**EX:No.4 LINEAR REGRESSION 221501040 21/01/25**

**AIM :** To implement linear regression on a time series data .

**ALGORITHM :**

Step 1: Load the sales data from a CSV file and set 'Order Date' as the index.

Step 2: Calculate daily percentage returns from the Sales column and remove missing or infinite values.

Step 3: Create a new column called Lagged\_Return by shifting the return values by one day, then drop any resulting NaNs.

Step 4: Split the cleaned data into training and testing sets using an 80/20 ratio.

Step 5: Train a Linear Regression model using the Lagged\_Return to predict the actual Return.

Step 6: Predict on the test set and plot the first 20 actual vs predicted return values for comparison.

**IMPLEMENTATION :**

import numpy as np

import pandas as pd

from pathlib import Path

import matplotlib.pyplot as plt

from sklearn.linear\_model import LinearRegression

# Enable inline plots (for Jupyter or Colab)

%matplotlib inline

# Load your dataset

sales\_df = pd.read\_csv(

Path("cleaned\_sales\_data.csv"), index\_col="Order Date", parse\_dates=True

)

# Calculate percentage returns for 'Sales'

sales\_df['Return'] = sales\_df['Sales'].pct\_change() \* 100

# Replace inf/-inf with NaN and drop all NaNs

returns = sales\_df.replace([np.inf, -np.inf], np.nan).dropna()

# Create lagged return

returns['Lagged\_Return'] = returns["Return"].shift()

returns = returns.dropna()

# Print latest date in data to verify

print("Latest date in dataset:", returns.index.max())

# Split into train/test using 80/20 index-based split

split\_idx = int(len(returns) \* 0.8)

train = returns.iloc[:split\_idx]

test = returns.iloc[split\_idx:]

# Create X and y sets

x\_train = train[["Lagged\_Return"]]

y\_train = train["Return"]

x\_test = test[["Lagged\_Return"]]

y\_test = test["Return"]

# Fit Linear Regression Model

model = LinearRegression()

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

# Make predictions

predictions = model.predict(x\_test)

# Assemble results

results = y\_test.to\_frame()

results["Predicted Return"] = predictions

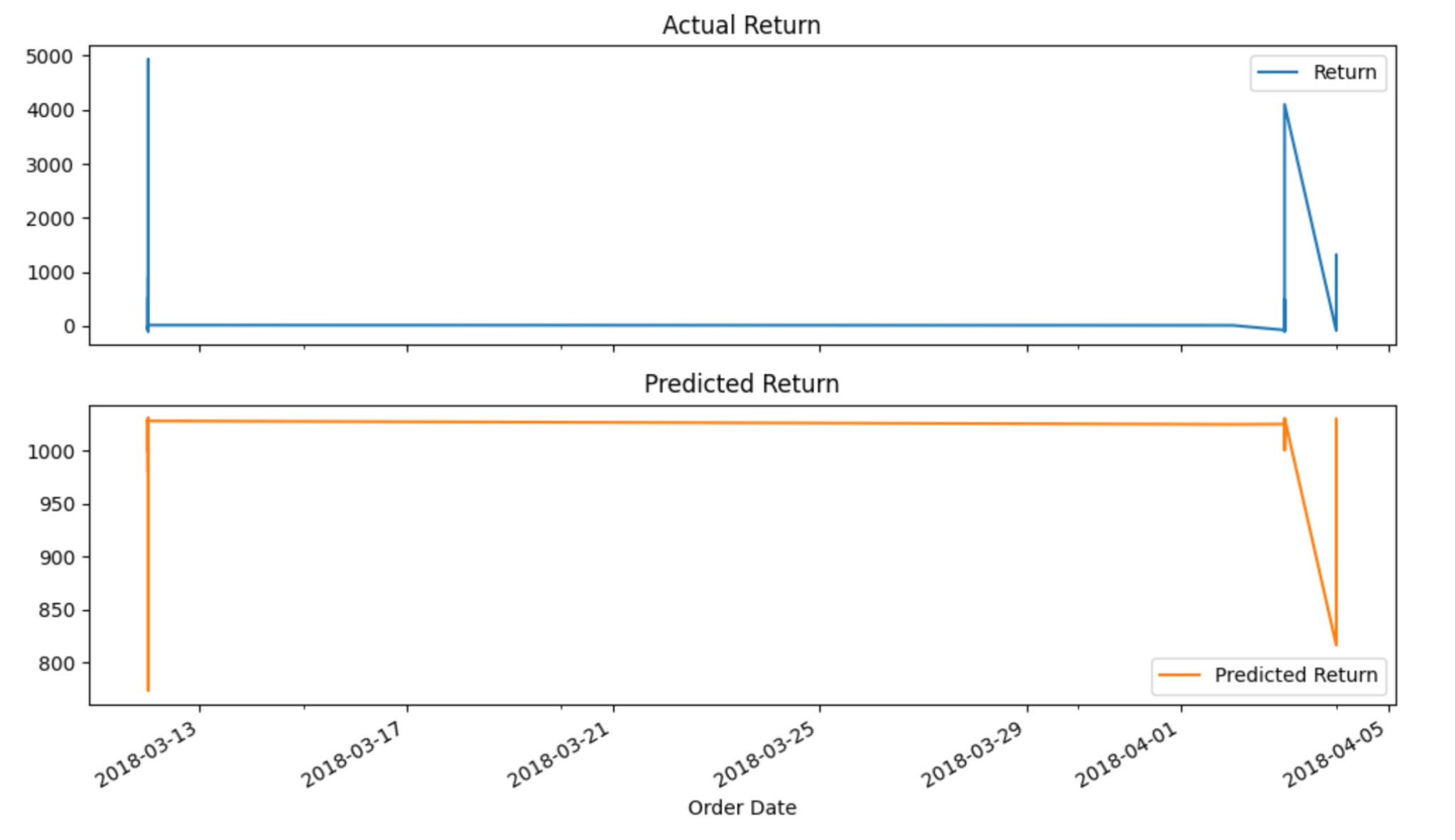
# Plot the first 20 predictions vs actual returns

results[:20].plot(subplots=True, figsize=(10, 6), title=["Actual Return", "Predicted Return"])

plt.tight\_layout()

plt.show()

**OUTPUT:**

****

**RESULT :** Thus linear regression has been implemented on a time series data.