FORECASTING

Whenever data is recorded at regular intervals of time, it is called a time series.

- Time series must be stationary i.e. it’s statistical properties such as mean, variance, autocorrelation must all be constant over time.

- We assume that there is no outlier in the series. Outliers may affect conclusions strongly and can be misleading.

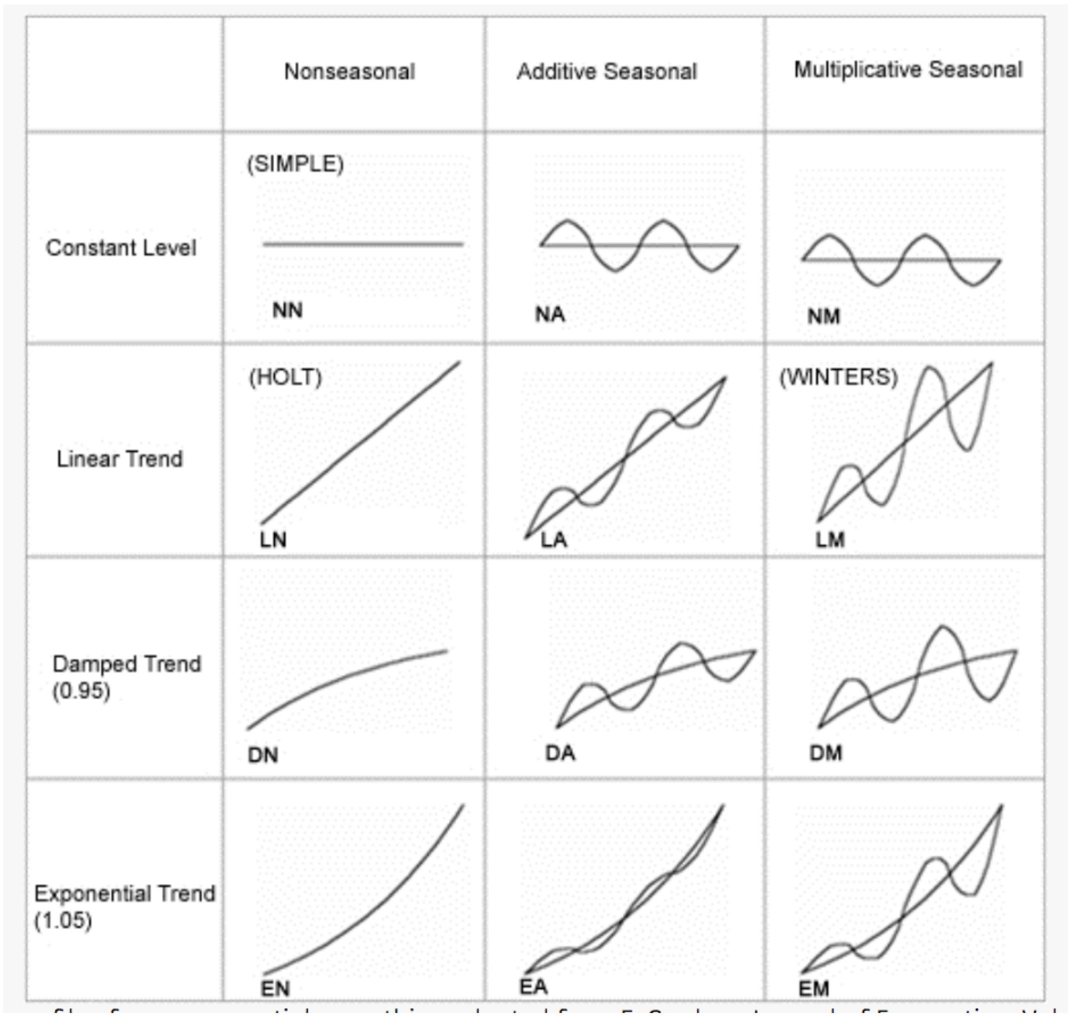
Types of Forecasting

If you use only the previous values of the time series to predict its future values, it is called Univariate Time Series Forecasting.

And if you use predictors other than the series (also known as exogenous variables) or if more than one input variable is used to forecast it is called Multi Variate Time Series Forecasting.

Important Terms

Y=level + trend + seasonality + noise



Level

The baseline value for the series if it were a straight line or the average value in the series.

Trend

It is the increasing or decreasing behavior of the series over time.

Seasonality

A pattern that repeats at particular intervals. This refers to the property of a time series that displays periodical patterns that repeats at a constant frequency (m). A seasonal pattern occurs when a time series is affected by seasonal factors such as the time of the year or the day of the week. Seasonality is always of a fixed and known frequency

Noise

It is the random fluctuation or unpredictable change. This is something which we cannot guess. A random component that does not have or follow a specific pattern. It is variability in the observations that cannot be explained by the model.

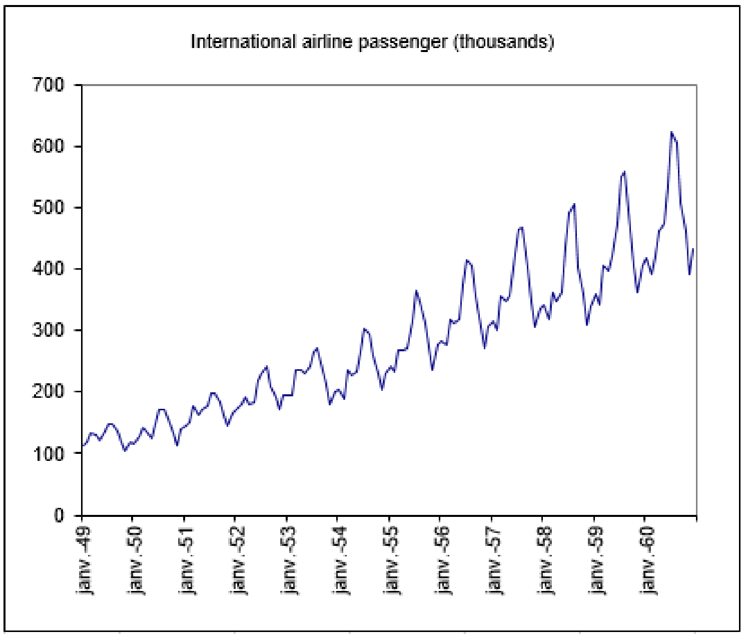
Stationarize time series

A Non-stationary time series show trends, seasonal effects. Mean and variance change over time and a drift in the model is captured.

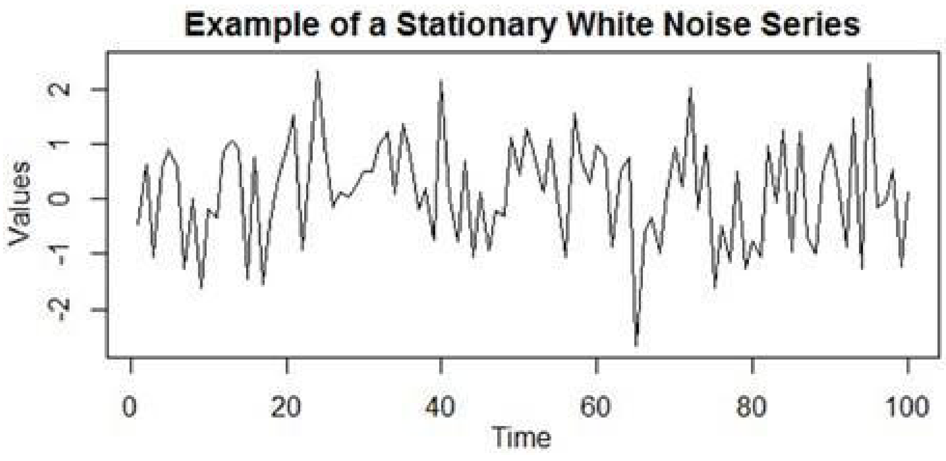
There are 2 major reasons behind non-stationarity of a TS:

Trend - varying mean over time. Ex: Number of passengers was growing over time.

Seasonality - variations at specific time-frames. Ex: People might have a tendency to buy cars in a particular month because of pay increment or festivals.



Stationary means that the statistical properties of a time series do not change over time. It does not mean that the series does not change over time, it’s just that although time series data is changing, the statistical properties of the time series remain same.



How to check for Stationarity?

- ADF (Augmented Dickey-Fuller) Test

If the test statistic is less than the critical value, we can reject the null hypothesis and say that the series is stationary.

Null Hypothesis (H0): If accepted, it suggests the time series has a unit root, meaning it is non-stationary.

Alternate Hypothesis (H1): The null hypothesis is rejected; it suggests the time series does not have a unit root, meaning it is stationary.

p-value > 0.05: Accept Null Hypothesis(H0), the data has a unit root and is non-stationary

p-value ≤ 0.05: Reject Null Hypothesis(H0), the data does not have a unit root and is stationary

KPSS (Kwiatkowski-Phillips-Schmidt-Shin) Test

The null and alternate hypothesis for the KPSS test are opposite that of the ADF test.

Null Hypothesis(H0): The series is stationary.

Alternate Hypothesis(H1): The series is not stationary

p-value > 0.05: Accept Null Hypothesis(H0), the data does not have a unit root and is stationary

p-value ≤ 0.05: Reject Null Hypothesis(H0), the data does have a unit root and is not stationary

It's always better to apply both the tests, so that we are sure that the series is truly stationary. Let us look at the possible outcomes of applying these stationary tests.

Case 1: Both tests conclude that the series is not stationary then it is series is not stationary

Case 2: Both tests conclude that the series is stationary then it is series is stationary

Case 3: KPSS = stationary and ADF = not stationary then it is trend stationary. Trend needs to be removed to make series strict stationary. The detrended series is checked for stationarity.

Case 4: KPSS = not stationary and ADF = stationary then it is difference stationary. Differencing is to be used to make series stationary. The differenced series is checked for stationarity.

How to make time series stationary?

Trend and Seasonality are major reasons behind non-stationarity of a time series. The underlying principle is to model or estimate the trend and seasonality in the series and remove those from the series to get a stationary series. Then statistical forecasting techniques can be implemented on this series. The final step would be to convert the forecasted values into the original scale by applying trend and seasonality constraints back.

How to remove trend?

- Transformation
- Differencing
- Differencing over Transformation

How to remove seasonality?

- Transformation
- Differencing
- Differencing over transformation
- Seasonality Differencing

Having performed the trend and seasonality estimation techniques, there can be two situations.

- A strictly stationary series with no dependence among the values. This is the easy case where in we can model the residuals as white noise. But this is very rare.
- A series with significant dependence among values. In this case we need to use some statistical models like ARIMA and more to forecast the data.

Python Forecasting Models

Autoregression (AR)

The autoregression (AR) method models the next step in the sequence as a linear function of the observations at prior time steps. This technique can be used on time series where input variables are taken as observations at previous time steps (also called lag variables). With the autoregression model, you are using previous data points and using them to predict future data point(s) but with multiple lag variables.

A pure Auto Regressive model is one where Yt depends only on its own lags. That is, Yt is a function of the 'lags of Yt'.

$$Y_t = \alpha + \beta_1 Y_{t-1} + \beta_2 Y_{t-2} + .. + \beta_p Y_{t-p} + \epsilon_1$$

An autoregression model makes an assumption that the observations at previous time steps are useful to predict the value at the next time step. This relationship between variables is called correlation and we can use statistical measures to calculate the correlation between the output variable and values at previous time steps at various different lags. The stronger the correlation between the output variable and a specific lagged variable, the more weight that autoregression model can put on that variable when modeling.

The method is suitable for univariate time series without trend and seasonal components i.e. for a stationary time series.

Moving Average (MA)

Many a times we are provided with a dataset in which the prices/sales of the object increased/decreased sharply some time periods ago. In order to use the previous average method, we have to use the mean of all the previous data but using all the previous data doesn't sound right. So instead of taking mean of all the previous periods we take mean of last few time periods only. Obviously the thinking here is that only the recent values matter. Such forecasting technique which uses window of time period for calculating the average is called Moving Average technique.

A pure Moving Average model is one where Yt depends on the lagged values.

Yt=(Y(t-1)+Y(t-2)+...Y(t-N))N

The method is suitable for univariate time series without trend and seasonal components i.e. for a stationary time series.

Autoregressive Moving Average (ARMA)

ARMA is a model of forecasting in which the methods of autoregression (AR) and moving average (MA) are both applied to time-series data to get forecasted values.

Autoregressive Moving Average model describes time series in terms of two polynomials. The first of these polynomials is for autoregression, the second for the moving average. Often this model is referred to as the ARMA (p, q) model where p is the order of the autoregressive polynomial and q is the order of the moving average polynomial.

The method is suitable for univariate time series without trend and seasonal components.

Autoregressive Integrated Moving Average (ARIMA)

It combines both Autoregression (AR) and Moving Average (MA) models with a Differencing pre-processing step of the sequence to make the sequence stationary.

Any non-seasonal time series that exhibits patterns and is not a random white noise can be modeled with ARIMA models.

An ARIMA model is characterized by 3 terms as p, d, q where

p - order of the Auto Regressive (AR) term and refers to the number of lags of Y to be used as predictors

d - the minimum number of differencing needed to make the series stationary

q - order of the Moving Average (MA) term and refers to the number of lagged forecast errors to be used

An ARIMA model is one where the time series was differenced to make it stationary and you combine the AR and the MA terms.

Predicted Yt = Constant + Linear combination Lags p lags + Moving average of q lags

How to find the number of AR and MA terms?

The ACF and PACF plots should be considered together to get the values of AR and MA terms.

ACF is an (complete) auto-correlation function which gives us values of auto-correlation of any series with its lagged values. In simple terms, it describes how well the present value of the series is related with its past values. A time series can have components like trend, seasonality. ACF considers all these components while finding correlations hence it’s a ‘complete auto-correlation plot’.

PACF is a partial auto-correlation function. Basically instead of finding correlations of present with lags like ACF, it finds correlation of the residuals (which remains after removing the effects which are already explained by the earlier lag(s)) with the lag values and hence ‘partial’ and not ‘complete’

How to find the order of differencing (d) in ARIMA model?

Adf Test

Auto Arima is an automated arima version which performs various combinations of parameters to find the best arima model at the run time.

Arima method is suitable for univariate time series with trend and without seasonal components

Seasonal Autoregressive Integrated Moving-Average (SARIMA)

An extension to ARIMA that supports the modeling of the seasonal component of the series is called SARIMA. If a time series has seasonal patterns, then you need to add seasonal terms for which we use SARIMA

The problem with plain ARIMA model is it does not support seasonality. ARIMA expects data that is either not seasonal or has the seasonal component removed, e.g. seasonally adjusted via methods such as seasonal differencing. If your time series has defined seasonality, then, go for SARIMA which uses seasonal differencing. Seasonal differencing is similar to regular differencing but instead of subtracting consecutive terms you subtract the value from previous season. It adds three new hyperparameters to specify the autoregression (AR), Differencing (I) and Moving Average (MA) for the seasonal component of the series as well as an additional parameter for the period of the seasonality.

So, the model will be represented as SARIMA (p, d, q) x (P, D, Q) where

P - order of Seasonal Autoregression

D - order of Seasonal Difference

Q - order of Seasonal Moving Average

X - Frequency of the time series

The method is suitable for univariate time series with trend and/or seasonal components.

Seasonal Autoregressive Integrated Moving-Average with Exogenous Regressors (SARIMAX)

The Seasonal Autoregressive Integrated Moving-Average with Exogenous Regressors (SARIMAX) is an extension of the SARIMA model that also includes the modeling of exogenous variables.

Exogenous variables are also called covariates and can be thought of as parallel input sequences that have observations at the same time steps as the original series. The primary series may be referred to as endogenous data to contrast it from the exogenous sequence(s). The observations for exogenous variables are included in the model directly at each time step and are not modeled in the same way as the primary endogenous sequence i.e. as an AR, MA process.

The method is suitable for univariate time series with trend and/or seasonal components and exogenous variables.

Simple Exponential Smoothing (SES)

After we have understood the above methods, we can note that both Simple average and Weighted moving average lie on completely opposite ends. We would need something between these two extremes approaches which takes into account all the data while weighing the data points differently. For example, it may be sensible to attach larger weights to more recent observations than to observations from the distant past. The technique which works on this principle is called Exponential smoothing.

Simple Exponential Smoothing (SES) methods models the next time step as an exponentially weighted linear function of observations at prior time steps. It requires a single parameter called alpha (a) also called the smoothing factor or smoothing coefficient. This parameter controls the rate at which the influence of the observations at prior time steps decay exponentially. Alpha is often set to a value between 0 and 1. Large values mean that the model pays attention mainly to the most recent past observations whereas smaller values mean more of the history is taken into account when making a prediction. A value close to 1 indicates fast learning i.e. only the most recent values influence the forecasts whereas a value close to 0 indicates slow learning i.e. past observations have a large influence on forecasts

Forecasts are calculated using weighted averages where the weights decrease exponentially as observations come from further in the past with the smallest weights being associated with the oldest observations. Exponential smoothing forecasting methods are similar in that a prediction is a weighted sum of past observations but the model explicitly uses an exponentially decreasing weight for past observations. Specifically, past observations are weighted with a geometrically decreasing ratio.

$$\hat{y}_{T+1|T} = \alpha y_T + \alpha(1-\alpha) y_{T-1} + \alpha(1-\alpha)^2 y_{T-2} + \cdots$$

where $0 \leq \alpha \leq 1$ is the smoothing parameter. The one-step-ahead forecast for time T+1 is a weighted average of all the observations in the series y1…yT. The rate at which the weights decrease is controlled by the parameter α.

The method is suitable for univariate time series without trend and seasonal components i.e. for a stationary time series.

Double Exponential Smoothing

Double Exponential Smoothing is an extension to Simple Exponential Smoothing that explicitly adds support for trends in the univariate time series.

Double Exponential Smoothing is classically referred to as Holt's linear trend model. Holt extended simple exponential smoothing to allow forecasting of data with a trend. It is nothing more than exponential smoothing applied to both level i.e. the average value in the series and trend.

It requires two parameters called alpha (Smoothing factor for the level) and beta (Smoothing factor for the trend). In addition to the alpha parameter for controlling smoothing factor for the level an additional smoothing factor is added to control the decay of the influence of the change in trend called beta.

To express this in mathematical notation we now need three equations i.e. one for level, one for the trend and one to combine the level and trend to get the expected forecast $\hat{y}$

Forecast equation : $\hat{y}_{t+h|t} = \ell_t + h\, b_t$

Level equation : $\ell_t = \alpha y_t + (1-\alpha)(\ell_{t-1}+b_{t-1})$

Trend equation : $b_t = \beta*(\ell_t - \ell_{t-1})+(1-\beta)b_{t-1}$

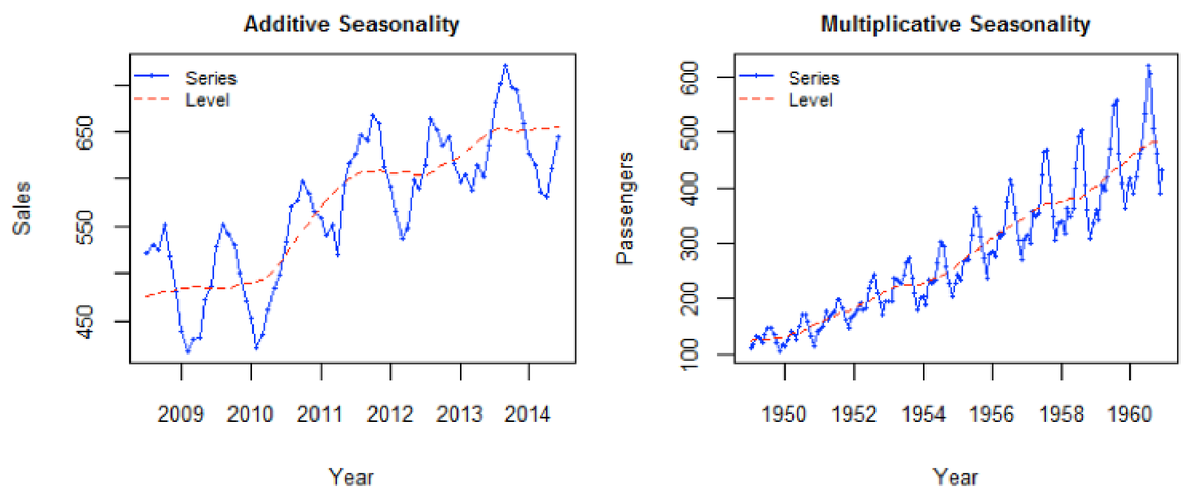
The level equation here shows that it is a weighted average of observation and the within-sample one-step-ahead forecast.

The trend equation shows that it is a weighted average of the estimated trend at time t based on $\ell(t)-\ell(t-1)$ and $b(t-1)$, the previous estimate of the trend.

We will add these equations to generate Forecast equation. We can also generate a multiplicative forecast equation by multiplying trend and level instead of adding it. When the trend increases or decreases linearly, additive equation is used whereas when the trend increases of decreases exponentially, multiplicative equation is used.

Additive Trend: Double Exponential Smoothing with a linear trend. Trend variation changes in the data stay roughly the same over time and don't fluctuate in relation to the overall data.

Multiplicative Trend: Double Exponential Smoothing with an exponential trend. Trend variation changes in relation to the overall changes in the data. So if the data is trending upward, the trend differences grow proportionally as well.



Each Time series dataset can be decomposed into its components which are Trend, Seasonality and Residual. Any dataset that follows a trend can use Holt's linear trend method for forecasting.

The method is suitable for univariate time series with trend and without seasonal components.

Triple Exponential Smoothing

Triple Exponential Smoothing is an extension of Double Exponential Smoothing that explicitly adds support for seasonality to the univariate time series. This method is sometimes called Holt-Winters Exponential Smoothing.

In addition to the alpha and beta smoothing factors, a new parameter is added called gamma that controls the influence on the seasonal component.

The Holt-Winters seasonal method comprises the forecast equation and three smoothing equations i.e. one for the level ℓt, one for trend bt and one for the seasonal component denoted by st with smoothing parameters α, β and γ.

$$\begin{aligned} \text{level} \quad L_t &= \alpha(y_t - S_{t-s}) + (1-\alpha)(L_{t-1} + b_{t-1}); \\ \text{trend} \quad b_t &= \beta(L_t - L_{t-1}) + (1-\beta)b_{t-1}, \\ \text{seasonal} \quad S_t &= \gamma(y_t - L_t) + (1-\gamma)S_{t-s} \\ \text{forecast} \; F_{t+k} &= L_t + kb_t + S_{t+k-s}, \end{aligned}$$

where s is the length of the seasonal cycle and 0 ≤ α ≤ 1, 0 ≤ β ≤ 1 and 0 ≤ γ ≤ 1.

The level equation shows a weighted average between the seasonally adjusted observation and the non-seasonal forecast for time t. The trend equation is identical to Holt's linear method. The seasonal equation shows a weighted average between the current seasonal index and the seasonal index of the same season last year (i.e., s time periods ago).

In this method also, we can implement both additive and multiplicative technique. Just like with the trend, the seasonality can be modeled as either an additive or multiplicative process for a linear or exponential change in the seasonality. The additive method is preferred when the seasonal variations are roughly constant through the series while the multiplicative method is preferred when the seasonal variations are changing proportional to the level of the series

Additive Seasonality: Triple Exponential Smoothing with a linear seasonality. Seasonality variation changes in the data stay roughly the same over time and don't fluctuate in relation to the overall changes

Multiplicative Seasonality: Triple Exponential Smoothing with an exponential seasonality. Seasonality variation changes in relation to the overall changes in the data. So if the data is trending upward, the seasonality differences grow proportionally as well.

Triple exponential smoothing is the most advanced variation of exponential smoothing and through configuration it can also develop double and single exponential smoothing models.

The method is suitable for univariate time series with trend and with seasonal components

Vector Autoregression (VAR)

Vector Autoregression (VAR) is a multivariate forecasting algorithm that is used when two or more-time series influence each other. The two basic requirements in order to use VAR are that there must be at least two timeseries and the two time series must influence each other. It is considered as an Autoregressive model because each time series of multivariate time series is modeled as a function of the past values i.e. the predictors are nothing but the lags (time delayed value) of the series.

Using Granger's Causality Test, it's possible to test this relationship before even building the model. Granger's causality tests the null hypothesis that the past values of time series (y1) do not cause the other series (y2). So, if the p-value obtained from the test is lesser than the significance level of 0.05, then you can safely reject the null hypothesis.

We have two variables y1 and y2. We need to forecast the value of these two variables at time t from the given data for past n values.For calculating y1(t), we will use the past value of y1 and y2. Similarly, to calculate y2(t), past values of both y1 and y2 will be used. For simplicity, I have considered the lag value to be 1. Below is a simple mathematical way of representing this relation:

$$y_1(t) = a_1 + w_1 11 * y_1(t-1) + w_1 12 * y_2(t-1) + e_1(t-1)$$

$$y_2(t) = a_2 + w_1 21 * y_1(t-1) + w_1 22 * y_2(t-1) + e_2(t-1)$$

where

a1 and a2 are the constant terms

w11, w12, w21, and w22 are the coefficients

e1 and e2 are the error terms

These equations are similar to the equation of an AR process. Since the AR process is used for univariate time series data, the future values are linear combinations of their own past values only. The following formulae explains the AR(1) process

y(t) = a + w*y(t-1) +e

In order to accommodate the multiple variable terms in each equation for VAR, we will use vectors. We can write the equations (1) and (2) in the following form:

$$\begin{bmatrix} y1(t) \\ y2(t) \end{bmatrix} = \begin{bmatrix} a1 \\ a2 \end{bmatrix} + \begin{bmatrix} w11 & w12 \\ w21 & w22 \end{bmatrix} \begin{bmatrix} y1(t\text{-}1) \\ y2(t\text{-}1) \end{bmatrix} + \begin{bmatrix} e1(t) \\ e2(t) \end{bmatrix}$$

The two variables are y1 and y2, followed by a constant, a coefficient metric, lag value, and an error metric. This is the vector equation for a VAR(1) process. For a VAR(2) process, another vector term for time (t-2) will be added to the equation to generalize for p lags:

$$\underset{K \times 1}{\begin{bmatrix} y1 \\ y2 \\ . \\ . \\ yk \end{bmatrix}} = \underset{K \times 1}{\begin{bmatrix} a1 \\ a2 \\ . \\ . \\ ak \end{bmatrix}} + \underset{K \times K}{\begin{bmatrix} w11 & . \\ w21 & . \\ . & . \\ . & . \\ wk1 & . \end{bmatrix}} \underset{K \times 1}{\begin{bmatrix} y1(t\text{-}1) \\ y2(t\text{-}1) \\ . \\ . \\ yk(t\text{-}1) \end{bmatrix}} + \ldots\ldots \underset{K \times K}{\begin{bmatrix} w'11 & . \\ w'21 & . \\ . & . \\ . & . \\ w'k1 & . \end{bmatrix}} \begin{bmatrix} y1(t\text{-}p) \\ y2(t\text{-}p) \\ . \\ . \\ yk(t\text{-}p) \end{bmatrix} + \underset{K \times 1}{\begin{bmatrix} e1 \\ e2 \\ . \\ . \\ ek \end{bmatrix}}$$

The above equation represents a VAR(p) process with variables y1, y2…yk. The same can be written as:

$$\begin{bmatrix} y \end{bmatrix} = \begin{bmatrix} a1 \end{bmatrix} + \begin{bmatrix} w1 \end{bmatrix} \begin{bmatrix} y1(t\text{-}1) \end{bmatrix} + \ldots\ldots \begin{bmatrix} wp \end{bmatrix} \begin{bmatrix} y1(t\text{-}p) \end{bmatrix} + \begin{bmatrix} e \end{bmatrix}$$

$$y(t) = a + w_1 * y(t-1) + ... + w_p * y(t-p) + \varepsilon t$$

The term εt in the equation represents multivariate vector white noise. For a multivariate time, series, εt should be a continuous random vector that satisfies the following conditions

Vector Autoregression Moving-Average (VARMA)

The Vector Autoregression Moving-Average (VARMA) method models the next step in each time series using an ARMA model. It is the generalization of ARMA to forecast multivariate time series.

The notation for the model involves specifying the order for the AR(p) and MA(q) models as parameters to a VARMA (p, q) function. A VARMA model can also be used to develop VAR or VMA models.

The method is suitable for multivariate time series without trend and seasonal components.

Vector Autoregression Moving-Average with Exogenous Regressors (VARMAX)

The Vector Autoregression Moving-Average with Exogenous Regressors (VARMAX) is an extension of the VARMA model that also includes the modeling of exogenous variables. It is a multivariate version of the ARMAX method.

Exogenous variables are also called covariates and can be thought of as parallel input sequences that have observations at the same time steps as the original series. The primary series(es) are referred to as endogenous data to contrast it from the exogenous sequence(s). The observations for exogenous variables are included in the model directly at each time step and are not modeled in the same way as the primary endogenous sequence i.e. through an AR, MA process.

The VARMAX method can also be used to model the subsumed models with exogenous variables such as VARX and VMAX.

The method is suitable for multivariate time series without trend and seasonal components with exogenous variables.

R Forecasting Models

Moving Average

A pure Moving Average model is one where Yt depends on the lagged values

Simple Exponential Smoothing

The Simple Exponential Smoothing technique is used for data that has no trend or seasonal pattern. Under simple exponential smoothing we weigh the recent values or observations more heavily rather than the old values or observations. The weight of each and every parameter is always determined by a smoothing parameter or alpha. The value of alpha lies between 0 and 1. When alpha is closer to 0 then it is considered as slow learning since the algorithm is giving more weight to the historical data. If the value of alpha is closer to 1 then it is referred to as fast learning since the algorithm is giving the recent observations more weight.

Holt Winters Exponential Smoothing

Holt-Winters Exponential Smoothing also known as Triple Exponential Smoothing is used for forecasting time series data that exhibits both a trend and a seasonal variation. The Holt-Winters Exponential Smoothing modifies the Holt Exponential Smoothing technique so that it can be used in the presence of both trend and seasonality. Holt-Winters Exponential Smoothing has a smoothing factor for the level, a smoothing factor for the trend and a smoothing factor for the seasonality

STD (Seasonal Trend Decomposition)

STL stands for Seasonal and Trend decomposition using Loess where Loess is a method for estimating nonlinear relationships. It is a statistical method of decomposing a time series data into 3 components containing seasonality, trend and residual.

To forecast with STL we first use STL to decompose the time series into three components. We then apply a standard forecasting algorithm to the remainder R(t) such as ARIMA or Exponential Smoothing, and generate an h-step ahead forecast for the remainder component R(t + h). Lastly, we calculate the h-step ahead trend component T(t + h) and S(t + h) and sum all three components to obtain a final forecast.

ARIMA (Autoregressive Integrated Moving Average)

Auto Regressive (AR) terms refer to the lags of the differenced series, Moving Average (MA) terms refer to the lags of errors and I is the number of difference used to make the time series stationary.

CES (State Space's Complex Exponential Smoothing)

CES has several advantages over conventional exponential smoothing models as it can model and forecast both stationary and nonstationary timeseries. Hence CES can capture both level and trend cases in the conventional exponential smoothing classification.

GUM (State Space's Generalised Exponential Smoothing Univariate Model)

GUM is the next step from CES and is a pure additive state-space model.

SSARIMA (State-space ARIMA or Several Seasonalities ARIMA)

SSARIMA constructs ARIMA in a state-space form estimating AR, MA terms and initial states and allows to model several seasonalities.

MSARIMA (Multiple Seasonal Autoregressive Integrated Moving)

MSARIMA constructs Multiple Seasonal State Space ARIMA estimating AR, MA terms and initial states.

TBATS (Trigonometric Exponential Smoothing State Space Model with Box-cox transformation, ARMA errors, Trend and Seasonal Components)

An innovative state space modeling framework is introduced for forecasting complex seasonal time series such as those with multiple seasonal periods, high-frequency seasonality, non-integer seasonality, and dual-calendar effects. The new framework incorporates Box–Cox transformations, Fourier representations with time varying coefficients and ARMA error correction. In addition, the proposed trigonometric formulation is presented as a means of decomposing complex seasonal time series and it is shown that this decomposition leads to the identification and extraction of seasonal components which are otherwise not apparent in the time series plot itself.

Under the hood TBATS will consider various alternatives and consider quite a few models such as with Box-Cox transformation and without it, with and without Trend, with and without Trend Damping, with and without ARMA (p,q) process used to model residuals, non-seasonal model, various amounts of harmonics used to model seasonal effects. The final model will be chosen using Akaike information criterion (AIC).

ETS (Error Trend Seasonal)

The ETS models are a family of time series models with an underlying state space model consisting of a level component, a trend component (T), a seasonal component (S) and an error term (E). ETS model uses exponential smoothing and this model is meant to be used if there is a trend and/or seasonality in the data as this model explicitly models these components

NNETAR (Neural Network Time Series Forecast)

NNETAR is a feed-forward neural network with a single hidden layer and lagged inputs for forecasting univariate time series.