**Source code:**

from google.colab import files

uploaded = files.upload()

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, Dropout

from datetime import datetime

from google.colab import files

# Step 1: Load and inspect data

def load\_groceries\_data(filename):

df = pd.read\_csv(filename)

print("Columns:", df.columns.tolist())

df['Date'] = pd.to\_datetime(df['Date'])

return df

# Step 2: Aggregate monthly item counts

def aggregate\_monthly\_sales(df):

df\_monthly = df.groupby(['year', 'month'])['itemDescription'].count().reset\_index()

df\_monthly['Date'] = pd.to\_datetime(df\_monthly[['year', 'month']].assign(day=1))

df\_monthly = df\_monthly[['Date', 'itemDescription']].set\_index('Date')

return df\_monthly

# Step 3: Preprocess for LSTM

def preprocess\_data(series, time\_step=12):

scaler = MinMaxScaler()

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

X, y = [], []

for i in range(time\_step, len(scaled\_data)):

X.append(scaled\_data[i-time\_step:i, 0])

y.append(scaled\_data[i, 0])

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

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

return X, y, scaler

# Step 4: Build LSTM model

def build\_model(input\_shape):

model = Sequential()

model.add(LSTM(50, return\_sequences=True, input\_shape=input\_shape))

model.add(Dropout(0.2))

model.add(LSTM(50))

model.add(Dropout(0.2))

model.add(Dense(25))

model.add(Dense(1))

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

return model

# Step 5: Plot results

def plot\_predictions(dates, actual, predicted):

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

plt.plot(dates, actual, label='Actual Sales')

plt.plot(dates, predicted, label='Predicted Sales')

plt.xlabel('Date')

plt.ylabel('Number of Items Sold')

plt.title('Grocery Sales Prediction')

plt.legend()

plt.grid(True)

plt.show()

# Step 6: Upload and run pipeline

def main():

uploaded = files.upload()

filename = list(uploaded.keys())[0]

df = load\_groceries\_data(filename)

df\_monthly = aggregate\_monthly\_sales(df)

X, y, scaler = preprocess\_data(df\_monthly['itemDescription'])

model = build\_model((X.shape[1], 1))

model.fit(X, y, epochs=10, batch\_size=4)

predictions = model.predict(X)

predictions\_rescaled = scaler.inverse\_transform(predictions)

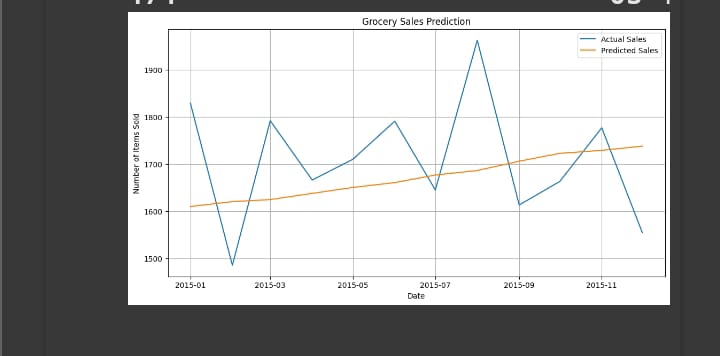
actual\_rescaled = scaler.inverse\_transform(y.reshape(-1, 1))

dates = df\_monthly.index[-len(predictions):]

plot\_predictions(dates, actual\_rescaled, predictions\_rescaled)

main()

**Output:**

****