_selection import train_test_split

Purpose: Provides a method to split the dataset into training and testing subsets, ensuring a controlled and reproducible division.

- from sklearn.linear_model import LinearRegression

Purpose: Implements a linear regression model for predictive analysis.

- from sklearn.svm import SVR

Purpose: Implements Support Vector Regression, a kernel-based approach used for both linear and non-linear regression tasks.

- import seaborn as sns

Purpose: Used for creating visually appealing statistical plots, such as scatter plots with regression lines or heatmaps.

- from sklearn.ensemble import RandomForestRegressor

Purpose: Implements Random Forest regression, an ensemble model that combines multiple decision trees to improve prediction accuracy.

Functions Used and Their Purposes:

- pd.read_csv()

Purpose: Reads a CSV file into a Pandas DataFrame.
Why: Allows for easy manipulation and analysis of structured data.

- dataset.iloc[:, :-1].values

Purpose: Extracts all columns except the last one (independent variables) as a NumPy array.
Why: This isolates the features (X) for the model.

- dataset.iloc[:, -1].values

Purpose: Extracts the last column (dependent variable or target variable) as a NumPy array (y).
Why: This isolates the target variable for the model.

- train_test_split(X, y, test_size=0.2, random_state=42)

Purpose: Splits the dataset into training (80%) and testing (20%) sets.
Why: Ensures the model is evaluated on unseen data, helping avoid overfitting. The random_state ensures reproducibility of the split.

- ColumnTransformer()

Purpose: Applies multiple transformations to different columns.
Why: This allows for encoding categorical variables while keeping other columns untouched.

- StandardScaler()

Purpose: Initializes the standard scaler, which normalizes features by removing the mean and scaling to unit variance.
Why: Ensures that features are on the same scale, which is crucial for many machine learning models.

- sc.fit_transform(X_train)

Purpose: Computes the scaling parameters (mean and standard deviation) from the training set and applies scaling to it.
Why: Prepares the data for training by normalizing it.

- sc.transform(X_test)

Purpose: Applies the scaling parameters (computed from training data) to the test data.
Why: Ensures consistent scaling between training and testing datasets.

- LinearRegression()

Purpose: Initializes the linear regression model.
Why: This is the base model for fitting a simple linear regression.

- regressor.fit(X_train, y_train)

Purpose: Trains the linear regression model on the training data (X_train, y_train).
Why: Builds the model by finding the relationship between the features and target.

- regressor.predict(X_test)

Purpose: Predicts outcomes for the test data (X_test).
Why: This generates the model's predictions for evaluation.

- plt.figure(figsize=(10, 6))

Purpose: Initializes a new figure for the plot with specified dimensions.
Why: Controls the size of the plot for better visibility.

- sns.scatterplot()

Purpose: Creates a scatter plot to visualize the relationship between the actual and predicted values.
Why: Allows for visual analysis of model performance.

- plt.plot()

Purpose: Plots a reference line representing perfect prediction (y = x).
Why: Helps to visualize how close the predictions are to the actual values.

- plt.title(), plt.xlabel(), plt.ylabel(), plt.grid()

Purpose: Adds a title, axis labels, and gridlines to the plot for better readability.
Why: Improves the clarity of the plot.

- sc.transform(test_input)

Purpose: Scales the new input data using the same parameters as the training data.
Why: Ensures consistency in feature scaling when predicting new values.

- regressor.predict(scaled_input)

Purpose: Predicts the house price for the scaled input.
Why: Makes predictions on new, unseen data using the trained model.

Why Each Step is Necessary:

- **Data Reading and Preparation:**
  Essential for loading and isolating the features and target variables for analysis.

- **Splitting the Dataset:**
  Ensures the model is evaluated on unseen data, avoiding overfitting and ensuring better generalization.

- **Feature Scaling:**
  Brings all features to the same scale, crucial for ensuring the algorithm performs optimally and handles all features appropriately.

- **Training the Model:**
  Builds the predictive relationship between features (X_train) and the target variable (y_train).

- **Prediction and Visualization:**
  Provides insights into the model's performance and allows for future predictions on new data.

- **Single Input Prediction:**
  Demonstrates how to use the trained model to make predictions for custom input data.

  ○