Experiment - 05

Aim: To implement Linear Regression and evaluate the performance evaluation metrics

Theory: Linear regression is one of the easiest and most popular Machine Learning algorithms. It is a statistical method that is used for predictive analysis. Linear regression makes predictions for continuous/real or numeric variables such as sales, salary, age, product price, etc.

Linear regression algorithm shows a linear relationship between a dependent (y) and one or more independent (y) variables, hence called linear regression. Since linear regression shows the linear relationship, which means it finds how the value of the dependent variable is changing according to the value of the independent variable.

There is one dependent or output variable which represents the Advertising data and is denoted by y. We want to build a linear relationship between these variables. This linear relationship can be modelled by mathematical equation of the form:- $Y = \beta0 + \beta1*X$ ---------- (1)

In this equation, X and Y are called independent and dependent variables respectively,

$\beta1$ is the coefficient for independent variable and

$\beta0$ is the constant term.

$\beta0$ and $\beta1$ are called parameters of the model.

For simplicity, we can compare the above equation with the basic line equation of the form:-

$y = ax + b$ ---------- (2)

We can see that

slope of the line is given by, $a = \beta1$, and

intercept of the line by $b = \beta0$.

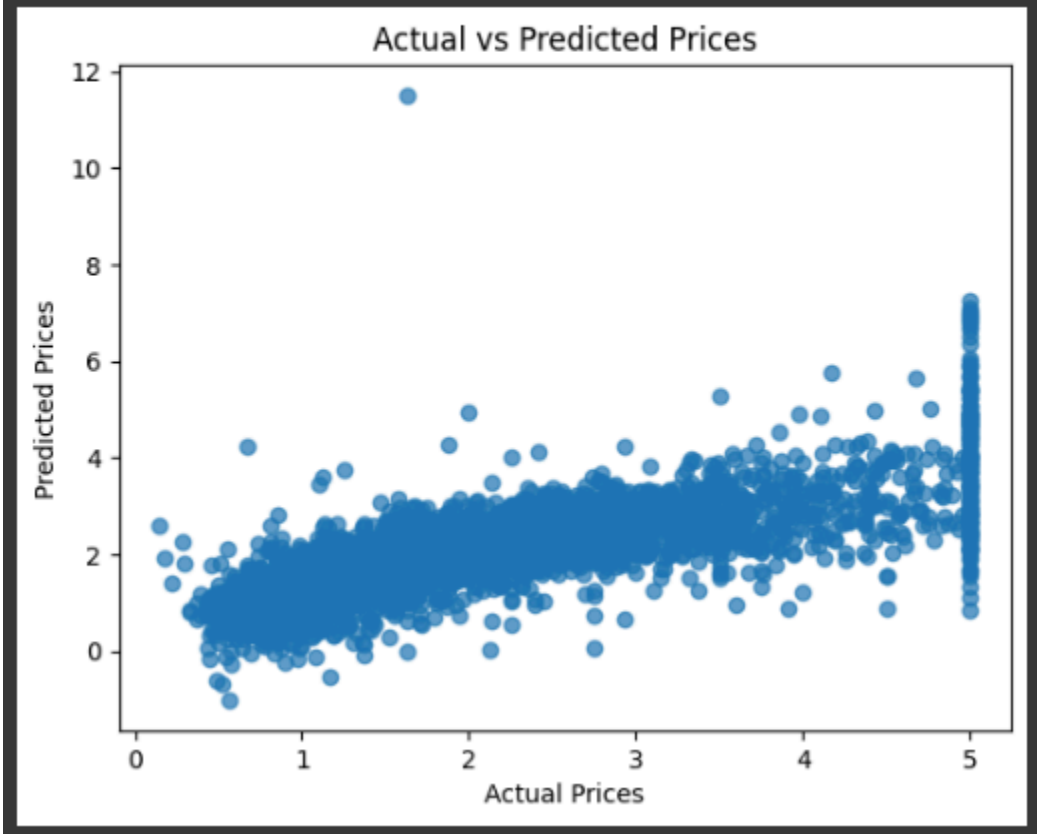
In this Simple Linear Regression model, we want to fit a line which estimates the linear relationship between X and Y. So, the question of fitting reduces to estimating the parameters of the model $\beta0$ and $\beta1$.

Code & Output:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_absolute_error, mean_squared_error,
r2_score
from sklearn.datasets import fetch_california_housing
from sklearn.preprocessing import MinMaxScaler  # Import MinMaxScaler

# Load California housing dataset
housing = fetch_california_housing()
df = pd.DataFrame(housing.data, columns=housing.feature_names)
df['PRICE'] = housing.target  # Add target variable

# Define features & target
X = df.drop(columns=['PRICE'])  # Features
y = df['PRICE']  # Target variable

# Split into train & test sets (80-20 split)
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=42)

# Create and fit MinMaxScaler
scaler = MinMaxScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)

# Train the model with scaled data
model = LinearRegression()
model.fit(X_train_scaled, y_train)

# Predictions with scaled data
y_pred = model.predict(X_test_scaled)

# Performance Metrics
mae = mean_absolute_error(y_test, y_pred)
mse = mean_squared_error(y_test, y_pred)
rmse = np.sqrt(mse)
r2 = r2_score(y_test, y_pred)

# Print results
print(f"Mean Absolute Error: {mae:.2f}")
print(f"Mean Squared Error: {mse:.2f}")
```

```
print(f"Root Mean Squared Error: {rmse:.2f}")
print(f"R² Score: {r2:.2f}")

# Plot actual vs predicted prices
plt.scatter(y_test, y_pred, alpha=0.7)
plt.xlabel("Actual Prices")
plt.ylabel("Predicted Prices")
plt.title("Actual vs Predicted Prices")
plt.show()
)
```

```
Mean Absolute Error: 0.53
Mean Squared Error: 0.56
Root Mean Squared Error: 0.75
R² Score: 0.58
```



Conclusion: Hence, we implemented Linear Regression and evaluated the performance evaluation metrics