كلية الحوسبة والمعلوماتية |
College of Computing and Informatics

الجامعة السعودية الإلكترونية
SAUDI ELECTRONIC UNIVERSITY
2011-1432

Semester 1 – 2021/2022

Course Code	DS550
Course Name	Machine Learning
Assignment type	Critical Thinking
Module	12

Student ID	G200007615
Student Name	Abdulaziz Alqumayzi
CRN	15062

Solutions:

Critical Thinking Assignment 5

Introduction

In this activity, we will define and explain ARIMA Time Series algorithm. Provide a python programming example on how to apply the algorithm on a dataset. The dataset used is **Daily Climate time series data** from Kaggle.

Data Set Information: The Dataset is exclusively for developers who wish to train a weather forecasting model for the Indian environment. This dataset contains data from the 1st of January 2013 to the 24th of April 2017 in Delhi, India. meantemp, humidity, wind_speed, and meanpressure are the four factors here.

Time Series Assumption

Sample statistics are data properties such as central tendency, dispersion, skewness, and kurtosis. Two of the most frequent sample statistics are mean and variance. Data is gathered from a sample of the greater population in any analysis. The sample data is then used to estimate mean, variance, and other attributes. As a result, sample statistics are used.

For sample statistics to be accurate, a key assumption in statistical estimation theory is that the population does not suffer any fundamental or systemic alterations over the persons in the sample or during the time the data was gathered.

This assumption also holds true for time series analysis, allowing the mean, variance, and autocorrelation calculated from the simple to be utilized as a credible forecast for future events. This assumption is known as stationarity in time series analysis, and it states that the series' underlying structures do not change over time. As a result, stationarity necessitates the invariance of mean, variance, and autocorrelation with regard to the actual time of observation.

A **stationary** time series has statistical features or moments that do not change throughout time (e.g., mean and variance). The status of a stationary time series is thus defined as stationarity. **Nonstationary**, on the other hand, is the state of a time series whose statistical features change with time.

ARIMA Time Series Algorithm

The Box-Jenkins model, commonly known as ARIMA, is a generalization of the ARMA model that includes integrated components. When data exhibits non-stationarity, the integrated components are important, and the integrated element of ARIMA aids in minimizing non-stationarity. To reduce the non-stationarity impact, the ARIMA applies differencing to the time series one or more times. The order of AR, MA, and differencing components is represented as ARIMA(p, d, q). The d component, which refreshes the series on which the forecasting model is formed, is the main distinction between ARMA and ARIMA models. The ARMA model may be used to the de-trended dataset after the d component detrends the signal to make it stationary. The following are the parameters of the ARIMA model:

p: The lag order is the number of lag observations incorporated in the model.

d: The degree of differencing refers to the number of times the raw observations differ.

q: The order of moving average refers to the size of the moving average window.

Applications of the ARIMA Algorithm

Using previous data to forecast the quantity of a good that will be required in the future. Forecasting sales and analyzing seasonal variations in sales. Estimating the impact of marketing events, new product launches, and other similar activities.

Limitations of the ARIMA Algorithm

Although ARIMA models may be extremely precise and dependable under the right conditions and with enough data, one of the model's major drawbacks is that the parameters (p, d, and q) must be manually specified, making obtaining the best fit a lengthy trial-and-error process. Similarly, the model is heavily reliant on the consistency and differencing of previous data. To guarantee that the model offers reliable findings and projections, it is critical to ensure that data was collected properly and over a lengthy period of time.

Python Code

```
# importing needed packages
import pandas as pd
import matplotlib.pyplot as plt
from statsmodels.tsa.arima.model import ARIMA
from statsmodels.tsa.stattools import adfuller
from pandas.plotting import autocorrelation_plot
from statsmodels.graphics.tsaplots import plot_acf,plot_pacf
import statsmodels.api as sm

#importing datasets
df = pd.read_csv('DailyDelhiClimateTrain.csv', index_col=0,
parse_dates=["date"])

# check if null values exists
df.info()

# plot the mean temperature
plt.figure(figsize = (18,8))
plt.plot(df['meantemp']);

# the function below considering the null hypothesis that data is not
stationary
# and the alternate hypothesis that data is stationary
def adfuller_test(temp):
    result=adfuller(temp)
    labels = ['ADF Test Statistic','p-value','#Lags Used','Number of
Observations']
    for value,label in zip(result,labels):
        print(label+' : '+str(value) )

    if result[1] <= 0.05:
        print("strong evidence against the null hypothesis(Ho), reject the
null hypothesis. Data is stationary")
    else:
        print("weak evidence against null hypothesis,indicating it is non-
stationary ")

# apply the function
adfuller_test(df['meantemp'])

# adding seasonality
df['Temperature First Difference'] = df['meantemp'] -
df['meantemp'].shift(1)
df['Seasonal First Difference']=df['meantemp']-df['meantemp'].shift(12)
df.head()

# testing if data is stationary
adfuller_test(df['Seasonal First Difference'].dropna())
```

```
# plot the mean seasonal
plt.figure(figsize = (18,8))
df['Seasonal First Difference'].plot();

# create auto-correlation
plt.figure(figsize = (18,8))
autocorrelation_plot(df['meantemp'])
plt.show()

fig = plt.figure(figsize=(18,8))
ax1 = fig.add_subplot(211)
fig = sm.graphics.tsa.plot_acf(df['Seasonal First
Difference'].dropna(),lags=60,ax=ax1)
ax2 = fig.add_subplot(212)
fig = sm.graphics.tsa.plot_pacf(df['Seasonal First
Difference'].dropna(),lags=60,ax=ax2)

# building ARIMA model
model=ARIMA(df['meantemp'],order=(1,1,1))
model_fit=model.fit()
model_fit.summary()
```

References

Nielsen, A. (2020). *Practical time series analysis: Prediction with statistics and machine learning*. O'Reilly.

Sumanthvrao. (2019, August 23). *Daily climate time series data*. Kaggle. Retrieved December 3, 2021, from https://www.kaggle.com/sumanthvrao/daily-climate-time-series-data

Pathak , P. (2020, October 29). *How to create an Arima model for time series forecasting in python?* Analytics Vidhya. Retrieved December 3, 2021, from https://www.analyticsvidhya.com/blog/2020/10/how-to-create-an-arima-model-for-time-series-forecasting-in-python/

Brownlee, J. (2020, December 9). *How to create an Arima model for time series forecasting in Python*. Machine Learning Mastery. Retrieved December 3, 2021, from https://machinelearningmastery.com/arima-for-time-series-forecasting-with-python/

Autoregressive Integrated moving average (ARIMA). Corporate Finance Institute. (2021, July 8). Retrieved December 3, 2021, from https://corporatefinanceinstitute.com/resources/knowledge/other/autoregressive-integrated-moving-average-arima/

Ryan E. Emanuel, Joshua S. Rice, and Jasmine N. Gregory. (2021). *Stationary and nonstationary behavior*. North Carolina State University. (2021, February 22). Retrieved December 3, 2021, from https://serc.carleton.edu/hydromodules/steps/236435.html#:~:text=A%20stationary%20time%20series%20has,properties%20are%20changing%20through%20time