We will use Arima Model for **DEMAND FORECASTING**

First, you need to have the **stats models** library installed. You can install it via pip if you haven't already:

pip install statsmodels

Here's a step-by-step guide to implementing an ARIMA model:

1. **Import necessary libraries**

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

from statsmodels.tsa.arima.model import ARIMA

1. **Load your time series data:**

# Load your time series data into a pandas DataFrame

data = pd.read\_csv('your\_data.csv', index\_col='date\_column', parse\_dates=True)

1. **Visualize your data:**

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

plt.plot(data)

plt.title('Time Series Data')

plt.xlabel('Date')

plt.ylabel('Value')

plt.show()

1. **Check for stationarity:**

ARIMA assumes that the time series is stationary. You can check for stationarity using techniques like the Dickey-Fuller test or visually inspecting the plot of the data.

from statsmodels.tsa.stattools import adfuller

def test\_stationarity(timeseries):

# Perform Dickey-Fuller test

dftest = adfuller(timeseries, autolag='AIC')

dfoutput = pd.Series(dftest[0:4], index=['Test Statistic','p-value','#Lags Used','Number of Observations Used'])

for key,value in dftest[4].items():

dfoutput['Critical Value (%s)'%key] = value

print(dfoutput)

test\_stationarity(data['your\_column'])

1. **Make the data stationary:**

If the data is not stationary, you can make it stationary by differencing.

# Take first difference

data\_diff = data.diff().dropna()

**6. Determine ARIMA parameters (p, d, q):**

p is the order of the AutoRegressive term.

d is the degree of differencing.

q is the order of the Moving Average term.

You can determine these parameters using ACF (AutoCorrelation Function) and PACF (Partial AutoCorrelation Function) plots or using techniques like grid search.

1. **Fit the ARIMA model**

model = ARIMA(data['your\_column'], order=(p, d, q))

results = model.fit()

1. **Make predictions:**

predictions = results.predict(start=start\_date, end=end\_date, dynamic=False)

Replace start\_date and end\_date with the dates for which you want to make predictions.

1. **Visualize predictions:**

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

plt.plot(data['your\_column'], label='Original')

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

plt.title('ARIMA Model Predictions')

plt.xlabel('Date')

plt.ylabel('Value')

plt.legend()

plt.show()

That's it! You've implemented an ARIMA model in Python. Make sure to adjust the code according to your specific dataset and requirements.