5.Develop a linear regression model for forecasting time series data

AIM:

To develop a **Linear Regression model** to **forecast future values** based on historical **time series** data from an expenses dataset.

PROCEDURE:

1) Import libraries (like pandas, matplotlib, sklearn, etc.)
2) Load the dataset (expenses.csv).
3) Explore the dataset (check columns, types, missing values).
4) Convert date column (if present) to datetime type.
5) Set date as index (common practice in time series).
6) Feature Engineering: Create a numerical time variable (for linear regression).
7) Split the data into training and testing sets.
8) Train the Linear Regression model.
9) Predict on test data.
10) Visualize the results (Actual vs Predicted).
11) Forecast future points.

CODE:

```
# Step 1: Import Libraries

import pandas as pd

import matplotlib.pyplot as plt

from sklearn.linear_model import LinearRegression

from sklearn.metrics import mean_squared_error


# Step 2: Load Dataset

data = pd.read_csv('/mnt/data/expenses.csv')


# Step 3: Explore Dataset

print(data.head())
```

```
print(data.info())

# Step 4: Convert 'Date' column to datetime
data['Date'] = pd.to_datetime(data['Date'])

# Step 5: Set Date as Index
data.set_index('Date', inplace=True)

# Step 6: Feature Engineering
# Create a numerical feature for time
data['Time'] = range(len(data))

# Step 7: Prepare features and target
X = data[['Time']]
y = data['Expenses']

# Step 8: Train-Test Split (80% train, 20% test)
split = int(0.8 * len(data))
X_train, X_test = X[:split], X[split:]
y_train, y_test = y[:split], y[split:]

# Step 9: Train the Linear Regression Model
model = LinearRegression()
model.fit(X_train, y_train)

# Step 10: Predict
y_pred = model.predict(X_test)

# Step 11: Evaluate the Model
```

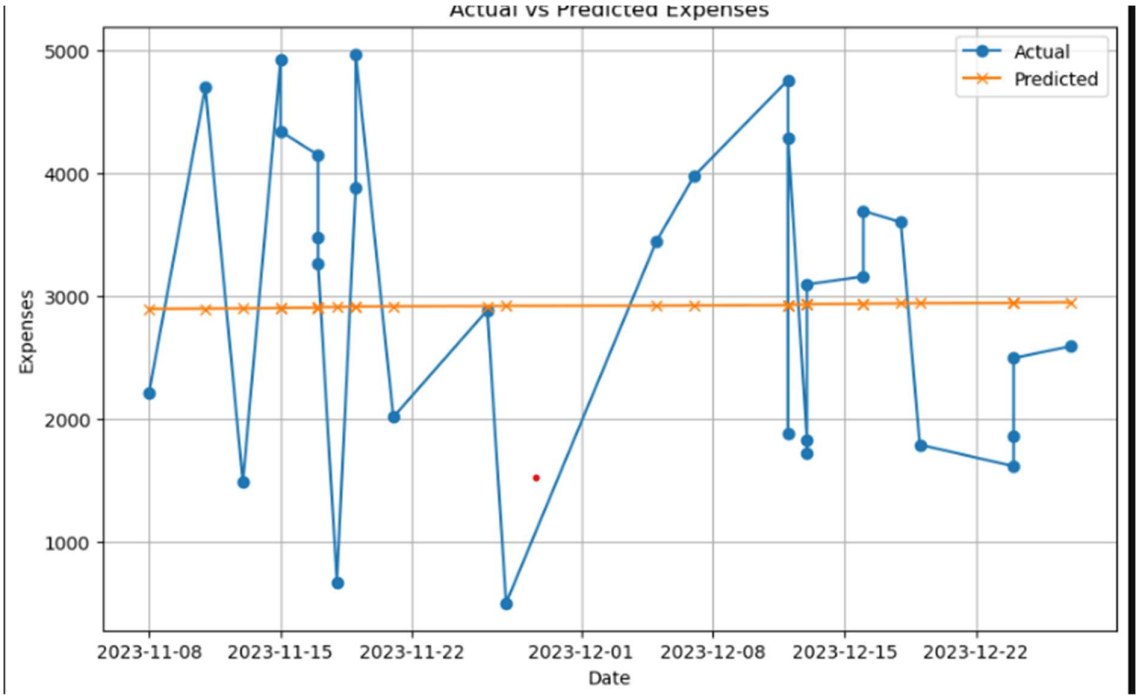
```
mse = mean_squared_error(y_test, y_pred)
print(f"Mean Squared Error: {mse}")

# Step 12: Plot Actual vs Predicted
plt.figure(figsize=(10,6))
plt.plot(data.index[split:], y_test, label='Actual', marker='o')
plt.plot(data.index[split:], y_pred, label='Predicted', marker='x')
plt.title('Actual vs Predicted Expenses')
plt.xlabel('Date')
plt.ylabel('Expenses')
plt.legend()
plt.grid(True)
plt.show()

# Step 13: Forecast Future
# Forecast next 10 time points
future_time = pd.DataFrame({'Time': range(len(data), len(data)+10)})
future_pred = model.predict(future_time)

# Display forecasted values
print("Forecasted Future Expenses:")
print(future_pred)
```

OUTPUT:



RESULT:

The program to develop a linear regression model is implemented successfully.