Implementation of Multiple Linear Regression using sklearn (House price prediction/Loan defaulter etc.)

Multiple Linear Regression

is a statistical technique used to model the **relationship between one dependent variable and two or more independent variables**. This method extends simple linear regression, which involves only one independent variable, to scenarios where multiple factors may influence the dependent variable.

Key Concepts

1. **Dependent Variable (Target):** The variable we aim to predict or explain.
2. **Independent Variables (Predictors):** The variables used to predict the dependent variable.
3. **Coefficients:** The values that multiply the independent variables. These coefficients represent the change in the dependent variable for a one-unit change in the independent variable, assuming all other variables are held constant.
4. **Intercept:** The value of the dependent variable when all independent variables are zero.

Mathematical Formulation

The formula for Multiple Linear Regression is:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_n x_n + \epsilon$$

where:

- y is the dependent variable.
- x1,x2,…,xnx_1, x_2, \ldots, x_nx1,x2,…,xn are the independent variables.
- β0\beta_0β0 is the intercept.
- β1,β2,…,βn\beta_1, \beta_2, \ldots, \beta_nβ1,β2,…,βn are the coefficients.
- ϵ epsilonϵ is the error term.

1. **No Multicollinearity:** The independent variables are not highly correlated with each other.

Steps to Perform Multiple Linear Regression

1. **Data Collection:** Gather the data that includes the dependent variable and multiple independent variables.
2. **Data Preprocessing:** Clean the data, handle missing values, and encode categorical variables if necessary.

3. **Model Training:** Use a portion of the data to train the model and estimate the coefficients.
4. **Model Evaluation:** Assess the model's performance using metrics like Mean Squared Error (MSE) and R-squared (R2R^2R2).
5. **Prediction:** Use the trained model to make predictions on new data.

1. **Import necessary libraries**
2. **Load and inspect the dataset**
3. **Preprocess the data**
4. **Split the data into training and testing sets**
5. **Train the model**
6. **Evaluate the model**

```
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
import matplotlib.pyplot as plt
import seaborn as sns

# Suppress warnings
import warnings
warnings.filterwarnings('ignore')

# Load the dataset
data = pd.read_csv('/content/Housing.csv')

# Inspect the first few rows of the dataset
print(data.head())
```

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

# Handle missing values if necessary (e.g., fill with mean, median, or
drop)
data = data.dropna()

# Encode categorical variables
data = pd.get_dummies(data, drop_first=True)

# Explore the data with some visualizations
sns.pairplot(data)
plt.show()

# Define the features (X) and the target (y)
X = data.drop('price', axis=1)  # Assuming 'PRICE' is the column name for
house prices
y = data['price']

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)

print(f"Training set size: {X_train.shape}")
print(f"Testing set size: {X_test.shape}")

# Initialize the Linear Regression model
model = LinearRegression()

# Train the model on the training data
model.fit(X_train, y_train)

# Print the coefficients of the model
print(f"Coefficients: {model.coef_}")
print(f"Intercept: {model.intercept_}")

# Make predictions on the testing data
y_pred = model.predict(X_test)

# Calculate the Mean Squared Error and R-squared value
```

```
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

print(f"Mean Squared Error: {mse}")
print(f"R-squared value: {r2}")

# Plot the predicted vs actual values
plt.scatter(y_test, y_pred)
plt.xlabel("Actual Prices")
plt.ylabel("Predicted Prices")
plt.title("Actual vs Predicted Prices")
plt.show()
```