Terna Engineering College

Department of Artificial Intelligence and Data Science

Program: Sem VI

Course: Data Analytics and Visualization Lab

Experiment No.07

PART A

(PART A: TO BE REFERRED BY STUDENTS)

A.1 Aim: To implement the ARIMA model in Python/R.

A.2 Theory:

Time Series Analysis:

A time series is an ordered sequence of equally spaced values over time or a sequence of data points that occur in successive order over some period of time. For e.g. In investing, it is common to use a time series to track the price of a security over time. This can be tracked over the short term, such as the price of a security on the hour over the course of a business day, or the long term, such as the price of a security at close on the last day of every month over the course of five years.

Types of time series:

Stationary: A dataset where the mean, variance and covariance values of them should be completely constant in the data during the analysis is a stationary time series.

Non-stationary: If either the mean-variance or covariance is changing with respect to time, the dataset is called non-stationary.

ARIMA Model:

The ARIMA model is a statistical model utilized for analyzing and predicting time series data. The ARIMA approach explicitly caters to standard structures found in time series, providing a simple yet powerful method for making skillful time series forecasts.

ARIMA stands for AutoRegressive Integrated Moving Average. It combines three key aspects:

- **Autoregression (AR)**: A model that uses the correlation between the current observation and lagged observations. The number of lagged observations is referred to as the lag order or p.
- **Integrated (I)**: The use of differencing of raw observations to make the time series stationary. The number of differencing operations is referred to as d.
- **Moving Average (MA)**: A model takes into account the relationship between the current observation and the residual errors from a moving average model applied to past observations. The size of the moving average window is the order or q.

The ARIMA model is defined with the notation ARIMA(p,d,q) where p, d, and q are substituted with integer values to specify the exact model being used.

- p means the number of preceding ("lagged") Y values that have to be added/subtracted to Y in the model, so as to make better predictions based on local periods of growth/decline in our data. This captures the "autoregressive" nature of ARIMA.
- d represents the number of times that the data have to be "differenced" to produce a stationary signal (i.e., a signal that has a constant mean over time). This captures the "integrated" nature of ARIMA. If d=0, this means that our data does not tend to go up/down in the long term (i.e., the model is already "stationary"). In this case, then technically you are performing just ARMA, not AR-I-MA. If p is 1, then it means that the data is going up/down linearly. If p is 2, then it means that the data is going up/down exponentially.
- q represents the number of preceding/lagged values for the error term that are added/subtracted to Y. This captures the "moving average" part of ARIMA.

ARIMA model is generally used for handling non-stationary data, as the differencing parameter helps to make the time series stationary for analyzing. Equation of ARIMA model is given by :

$$y'_t = c + \phi_1 y'_{t-1} + \cdots + \phi_p y'_{t-p} + \theta_1 \varepsilon_{t-1} + \cdots + \theta_q \varepsilon_{t-q} + \varepsilon_t,$$

SARIMA & SARIMAX Models

SARIMA model

$$y_t = c + \sum_{n=1}^{p} \alpha_n y_{t-n} + \sum_{n=1}^{q} \theta_n \epsilon_{t-n} + \sum_{n=1}^{P} \phi_n y_{t-sn} + \sum_{n=1}^{Q} \eta_n \epsilon_{t-sn} + \epsilon_t$$

This model is very similar to the ARIMA model, except that there is an additional set of autoregressive and moving average components. The additional lags are offset by the frequency of seasonality (ex. 12 — monthly, 24 — hourly). SARIMA models allow for differencing data by seasonal frequency, yet also by non-seasonal differencing.

SARIMAX model

$$d_t = c + \sum_{n=1}^{p} \alpha_n d_{t-n} + \sum_{n=1}^{q} \theta_n \epsilon_{t-n} + \sum_{n=1}^{r} \beta_n x_{n_t} + \sum_{n=1}^{P} \phi_n d_{t-sn} + \sum_{n=1}^{Q} \eta_n \epsilon_{t-sn} + \epsilon_t$$

Above is the equation of the SARIMAX model. This model takes into account exogenous variables, or in other words, external data in our forecast. Some real-world examples of exogenous variables include gold price, oil price, outdoor temperature, and exchange rate.

PART B

(PART B: TO BE COMPLETED BY STUDENTS)

(Students must submit the soft copy as per the following segments within two hours of the practical. The soft copy must be uploaded on the Blackboard or emailed to the concerned lab in charge faculties at the end of the practical in case there is no Blackboard access available).

Roll. No. A12	Name: Sufiyan Khan
Class: TE – AI & DS	Batch: A1
Date of Experiment:	Date of Submission: 24-03-24
Grade:	

B.1 Input and Output:

1. Import libraries and dataset

```
import numpy as np
import pandas as pd
import matplotlib.pylab as plt
from statsmodels.tsa.seasonal import seasonal_decompose
import warnings
warnings.filterwarnings('ignore')
```

✓ 0.0s

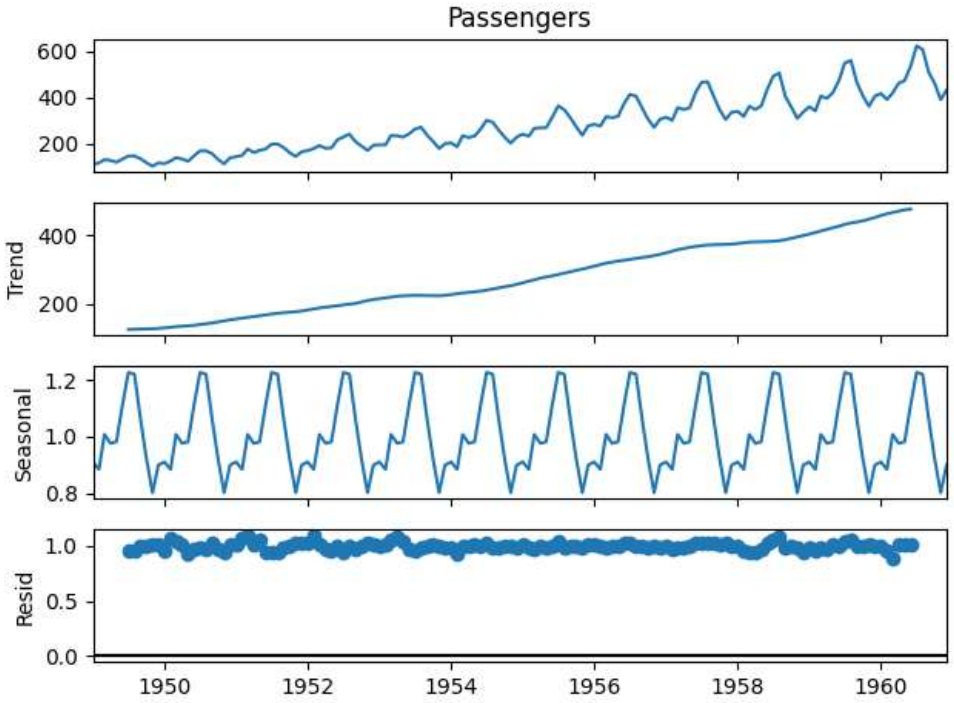
```
airline = pd.read_csv('AirPassengers.csv', index_col ='Month', parse_dates = True)
airline.rename(columns={'#Passengers':'Passengers'},inplace=True)
airline.head()
```

✓ 0.0s

Month	Passengers
1949-01-01	112
1949-02-01	118
1949-03-01	132
1949-04-01	129
1949-05-01	121

2. Perform ETS decomposition (Error, Trend, Seasonality) to separate the different components of the time series dataset

```
result = seasonal_decompose(airline['Passengers'],  model ='multiplicative')
result.plot()
```
✓ 1.2s



3. Import pmdarima library and perform stepwise fitting of model on the data. Use auto_arima method o identify the most optimal parameters for an ARIMA model and returns a fitted ARIMA model.

```
from pmdarima import auto_arima

stepwise_fit = auto_arima(airline['Passengers'], start_p = 1, start_q = 1,
                          max_p = 3, max_q = 3, m = 12,
                          start_P = 0, seasonal = True,
                          d = None, D = 1, trace = True,
                          error_action ='ignore',
                          suppress_warnings = True,
                          stepwise = True)

stepwise_fit.summary()
```

SARIMAX Results

Dep. Variable:			y	No. Observations:		144
Model:	SARIMAX(0, 1, 1)x(2, 1, [], 12)			Log Likelihood		-505.589
Date:			Sat, 16 Mar 2024	AIC		1019.178
Time:			13:24:37	BIC		1030.679
Sample:			01-01-1949	HQIC		1023.851
			- 12-01-1960			
Covariance Type:			opg			
	coef	std err	z	P>|z|	[0.025	0.975]
ma.L1	-0.3634	0.074	-4.945	0.000	-0.508	-0.219
ar.S.L12	-0.1239	0.090	-1.372	0.170	-0.301	0.053
ar.S.L24	0.1911	0.107	1.783	0.075	-0.019	0.401
sigma2	130.4480	15.527	8.402	0.000	100.016	160.880
Ljung-Box (L1) (Q):		0.01	Jarque-Bera (JB):		4.59	
Prob(Q):		0.92	Prob(JB):		0.10	
Heteroskedasticity (H):		2.70	Skew:		0.15	
Prob(H) (two-sided):		0.00	Kurtosis:		3.87	

4. Fit ARIMA Model to AirPassengers dataset. We use the SARIMAX model which is Seasonal ARIMA with exogenous variables. This model takes into account exogenous variables, or in other words, it uses external data in our forecast.

```
train = airline.iloc[:len(airline)-12]
test = airline.iloc[len(airline)-12:]

from statsmodels.tsa.statespace.sarimax import SARIMAX

model = SARIMAX(train['Passengers'],
                order = (0, 1, 1),
                seasonal_order =(2, 1, 1, 12))

result = model.fit()
result.summary()
```

✓ 1.3s

SARIMAX Results

Dep. Variable:			Passengers	No. Observations:		132
Model:	SARIMAX(0, 1, 1)x(2, 1, 1, 12)			Log Likelihood		-443.013
Date:			Sat, 16 Mar 2024	AIC		896.026
Time:			13:24:38	BIC		909.921
Sample:			01-01-1949	HQIC		901.668
			- 12-01-1959			
Covariance Type:			opg			
	coef	std err	z	P>|z|	[0.025	0.975]
ma.L1	-0.2985	0.074	-4.032	0.000	-0.444	-0.153
ar.S.L12	0.7099	0.222	3.194	0.001	0.274	1.146
ar.S.L24	0.2892	0.100	2.891	0.004	0.093	0.485
ma.S.L12	-0.9806	2.139	-0.459	0.647	-5.172	3.211
sigma2	88.1757	170.744	0.516	0.606	-246.476	422.828
Ljung-Box (L1) (Q):		0.03	Jarque-Bera (JB):		0.00	
Prob(Q):		0.85	Prob(JB):		1.00	
Heteroskedasticity (H):		1.62	Skew:		-0.00	
Prob(H) (two-sided):		0.13	Kurtosis:		2.99	

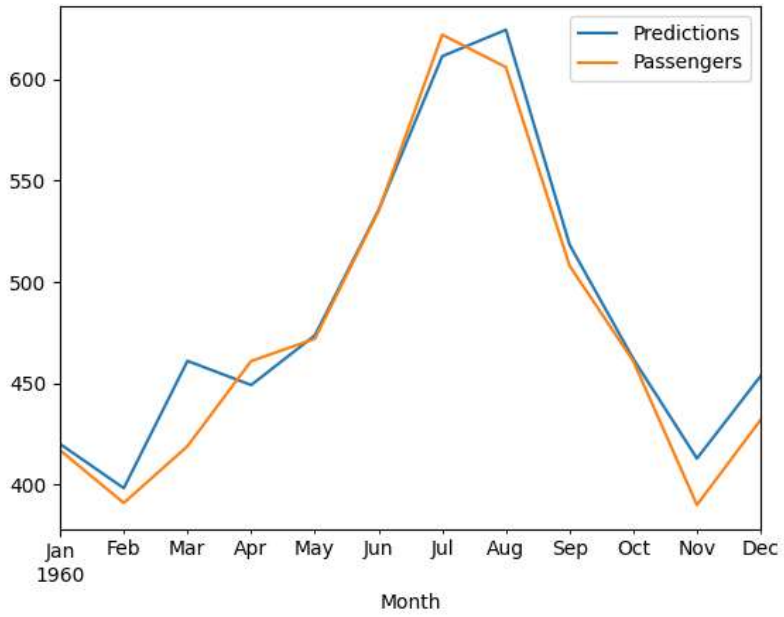
5. Predict or forecast the model's results using testing set.

```
start = len(train)
end = len(train) + len(test) - 1

predictions = result.predict(start, end,
                             typ = 'levels').rename("Predictions")

predictions.plot(legend = True)
test['Passengers'].plot(legend = True)
```

✓ 0.1s




```
from statsmodels.tools.eval_measures import rmse
print("RMSE: ",rmse(test["Passengers"], predictions))
```

✓ 0.0s

```
RMSE:  17.150775038270297
```

6. Forecast no. of passengers using seasonal cycles and trends for the given data.

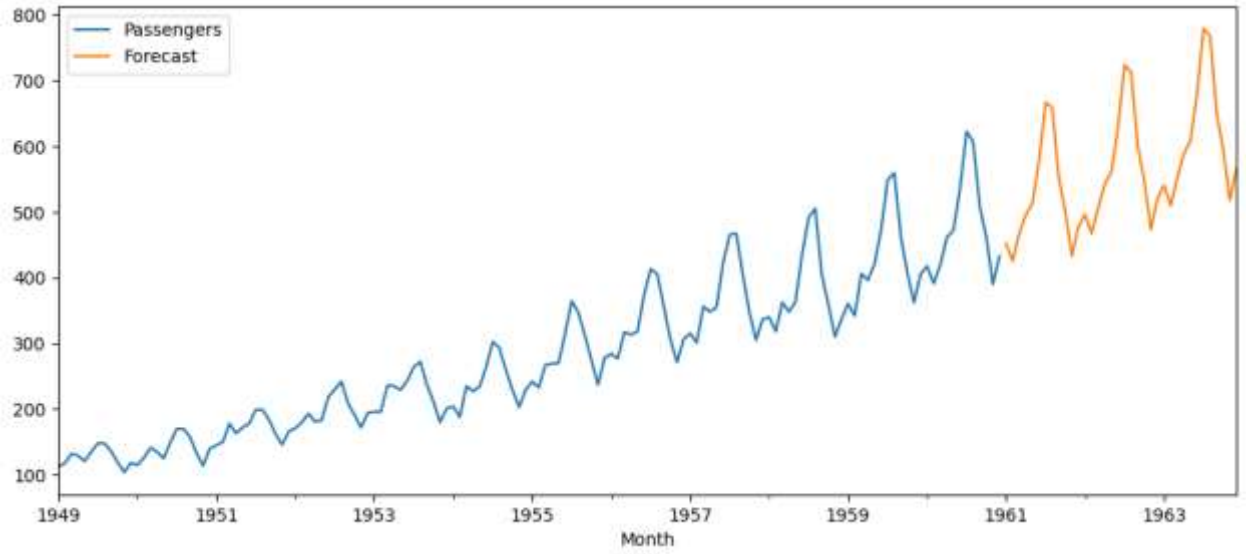
```
model = model = SARIMAX(airline['Passengers'],
                        order = (0, 1, 1),
                        seasonal_order =(2, 1, 1, 12))
result = model.fit()
```

✓ 1.4s

```
forecast = result.predict(start = len(airline),
                          end = (len(airline)-1) + 3 * 12,
                          typ = 'levels').rename('Forecast')

airline['Passengers'].plot(figsize = (12, 5), legend = True)
forecast.plot(legend = True)
```

✓ 0.1s



B.2 Conclusion:

Thus we have successfully implemented ARIMA model for time series analysis in Python on air passengers dataset & understood how the data is distributed and observed the trends in the series. We have also trained the model to forecast future data based on observed trends.