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

**AIM:**

To Develop a neural network based time series forecasting using autism dataset

**PROCEDURE:**

**1.Import the necessary 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

from sklearn.model\_selection import train\_test\_split

from pandas.plotting import register\_matplotlib\_converters

register\_matplotlib\_converters()

**2.Load dataset**

file\_path = 'autism\_screening.csv' # Update the filename if needed

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

**3.Simulate a time-series index (assuming daily records)**

df['Date'] = pd.date\_range(start='2024-01-01', periods=len(df), freq='D')

df.set\_index('Date', inplace=True)

**4. Check for and handle zero/negative values**

if (series <= 0).any():

min\_value = series.min()# Shift the series to be positive

if min\_value <= 0:

series = series + abs(min\_value) + 1 # Add 1 to ensure all values are strictly positive

**5. Visualize the data**

monthly\_positive.plot(title="Monthly Positive Autism Screenings")

plt.xlabel("Month")

plt.ylabel("Number of Positive Screenings")

plt.show()

**6.** **Prepare the data for LSTM**

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

scaled\_data = scaler.fit\_transform(monthly\_positive.values.reshape(-1, 1))

def create\_dataset(data, time\_step=1):

X, y = [], []

for i in range(len(data) - time\_step - 1):

X.append(data[i:(i + time\_step), 0])

y.append(data[i + time\_step, 0])

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

X, y = create\_dataset(scaled\_data, time\_step)

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

**7.** **Split the data into training and testing sets**

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, shuffle=False)

**8.** **Build the LSTM model04**

model = Sequential()

model.add(LSTM(units=50, return\_sequences=True, input\_shape=(time\_step, 1)))

model.add(LSTM(units=50, return\_sequences=False))

model.add(Dense(units=1))

**9. Compile and train the model**

model.compile(optimizer='adam', loss='mean\_squared\_error')

history = model.fit(X\_train, y\_train, epochs=20, batch\_size=32, validation\_data=(X\_test, y\_test))

**10. Plot training & validation loss**

plt.plot(history.history['loss'], label='train')

plt.plot(history.history['val\_loss'], label='test')

plt.title("Model Loss")

plt.xlabel('Epochs')

plt.ylabel('Loss')

plt.legend()

plt.show()

**11. Plot the actual vs predicted values**

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

plt.plot(y\_test\_actual, color='blue', label="Actual Positive Screenings")

plt.plot(predictions, color='red', label="Predicted Positive Screenings")

plt.title("Actual vs Predicted Positive Screenings")

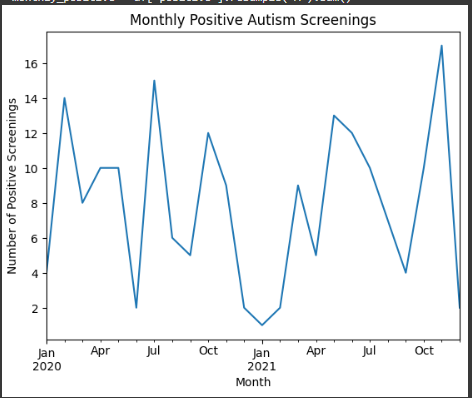
plt.xlabel("Month")

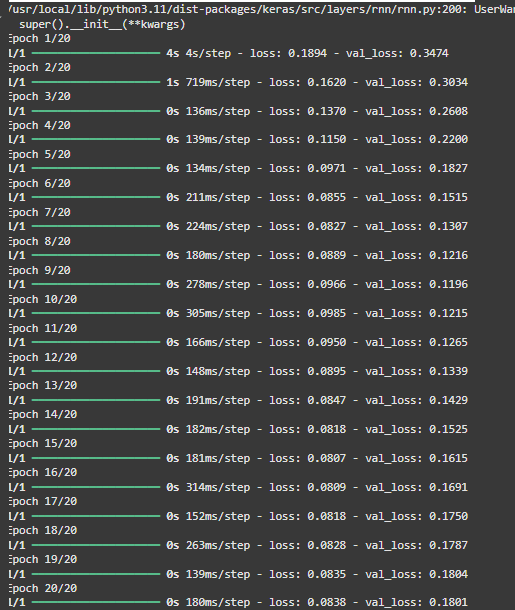
plt.ylabel("Number of Positive Screenings")

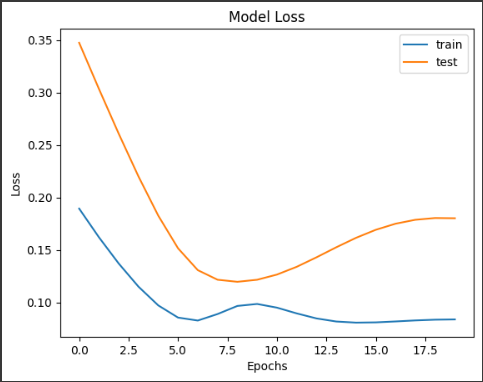
plt.legend()

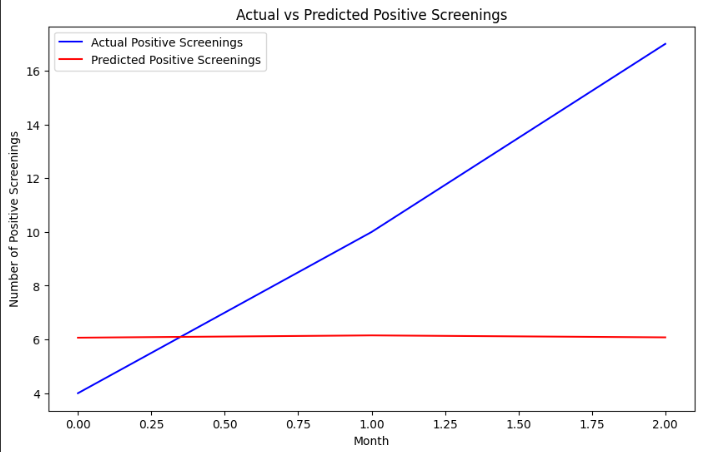
plt.show()

**OUTPUT :**

****



****

****

**RESULT:**

To Develop a neural network based time series forecasting(LSTM) using autism dataset on the autism screening dataset has been implemented successfully.