Experiment-3

IMPLEMENT PROGRAM TO ANALYZE TIME SERIES DATA USING LINEAR REGRESSION

AIM:
TO WRITE A TO IMPLEMENT PROGRAM TO ANALYZE TIME SERIES DATA USING LINEAR REGRESSION

PROCEDURE:
1) Import necessary libraries.
2)Load the necessary libraries
3)Split the training and testing data
4)Train the linear regression model using the train data
5)Use the trained model to check the test data.
6)Use scatter plot and line plot to visualize the predicted and actual values

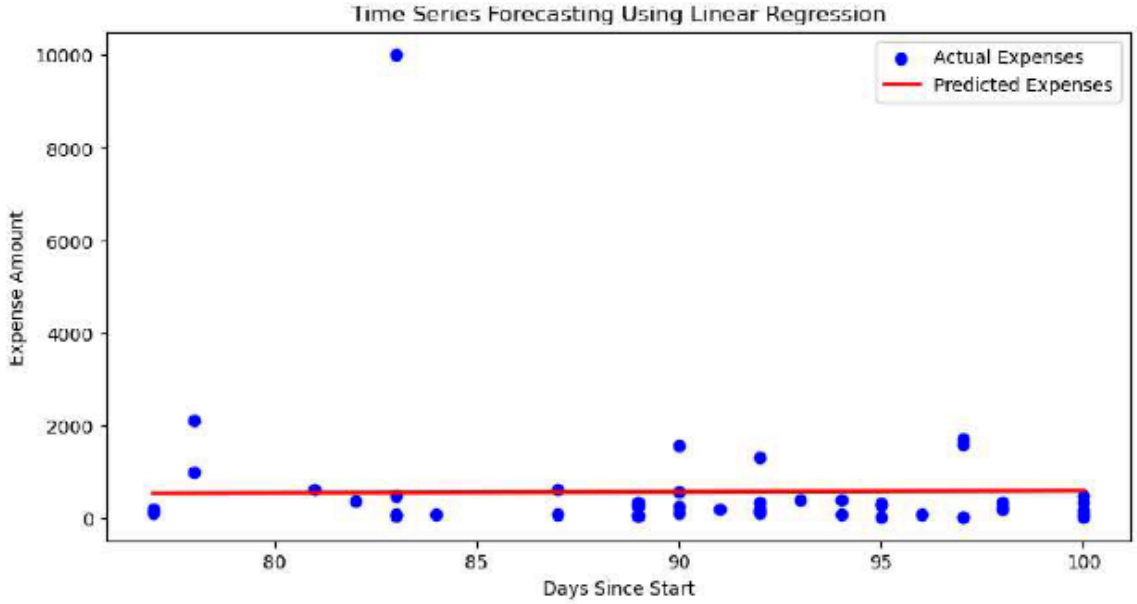
CODE:

```
# Importing libraries
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt

from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression

# Load the dataset
df = pd.read_csv("expense_data_1.csv")



# Convert 'Date' to datetime and drop rows with missing 'Date'
df['Date'] = pd.to_datetime(df['Date'], errors='coerce')
df = df.dropna(subset=['Date'])
```

```
# Group by 'Date' and sum the 'Amount'
df_grouped = df.groupby('Date')['Amount'].sum().reset_index()

# Calculate the number of days since the start date
df_grouped['Days'] = (df_grouped['Date'] - df_grouped['Date'].min()).dt.days

# Prepare training data
X = df_grouped[['Days']]
y = df_grouped['Amount']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42,
shuffle=False)

# Train the linear regression model
model = LinearRegression()
model.fit(X_train, y_train)

# Make predictions
y_pred = model.predict(X_test)

# Plot the actual vs predicted expenses
plt.figure(figsize=(10, 5))
plt.scatter(X_test, y_test, color='blue', label='Actual Expenses')
plt.plot(X_test, y_pred, color='red', label='Predicted Expenses', linewidth=2)
plt.xlabel("Days Since Start")
plt.ylabel("Expense Amount")
plt.title("Time Series Forecasting Using Linear Regression")
plt.legend()
plt.show()

# Predict future expenses for the next 30 days
days_future = np.array([[df_grouped['Days'].max() + i] for i in range(1, 31)])
future_predictions = model.predict(days_future)

# Generate future dates
future_dates = pd.date_range(start=df_grouped['Date'].max(), periods=30, freq='D')
future_df = pd.DataFrame({'Date': future_dates, 'Predicted_Expense': future_predictions})
print(future_df)
```

OUTPUT:

	Date	Predicted_Expense
0	2022-03-02 10:11:00	602.515742
1	2022-03-03 10:11:00	605.065485
2	2022-03-04 10:11:00	607.615228
3	2022-03-05 10:11:00	610.164971
4	2022-03-06 10:11:00	612.714714
5	2022-03-07 10:11:00	615.264457
6	2022-03-08 10:11:00	617.814199
7	2022-03-09 10:11:00	620.363942
8	2022-03-10 10:11:00	622.913685
9	2022-03-11 10:11:00	625.463428
10	2022-03-12 10:11:00	628.013171
11	2022-03-13 10:11:00	630.562913
12	2022-03-14 10:11:00	633.112656
13	2022-03-15 10:11:00	635.662399
14	2022-03-16 10:11:00	638.212142
15	2022-03-17 10:11:00	640.761885
16	2022-03-18 10:11:00	643.311628
17	2022-03-19 10:11:00	645.861370
18	2022-03-20 10:11:00	648.411113
19	2022-03-21 10:11:00	650.960856
20	2022-03-22 10:11:00	653.510599
21	2022-03-23 10:11:00	656.060342
22	2022-03-24 10:11:00	658.610085
23	2022-03-25 10:11:00	661.159827
24	2022-03-26 10:11:00	663.709570
25	2022-03-27 10:11:00	666.259313
26	2022-03-28 10:11:00	668.809056
27	2022-03-29 10:11:00	671.358799
28	2022-03-30 10:11:00	673.908542
29	2022-03-31 10:11:00	676.458284



RESULT:
The program to implementing linear regression for time series forecasting is done