**Exp:9**

**24.04.2025**

**Develop neural network-based time series forecasting model**

**Aim:**

To develop a python program Developing a neural network-based time series forecasting model

**1. Importing Required Libraries**

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

from sklearn.preprocessing import MinMaxScaler

from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import LSTM, Dense

**Explanation:**

We import numpy (np) is used for numerical operations, pandas (pd) for data manipulation, matplotlib.pyplot (plt) for plotting.

Adfuller for stationary check

**2. Loading the Dataset**

file\_path = '/content/gold (1).

data = pd.read\_csv(file\_path)

**Explanation:**

We use pd.read\_csv() to load a CSV file containing Gold data.

**3. Display the first few rows to understand the structure**

print(data.head())

print(data.info())

**4.** **Preprocessing the data**

### data.fillna(method='ffill', inplace=True)

### 5. Normalization

scaler = MinMaxScaler(feature\_range=(0, 1))

scaled\_values = scaler.fit\_transform(values)

**6.** **Create Sequences for LSTM**

sequence\_length = 60  # Use the last 60 days to predict the next value

X, y = [], []

for i in range(sequence\_length, len(scaled\_values)):

    X.append(scaled\_values[i-sequence\_length:i, 0])

    y.append(scaled\_values[i, 0])

X, y = np.array(X), np.array(y)

**7.** **Visualization**

actual\_prices = scaler.inverse\_transform(y\_test.reshape(-1, 1))

plt.figure(figsize=(10, 6))

plt.plot(actual\_prices, label='Actual Prices', color='blue')

plt.plot(predictions, label='Predicted Prices', color='red')

plt.title('USD (AM) Prices Prediction')

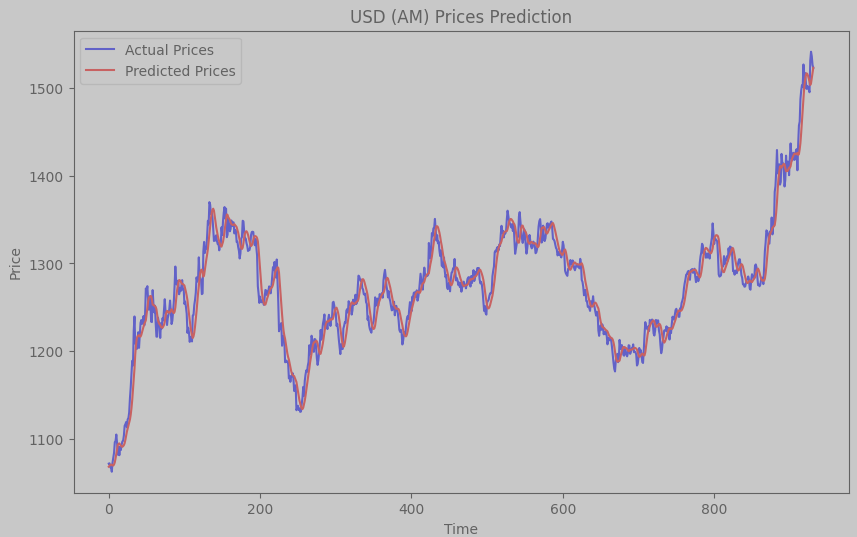
plt.xlabel('Time')

plt.ylabel('Price')

plt.legend()

plt.show()

**Outputs:**

****

**Result:**

Thus the Program for creating an ARIMA model for forecasting has been implemented successfully.