9. Neural network-based time series forecasting model

Aim:

To develop neural network-based time series forecasting model.

Procedure:

1. Importing Libraries

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.preprocessing import MinMaxScaler
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import LSTM, Dense
```

2. Load and Prepare the Dataset

```
df = pd.read_csv('/content/supermarket_sales - Sheet1.csv')
df['Date'] = pd.to_datetime(df['Date'])
df.sort_values('Date', inplace=True)
```

3. Aggregate Daily Sales

```
daily_sales = df.groupby('Date')['Total'].sum().reset_index()
```

4. Resample Data to Daily Frequency

```
daily_sales = daily_sales.set_index('Date').resample('D').sum().fillna(0)
```

5. Normalize the Sales Data

```
scaler = MinMaxScaler()
scaled = scaler.fit_transform(daily_sales)
```

6. Create Sequences for LSTM

```
def create_dataset(data, time_steps=7):
```

```
    X, y = [], []
    for i in range(len(data) - time_steps):
        X.append(data[i:i + time_steps])
        y.append(data[i + time_steps])
    return np.array(X), np.array(y)
```

7. Prepare Input for the Model

```
X = X.reshape((X.shape[0], X.shape[1], 1))
```

8. Train-Test Split

```
split = int(0.8 * len(X))
X_train, X_test = X[:split], X[split:]
y_train, y_test = y[:split], y[split:]
```

9. Build and Train the LSTM Model

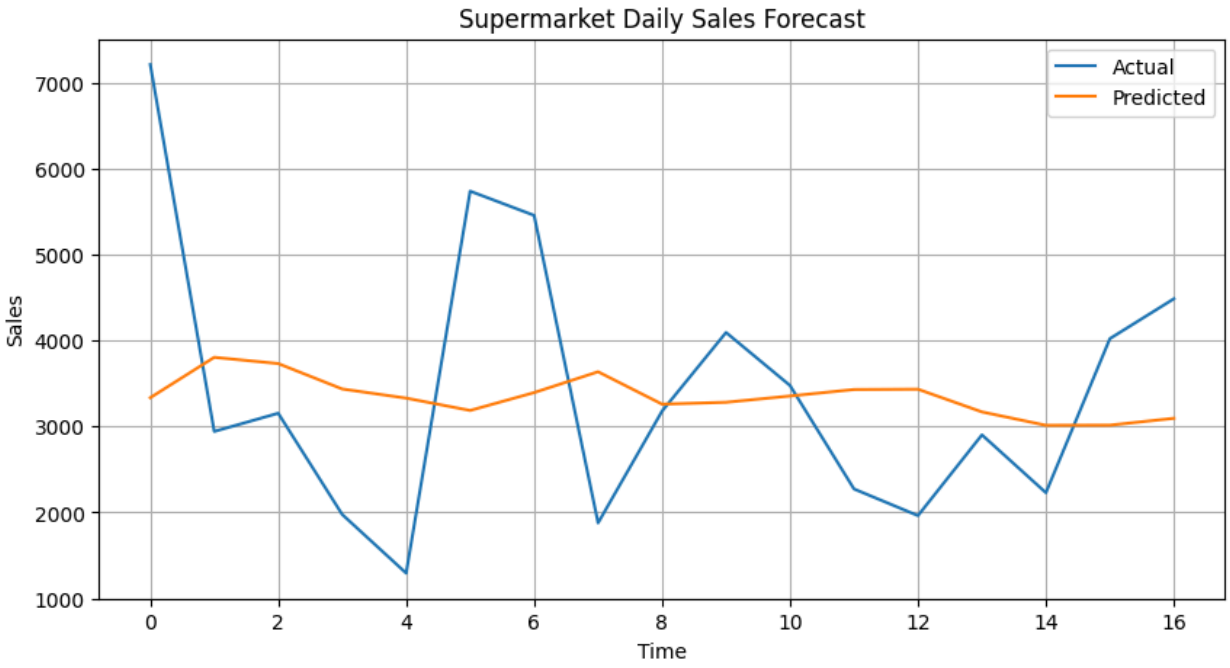
```
model = Sequential([
    LSTM(64, activation='relu', input_shape=(time_steps, 1)),
    Dense(1)
])
model.compile(optimizer='adam', loss='mse')
model.fit(X_train, y_train, epochs=30, validation_data=(X_test, y_test),
verbose=1)
```

10. Plot the Results

```
plt.figure(figsize=(10, 5))
plt.plot(y_test_inv, label="Actual")
plt.plot(y_pred_inv, label="Predicted")
plt.title("Supermarket Daily Sales Forecast")
plt.xlabel("Time")
plt.ylabel("Sales")
plt.legend()
plt.grid()
```

plt.show()

Output



Result:

Thus the program has been successfully implemented and verified.