DL LAB

CLASS:- CSAI-A
ROLL NO:- 71
PRN:- 12320206
NAME:- Vaishnavi Gosavi
ASSIGNMENT NO:-1

Title:- Implement Simple and Multiple Linear Regression to predict continuous variables.

1. Perform data preprocessing (handle missing values, feature scaling).
2. Fit a Simple Linear Regression model on a dataset (e.g., predicting house prices).
3. Extend to Multiple Linear Regression with multiple features.
4. Evaluate models using MSE, RMSE, and R^2 Score.
5. Visualize the regression line and predictions.

THEORY

Simple Linear Regression

Simple Linear Regression is used to establish a relationship between one independent variable (X) and a dependent variable (Y). The mathematical representation is:

$Y = b_0 + b_1X + \varepsilon$

Where:

- **Y** represents the dependent variable (output or target).
- **X** is the independent variable (input feature).
- $\mathbf{b_0}$ is the intercept, indicating the value of **Y** when **X = 0**.

- **b_1** is the slope, which shows how much **Y** changes with a one-unit increase in **X**.
- **ε** denotes the error term, which accounts for the difference between actual and predicted values.

For instance, in predicting house prices using only square footage, the model can be written as:

House Price = $b_0 + b_1$ (Square Footage) + ε

Multiple Linear Regression

Multiple Linear Regression is an extension of Simple Linear Regression, where multiple independent variables influence the dependent variable. The general form is:

$Y = b_0 + b_1X_1 + b_2X_2 + ... + b_nX_n + \varepsilon$

Here:

- **$X_1, X_2, ..., X_n$** represent different independent variables.
- **$b_1, b_2, ..., b_n$** are the coefficients that determine the contribution of each independent variable to **Y**.

For example, in house price prediction, we can incorporate multiple factors:

- **X_1** = Square Footage
- **X_2** = Number of Bedrooms
- **X_3** = Location Score

Thus, the model equation becomes:

House Price = $b_0 + b_1$ (Square Footage) + b_2 (Bedrooms) + b_3 (Location Score) + ε

ALGORITHM

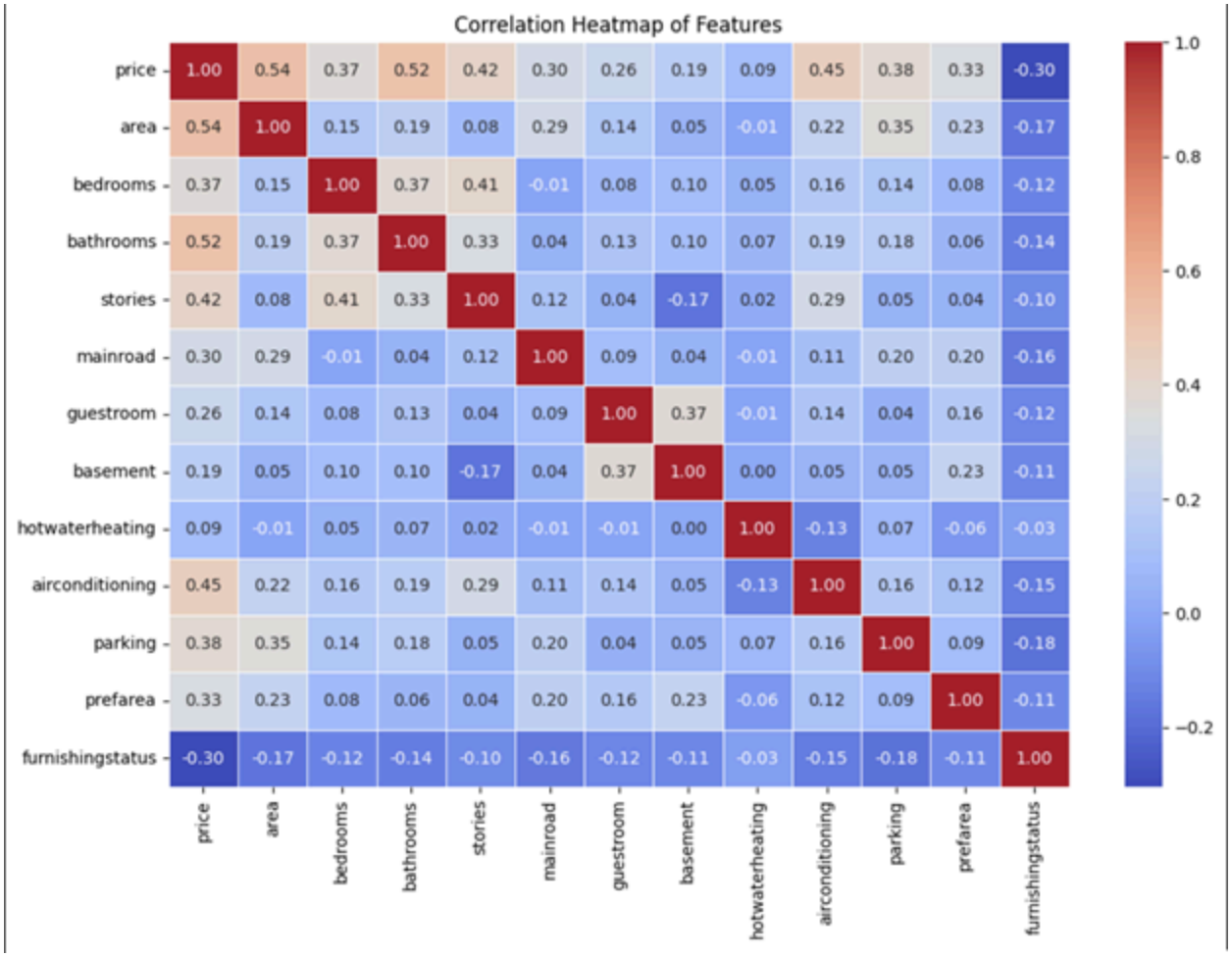
- Load the dataset (house prices with features like square footage, bedrooms, etc.).
- Preprocess the data (handle missing values, normalize if needed).
- Split into training and test sets.
- Train a linear regression model.
- Evaluate performance using metrics like Mean Squared Error (MSE),Root Mean Squared Error(RMSE) and R2.

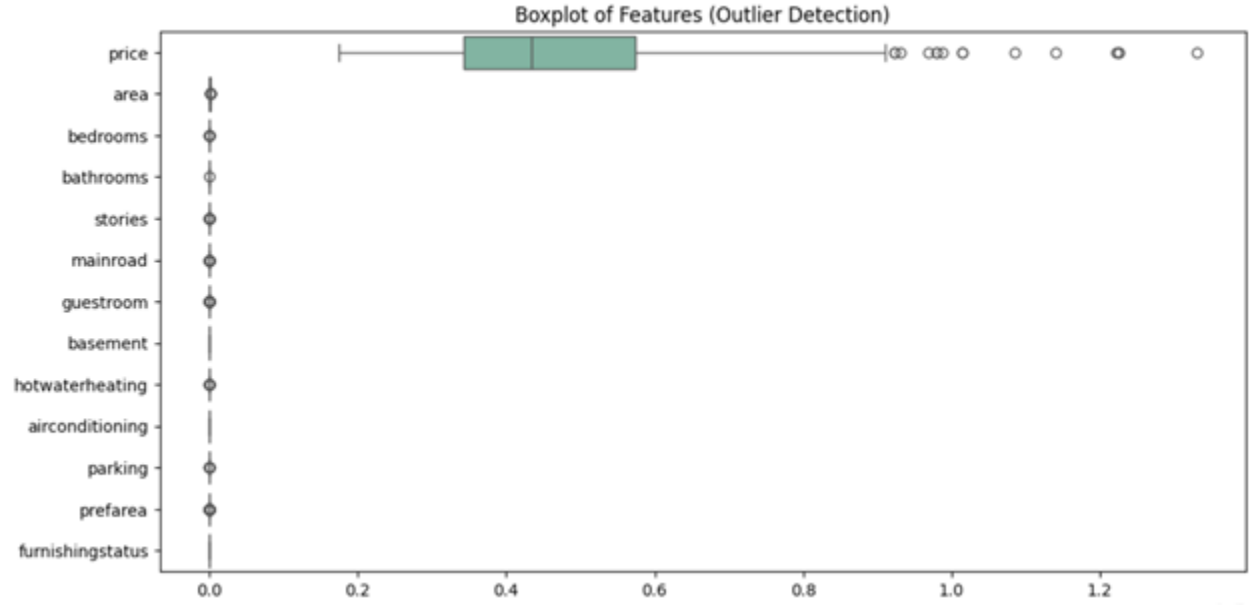
CODE

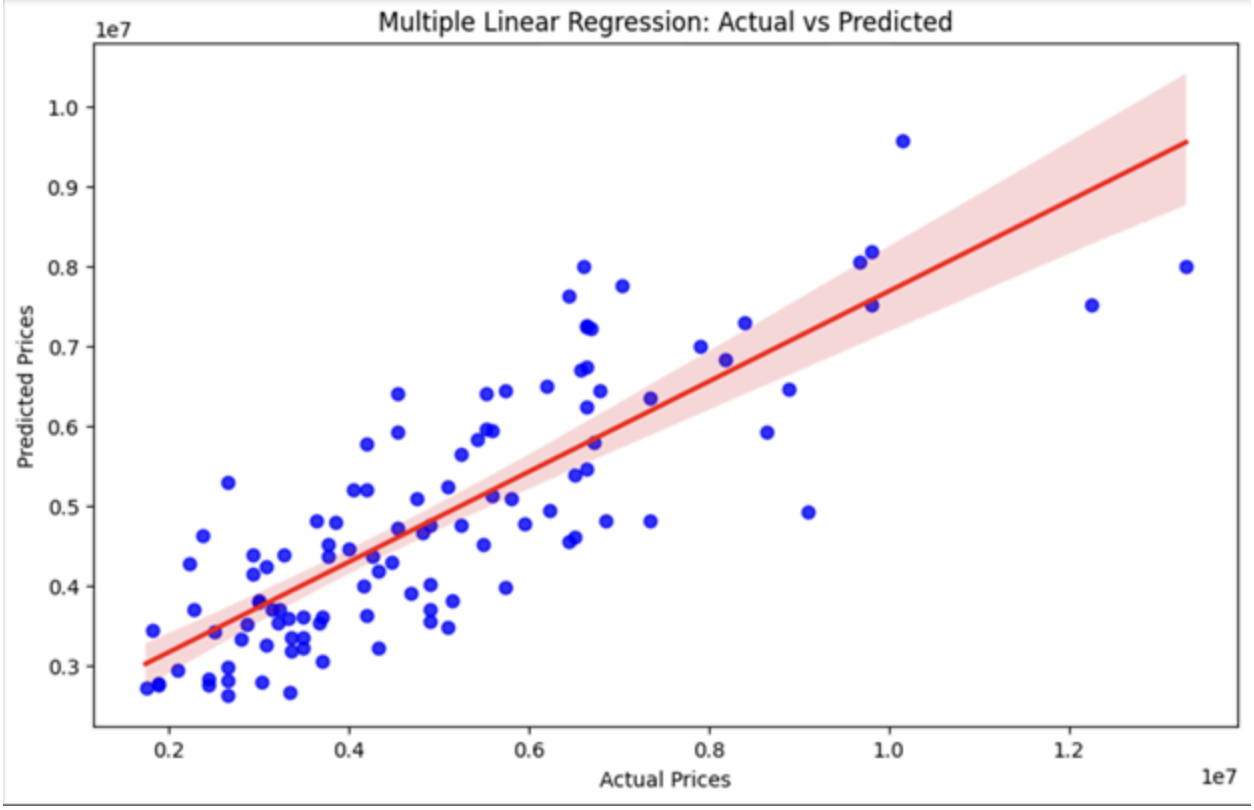
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler, LabelEncoder
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score

# Load dataset
df = pd.read_csv("Housing.csv")
print(df.head())
print("\nMissing values:\n", df.isnull().sum())

# Encode categorical variables
categorical_cols = df.select_dtypes(include=['object']).columns.tolist()
label_encoders = {}
for col in categorical_cols:
    le = LabelEncoder()
    df[col] = le.fit_transform(df[col])
    label_encoders[col] = le

# Define features and target variable
```

```
target = "price"
features = [col for col in df.columns if col != target]
X = df[features]
y = df[target]

# Split dataset
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Standardize features
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)

# Train simple linear regression model
single_feature = "area"
single_feature_idx = features.index(single_feature)
X_train_simple = X_train_scaled[:, single_feature_idx].reshape(-1, 1)
X_test_simple = X_test_scaled[:, single_feature_idx].reshape(-1, 1)

simple_model = LinearRegression()
simple_model.fit(X_train_simple, y_train)
y_pred_simple = simple_model.predict(X_test_simple)

# Train multiple linear regression model
multi_model = LinearRegression()
multi_model.fit(X_train_scaled, y_train)
y_pred_multi = multi_model.predict(X_test_scaled)

# Function to evaluate models
def evaluate_model(y_true, y_pred, model_name):
    mse = mean_squared_error(y_true, y_pred)
    rmse = np.sqrt(mse)
    r2 = r2_score(y_true, y_pred)
    print(f"\n{model_name} Model Performance:")
    print(f"Mean Squared Error (MSE): {mse:.2f}")
    print(f"Root Mean Squared Error (RMSE): {rmse:.2f}")
    print(f"R² Score: {r2:.4f}")

evaluate_model(y_test, y_pred_simple, "Simple Linear Regression")
evaluate_model(y_test, y_pred_multi, "Multiple Linear Regression")

# Visualization
plt.figure(figsize=(10, 5))
sns.heatmap(df.corr(), annot=True, cmap="magma", fmt=".2f", linewidths=0.5)
plt.title("Feature Correlation Heatmap")
plt.show()
```

```
fig = px.scatter_3d(df, x='area', y='bedrooms', z='price', color='price',
                    title="3D Scatter Plot: Area vs Bedrooms vs Price",
                    opacity=0.8)
fig.show()

plt.figure(figsize=(10, 5))
sns.violinplot(data=df, palette="pastel")
plt.title("Violin Plot of Features (Outlier Detection)")
plt.xticks(rotation=45)
plt.show()

sns.pairplot(df, diag_kind="kde", plot_kws={'alpha':0.5, 's':10})
plt.show()

plt.figure(figsize=(10, 5))
sns.scatterplot(x=X_test_simple.flatten(), y=y_test, color='blue', label='Actual')
sns.scatterplot(x=X_test_simple.flatten(), y=y_pred_simple, color='red', label='Predicted')
plt.plot(X_test_simple.flatten(), simple_model.predict(X_test_simple), color='black', linewidth=2)
plt.xlabel(single_feature)
plt.ylabel("Price")
plt.title("Simple Linear Regression: Actual vs Predicted")
plt.legend()
plt.show()

plt.figure(figsize=(10, 5))
sns.lmplot(x=y_test, y=y_pred_multi, aspect=1.5, scatter_kws={'color': "purple"}, line_kws={'color':
"green"})
plt.xlabel("Actual Prices")
plt.ylabel("Predicted Prices")
plt.title("Multiple Linear Regression: Actual vs Predicted")
plt.show()
```


Correlation Heatmap of Features
price 1.00 0.54 0.37 0.52 0.42 0.30 0.26 0.19 0.09 0.45 0.38 0.33 -0.30
area 0.54 1.00 0.15 0.19 0.08 0.29 0.14 0.05 -0.01 0.22 0.35 0.23 -0.17
bedrooms 0.37 0.15 1.00 0.37 0.41 -0.01 0.08 0.10 0.05 0.16 0.14 0.08 -0.12
bathrooms 0.52 0.19 0.37 1.00 0.33 0.04 0.13 0.10 0.07 0.19 0.18 0.06 -0.14
stories 0.42 0.08 0.41 0.33 1.00 0.12 0.04 -0.17 0.02 0.29 0.05 0.04 -0.10
mainroad 0.30 0.29 -0.01 0.04 0.12 1.00 0.09 0.04 -0.01 0.11 0.20 0.20 -0.16
guestroom 0.26 0.14 0.08 0.13 0.04 0.09 1.00 0.37 -0.01 0.14 0.04 0.16 -0.12
basement 0.19 0.05 0.10 0.10 -0.17 0.04 0.37 1.00 0.00 0.05 0.05 0.23 -0.11
hotwaterheating 0.09 -0.01 0.05 0.07 0.02 -0.01 -0.01 0.00 1.00 -0.13 0.07 -0.06 -0.03
airconditioning 0.45 0.22 0.16 0.19 0.29 0.11 0.14 0.05 -0.13 1.00 0.16 0.12 -0.15
parking 0.38 0.35 0.14 0.18 0.05 0.20 0.04 0.05 0.07 0.16 1.00 0.09 -0.18
prefarea 0.33 0.23 0.08 0.06 0.04 0.20 0.16 0.23 -0.06 0.12 0.09 1.00 -0.11
furnishingstatus -0.30 -0.17 -0.12 -0.14 -0.10 -0.16 -0.12 -0.11 -0.03 -0.15 -0.18 -0.11 1.00
price area bedrooms bathrooms stories mainroad guestroom basement hotwaterheating airconditioning parking prefarea furnishingstatus
1.0 0.8 0.6 0.4 0.2 0.0 −0.2


Boxplot of Features (Outlier Detection)
price
area
bedrooms
bathrooms
stories
mainroad
guestroom
basement
hotwaterheating
airconditioning
parking
prefarea
furnishingstatus
0.0 0.2 0.4 0.6 0.8 1.0 1.2


Multiple Linear Regression: Actual vs Predicted
1e7
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
Predicted Prices
0.2
0.4
0.6
0.8
1.0
1.2
1e7
Actual Prices