**Introduction to ARIMA for Time Series Forecasting**

Getting to know one of the more popular ML algorithms for time-series forecasting.

Auto Regressive Integrated Moving Average (ARIMA) model is among one of the more popular and widely used statistical methods for time-series forecasting. It is a class of statistical algorithms that captures the standard temporal dependencies that is unique to a time series data. In this post, I will introduce you to the basic principles of ARIMA and present a hands-on tutorial to develop ARIMA for time-series forecasting in Python.

# ****What is ARIMA?****

## ****Keywords: Stationarity and Autocorrelation****

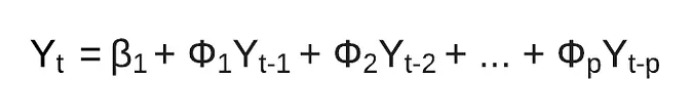
Before we dig right into ARIMA’s formal mathematical definition, let me introduce you to the concept of **stationarity**. Stationarity simply means observations that do not depend on time. For data that depends on time (eg. seasonal rainfall), the stationarity condition may not hold as different timing will yield different values for these observations.

Another important concept to understanding ARIMA is **autocorrelation**. How does it different from the typical correlation? First of all, correlation relates two different sets of observations (eg. between housing prices and the number of available public amenities) while autocorrelation relates the same set of observation but across different timing (eg. between rainfall in the summer versus that in the fall).

Now let’s break down the different components of ARIMA: Auto Regressive (AR), Integrated (I), and the Moving Average (MA)

## The AR in [AR]IMA: Auto Regressive

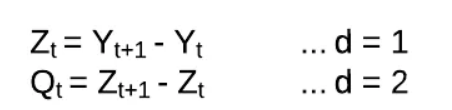
As you may have guessed it, Auto Regressive (AR) regression model is built on top of the autocorrelation concept, where the dependent variable depends on the past values of itself (eg. rainfall today may depend on rainfall yesterday, and so on). The general equation is:



As illustrated, an observation Y at time t, Yt, depends on Yt-1, Yt-2, ..., Yt-p . Why Yt-p and not Y0(ie. the initial value)? The **p**here is called the lag order which indicates the number of prior lag observations we include in the model (eg. Maybe we exclude observations beyond 5 days prior to the present time because these earlier rainfall observations might not be correlated) .

## ****The I in AR[I]MA: Integrated****

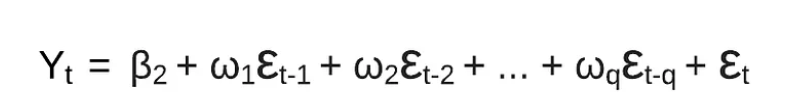
Recall our explanation on stationarity. The integrated part of ARIMA attempts to convert the non-stationarity nature of the time-series data to something a little bit more stationary. How can we do that? By performing prediction on the **difference** between any two pair of observation rather than directly on the data itself.



Notice how we could perform the differencing operations many times levels (**Y →Z**and **Z →Q**), depending on the hyperparameter **d** that we set when training the ARIMA model.

## The MA in ARI[MA]: Moving Average

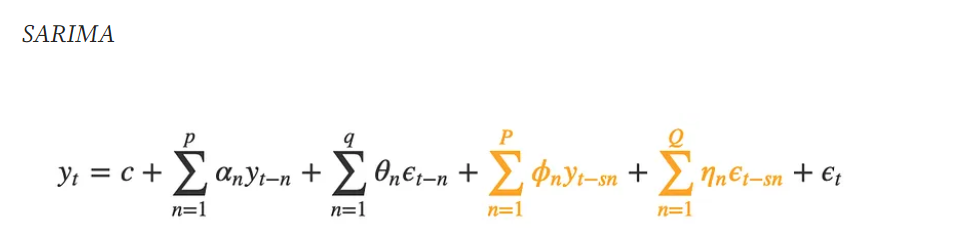
Now the final piece of ARIMA is the MA or Moving Average. It attempts to reduce the noise in ourtime series data by performing some sort of aggregation operation to your past observations in terms of residual error ε (read: epsilon).



The ε terms represent the residual errors from the aggregation function and **q**here is another hyperparameter that is identical to**p.**But instead of identifying the time window (**p**) to the time series data itself, **q** specifies the time window for the moving average’s residual error.

## SARIMA, ARIMAX, SARIMAX Models

The ARIMA model is great, but to include seasonality and exogenous variables in the model can be extremely powerful. Since the ARIMA model assumes that the time series is stationary, we need to use a different model.

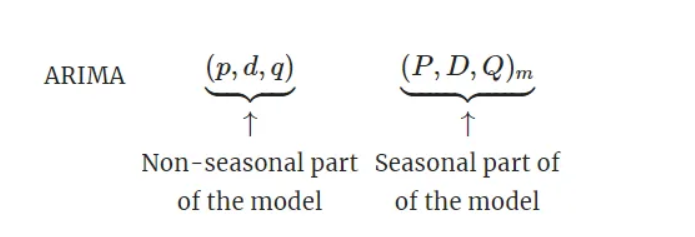


Enter SARIMA (Seasonal ARIMA). This model is very similar to the ARIMA model, except that there is an additional set of autoregressive and moving average components. The additional lags are offset by the frequency of seasonality (ex. 12 — monthly, 24 — hourly).

SARIMA models allow for differencing data by seasonal frequency, yet also by non-seasonal differencing. Knowing which parameters are best can be made easier through automatic parameter search frameworks such as [pmdarina](http://alkaline-ml.com/pmdarima/" \t "_blank).

**SARIMA (Seasonal Auto-Regressive Integrated Moving Average)**is an extension of the [**ARIMA (Autoregressive Integrated Moving Average)**](https://medium.com/@ritusantra/introduction-to-arima-model-c8925103f4c7) model that incorporates **seasonality** in addition to the non-seasonal components. ARIMA models are widely used for time series analysis and forecasting, while SARIMA models are specifically designed to handle data with seasonal patterns.

SARIMA Model is represented as,



where,

**m**= number of observations per year; **P**= Number of seasonal AR terms;

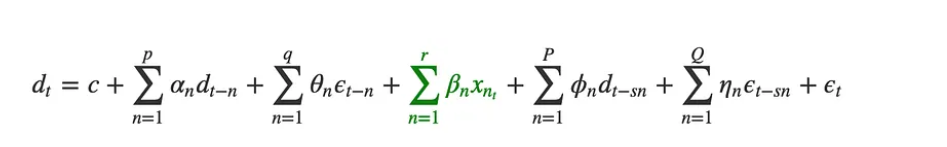
**D**= Number of seasonal differences; **Q**= Number of seasonal MA terms

We use uppercase notation for the seasonal parts of the model, and lowercase notation for the non-seasonal parts of the model.

The seasonal component of SARIMA models adds the following three components:

1. **Seasonal Autoregressive (P)**: This component captures the relationship between the current value of the series and its past values, specifically at seasonal lags.
2. **Seasonal Integrated (D)**: Similar to the non-seasonal differencing, this component accounts for the differencing required to remove seasonality from the series.
3. **Seasonal Moving Average (Q)**: This component models the dependency between the current value and the residual errors of the previous predictions at seasonal lags.

## ARIMAX and SARIMAX



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

It is interesting to think that all exogenous factors are still technically indirectly modeled in the historical model forecast. That being said, if we include external data, the model will respond much quicker to its affect than if we rely on the influence of lagging terms.