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

**AIM:**

To develop a neural network based time series forecasting modelusing jupyter notebook.

**ALGORITHM:**

1. Load the taxi dataset then clean and load the values

2. Train a neural network to predict the next hour's total\_amount using the previous 24 hours as input features.

3. Plot the graph and visualize the values

**CODE:**

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

from sklearn.preprocessing import MinMaxScaler

from sklearn.neural\_network import MLPRegressor

from sklearn.metrics import mean\_squared\_error

# Load data

df = pd.read\_csv("dataset.csv")

# Create datetime column

df['datetime'] = pd.to\_datetime(df[['year', 'month', 'day']]) + pd.to\_timedelta(df['hour\_of\_day'], unit='h')

# Aggregate total\_amount per hour

ts\_df = df.groupby('datetime')['total\_amount'].sum().reset\_index().sort\_values('datetime')

# Normalize total\_amount

scaler = MinMaxScaler()

ts\_df['scaled'] = scaler.fit\_transform(ts\_df[['total\_amount']])

# Create sequences

def create\_sequences(data, window\_size=24):

X, y = [], []

for i in range(window\_size, len(data)):

X.append(data[i-window\_size:i])

y.append(data[i])

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

X, y = create\_sequences(ts\_df['scaled'].values, window\_size=24)

# Train/test split

split = int(len(X) \* 0.8)

X\_train, X\_test = X[:split], X[split:]

y\_train, y\_test = y[:split], y[split:]

# Create and train neural network model

model = MLPRegressor(hidden\_layer\_sizes=(64, 32), activation='relu', solver='adam', max\_iter=500, random\_state=42)

model.fit(X\_train, y\_train)

# Predict

predictions = model.predict(X\_test)

# Inverse transform predictions

predicted\_values = scaler.inverse\_transform(predictions.reshape(-1, 1))

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

# Plot results

plt.figure(figsize=(12, 6))

plt.plot(actual\_values, label='Actual')

plt.plot(predicted\_values, label='Predicted')

plt.title('Neural Network Forecast (MLPRegressor)')

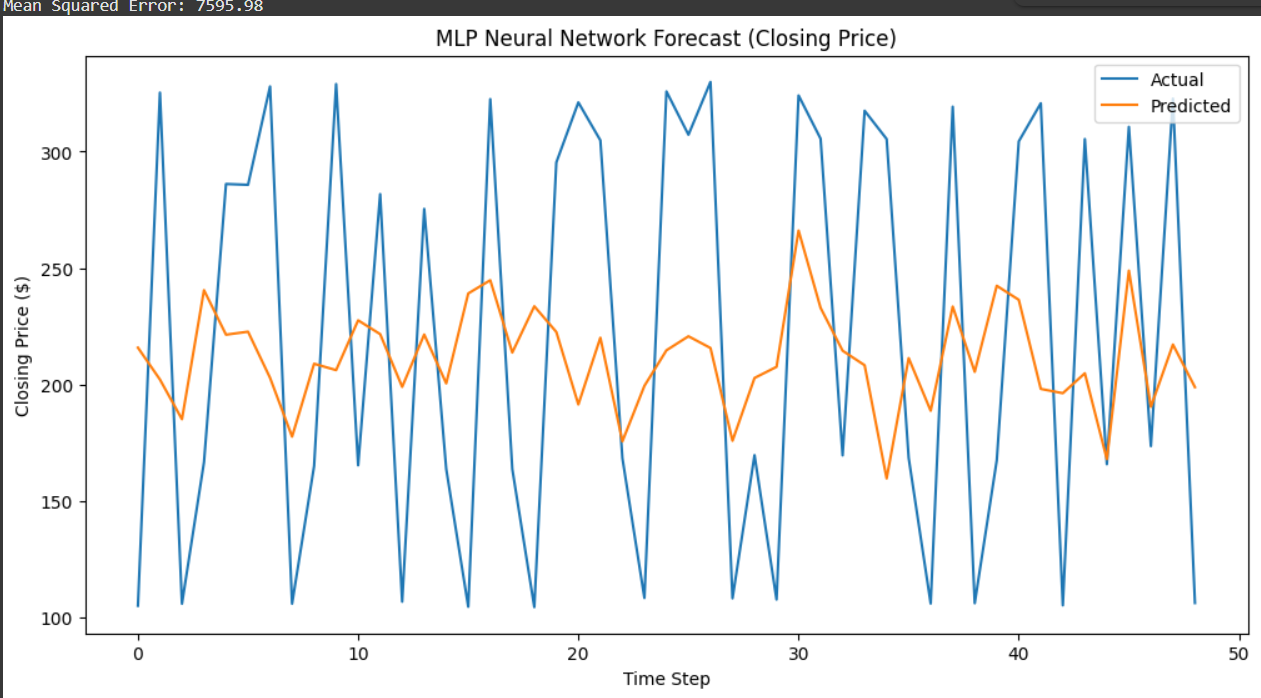
plt.xlabel('Time Step (Hour)')

plt.ylabel('Total Amount')

plt.legend()

plt.show()

**OUTPUT:**

****

**RESULT:**

ToDevelop a neural network based time series forecasting model completed

successfully and the output is verified.