**Exp no: 6 Program to apply moving average smoothing for data preparation and time series forecasting.**

**Date: 1/4/25**

**Aim:**

The aim of this project is to analyze and forecast air passenger traffic using time series analysis techniques. Specifically, it focuses on smoothing historical data using a moving average method and predicting future passenger numbers based on observed trends. This helps in understanding seasonality and making informed decisions in aviation-related planning.

**Objectives:**

The objective of this study is to preprocess and analyze the AirPassengers dataset by converting the date column into a proper time index, followed by applying a 12-month moving average to smooth the data and reveal underlying trends. It further aims to forecast the number of passengers for the next twelve months based on historical trends and to visualize both the original and forecasted values to interpret seasonal variations and growth patterns effectively.

**Background/Scope:**

This project utilizes the AirPassengers dataset, which contains monthly records of international airline passengers over a period of several years. The scope of the study includes cleaning and preparing the data for time series analysis, applying smoothing techniques to highlight trends, and generating a simple forecast for future passenger numbers. The project offers foundational insight into time series forecasting and can serve as a base for more advanced predictive modeling, with potential applications in airline capacity planning and strategic resource management.

**Steps for Time Series Sales Data Preprocessing:**

**Step 1: Load the Dataset**

The first step is to load the Air Passenger dataset into a pandas DataFrame. The dataset contains monthly international airline passenger data indexed by date.

import pandas as pd

import matplotlib.pyplot as plt

url = 'https://raw.githubusercontent.com/jbrownlee/Datasets/master/airline-passengers.csv'

data = pd.read\_csv(url, parse\_dates=['Month'], index\_col='Month')

data.columns = ['Passengers']

**Step 2: Visualize the Time Series Data**

Plot the time series data to visualize the trend of airline passengers over time.

import matplotlib.pyplot as plt

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

plt.plot(df.index, df['#Passengers'], label='Passengers')

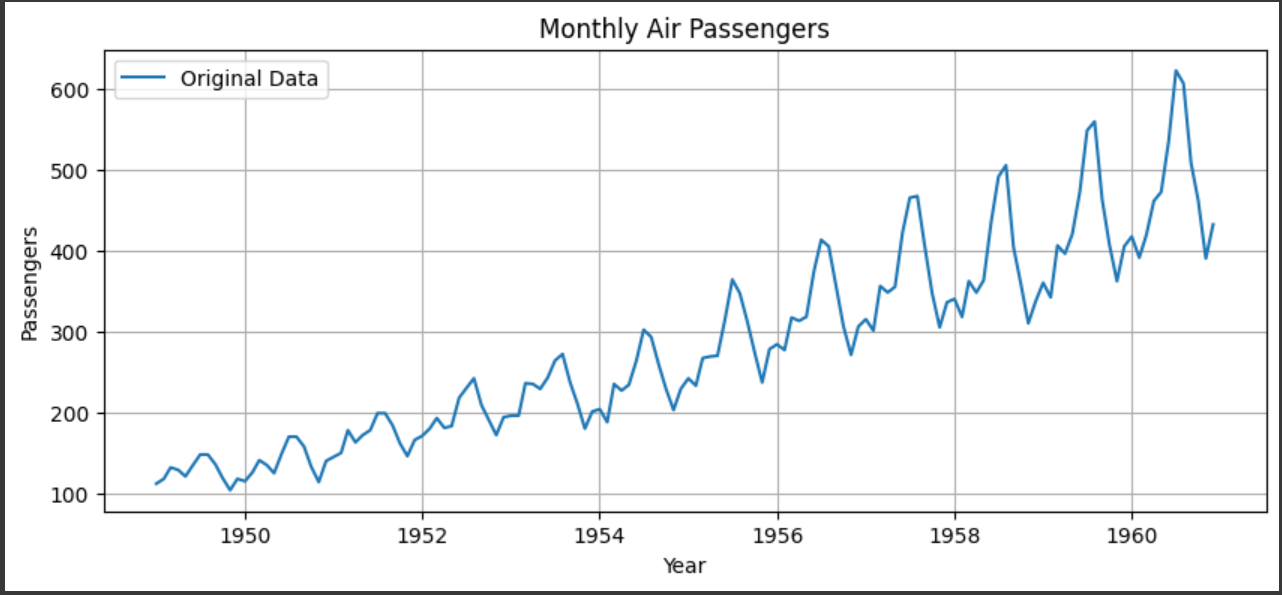
plt.title('Monthly International Airline Passengers (1949-1960)')

plt.xlabel('Date')

plt.ylabel('Number of Passengers')

plt.legend()

plt.show()



**Step 3: Smooth the Data Using Moving Average**

Apply a 12-month moving average to reduce short-term fluctuations and reveal trends. This helps to better visualize the underlying seasonality or growth pattern in the data.

window\_size = 12

df['Smoothed'] = df['# Passengers'].rolling(window=window\_size).mean()

**Step 4: Forecast Future Values Using the Mean of the Last Window**

Calculate the average of the last 12 values and use it to forecast the next 12 months. This simple method assumes future values will follow recent trends.

last\_window = df['# Passengers'][-window\_size:]

forecast\_values = [last\_window.mean()] \* 12

**Step 5: Create Date Index for Forecasted Values**

Generate a list of 12 future monthly dates starting from the month after the last date in the dataset. These dates will serve as indices for the forecasted values.

last\_date = df.index[-1]

future\_dates = pd.date\_range(start=last\_date + pd.DateOffset(months=1), periods=12, freq='MS')

**Step 6: Create Forecast DataFrame and Combine with Original**

Create a new DataFrame for the forecasted values and concatenate it with the smoothed series. This prepares the full dataset for visualization.

forecast\_df = pd.DataFrame({'Smoothed Forecast': forecast\_values}, index=future\_dates)

combined\_df = pd.concat([df[['Smoothed']], forecast\_df])

**Step 7: Visualize Original, Smoothed, and Forecasted Data and Print Forecasted Values**

Plot the original time series, the smoothed trend, and the forecasted values. Add appropriate labels, legend, and styling for clarity. Display the forecasted passenger numbers for the next 12 months in a readable format. This allows for quick verification of predicted results.

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

plt.plot(df['# Passengers'], label='Original Time Series', color='skyblue', alpha=0.5)

plt.plot(df['Smoothed'], label=f'{window\_size}-Month Moving Average', color='darkred', linewidth=2)

plt.plot(forecast\_df, label='Forecast (Next 12 Months)', linestyle='--', color='green')

plt.title('AirPassengers: Original, Smoothed & Forecasted Time Series')

plt.xlabel('Date')

plt.ylabel('Number of Passengers')

plt.legend()

plt.grid(True)

plt.tight\_layout()

plt.show()

print("Forecasted Values (Next 12 Months):")

print(forecast\_df)

A graph with a red line

AI-generated content may be incorrect.

**OUTPUT:**

Forecasted Values (Next 12 Months):

Smoothed Forecast

1961-01-01 476.166667

1961-02-01 476.166667

1961-03-01 476.166667

1961-04-01 476.166667

1961-05-01 476.166667

1961-06-01 476.166667

1961-07-01 476.166667

1961-08-01 476.166667

1961-09-01 476.166667

1961-10-01 476.166667

1961-11-01 476.166667

1961-12-01 476.166667

**Result:**

Thus, the AirPassengers time series was successfully smoothed and forecasted using moving average techniques, revealing trends and predicting future values effectively.