10.Develop neural network based time series forecasting model

AIM:

To implement neural network based time series forecasting model.

PROCEDURE:

1.Import the necessary libraries:

```
import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

from statsmodels.tsa.api import VAR

from sklearn.metrics import mean_squared_error

from google.colab import files

import io
```

2.Load dataset:

```
df = pd.read_csv('PRICE_AND_DEMAND_201801_NSW1.csv')

df['SETTLEMENTDATE'] = pd.to_datetime(df['SETTLEMENTDATE'], format='%Y/%m/%d %H:%M:%S')

df.set_index('SETTLEMENTDATE', inplace=True)
```

3. Plot original data scaler = StandardScaler()

```
data_scaled = scaler.fit_transform(data)

data_scaled = pd.DataFrame(data_scaled, columns=data.columns, index=data.index)

data.plot(title="Original Multivariate Time Series", figsize=(12, 5))

plt.grid(True)

plt.show()
```

4. Split into train and test

```
n_obs = 24 * 7  # last week for test
train, test = data_diff[:-n_obs], data_diff[-n_obs:]
```

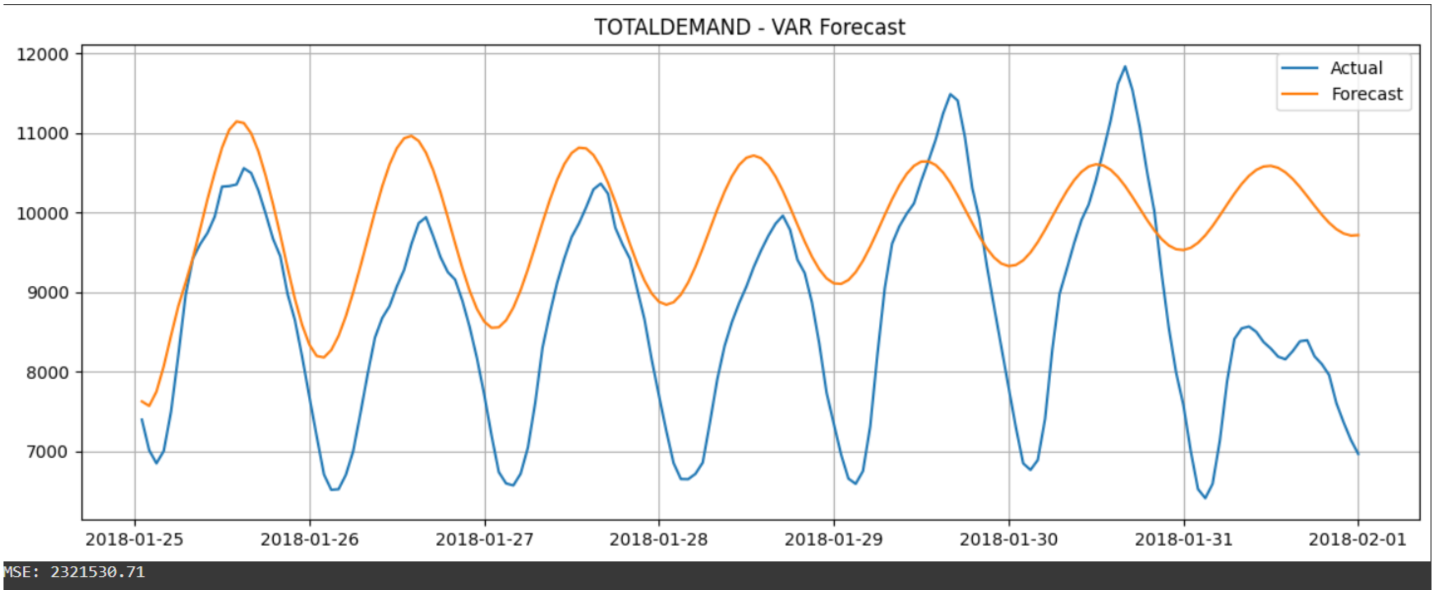
5. Fit VAR model

```
model = VAR(train)
lag_order = model.select_order(maxlags=15)
print("Optimal lag order:", lag_order.summary())
model_fitted = model.fit(lag_order.aic)
```

6. Plot actual vs forecast

```
plt.figure(figsize=(14, 5))
plt.plot(data['TOTALDEMAND'].iloc[-n_obs:], label='Actual')
plt.plot(forecast_values['TOTALDEMAND'], label='Forecast')
plt.title("TOTALDEMAND - VAR Forecast")
plt.legend()
plt.grid(True)
plt.show()
```

OUTPUT:



```
MSE: 2321530.71
```

RESULT:

Thus the program has been executed and implemented successfully.