Time Series Forecasting – Algorithms, Terms, Maths

1. **Sequence** of observations taken sequentially in time. Model fit on historic data to predict future data
2. **TS models assume Stationarity** – mean, variance are constant over time and Autocovariance that does not depend on time
3. **Most TS are non-stationary** – Trend (varying mean over time), Seasonality (variations over specific time frames), so need to transform
4. **Process** - Model the trend/seasonality, remove it, forecast, convert to original scale by applying trend/seasonality back
5. **Estimate, Eliminate Trend**
   - 5.1. Reduce trend – log, sq rt, etc. log transform penalises higher values
   - 5.2. Aggregation - average
   - 5.3. Smoothing – moving average. Pd.rolling_mean(), pd.ewma()
     - 5.3.1. Use log of values and then apply smoothing
   - 5.4. EWMA – exponential weighted ma. Higher weight to more recent values. Exponential applies a decay factor
     - 5.4.1. 'halflife' is used to define the amount of exponential decay
   - 5.5. Polynomial fitting – regression
6. **Eliminate Trend and Seasonality**
   - 6.1. Differencing – difference with a time lag
   - 6.2. Decomposition – model trend & seasonality and remove from the model
7. Use time lag to transform into supervised ML problem
8. Trends, seasonality (repeating pattern) in data (upward, downward)
9. Can't do time series - cyclical/sin series, random
10. White noise (no relationship). Smooth it by taking average
11. Rolling statistics – plot moving average, variance. Avg of time window
12. ADCF test – Augmented Dickey Fuller Test. Hypothesis for stationarity. Test statistic < critical value then reject null hypothesis. Stasmodels.tsa.stattools, adfuller
13. Check if series is stationary – use avg, movging avg, log to see trend
14. Rolling mean, std in python
15. .tsa.seasonal, seasonal_decompose – to see trend, seasonality, residuals
16. **ARIMA** forecasting for a stationary time series is nothing but a linear (regression) equation. PDQ are the three variables for the 3 terms. .tsa.arima_model, ARIMA
17. **AR** terms are just lags of dependent variable. For instance if p is 5, the predictors for x(t) will be x(t-1)….x(t-5). Yt = B0+B1Yt-1+B2Yt-2+…. Correlation between previous time period to current.
18. **MA** terms are lagged forecast errors in prediction equation. For instance if q is 5, the predictors for x(t) will be e(t-1)….e(t-5) where e(i) is the difference between the moving average at i^{th} instant and actual value. Error is smoothened by averaging.
19. **Integrated** - number of nonseasonal differences, i.e. in this case we took the first order difference. So either we can pass that variable and put d=0 or pass the original variable and put d=1. Both will generate same results.
20. **ACF, PCF** (Auto correlation, Partial Auto correlation) graphs for determining values for P, Q which is the no of AR and MA

21. **Autocorrelation Function (ACF):** It is a measure of the correlation between the the TS with a lagged version of itself. For instance at lag 5, ACF would compare series at time instant 't1'…'t2' with series at instant 't1-5'…'t2-5' (t1-5 and t2 being end points).
22. **Partial Autocorrelation Function (PACF):** This measures the correlation between the TS with a lagged version of itself but after eliminating the variations already explained by the intervening comparisons. Eg at lag 5, it will check the correlation but remove the effects already explained by lags 1 to 4.
23. From tsa.stattools, ACF, PCF
24. In this plot, the two dotted lines on either sides of 0 are the confidence interevals. These can be used to determine the 'p' and 'q' values as:
25. **p** – The lag value where the **PACF** chart crosses the upper confidence interval for the first time. If you notice closely, in this case p=2.
26. **q** – The lag value where the **ACF** chart crosses the upper confidence interval for the first time. If you notice closely, in this case q=2.
27. ARIMA - The p,d,q values can be specified using the order argument of ARIMA which take a tuple (p,d,q)
28. RSS is better with ARIMA vs AR or MA
29. Convert time series back to original scale.

1. **AR** - autoregression (AR) method models the next step in the sequence as a linear function of the observations at prior time steps. Suitable for univariate time series without trend and seasonal components.
2. **MA** - moving average (MA) method models the next step in the sequence as a linear function of the residual errors from a mean process at prior time steps. The notation for the model involves specifying the order of the model q as a parameter to the MA function, e.g. MA(q). For example, MA(1) is a first-order moving average model.
3. **ARMA** - The notation for the model involves specifying the order for the AR(p) and MA(q) models as parameters to an ARMA function, e.g. ARMA(p, q). An ARIMA model can be used to develop AR or MA models. Autoregressive Moving Average (ARMA) method models the next step in the sequence as a linear function of the observations and resiudal errors at prior time steps.
4. **ARIMA** - The Autoregressive Integrated Moving Average (ARIMA) method models the next step in the sequence as a linear function of the differenced observations and residual errors at prior time steps. It combines both Autoregression (AR) and Moving Average (MA) models as well as a **differencing pre-processing step of the sequence to make the sequence stationary, called integration (I).** The notation for the model involves specifying the order for the AR(p), I(d), and MA(q) models as parameters to an ARIMA function, e.g. ARIMA(p, d, q). An ARIMA model can also be used to develop AR, MA, and ARMA models. The method is suitable for univariate time series with trend and without seasonal components.
5. **SARIMA** - method models the next step in the sequence as a linear function of the differenced observations, errors, **differenced seasonal observations, and seasonal errors** at prior time steps. It combines the ARIMA model with the ability to perform the same autoregression, differencing, and moving average modeling at the **seasonal level**. The notation for the model involves specifying the order for the AR(p), I(d), and MA(q) models as parameters to an ARIMA function and AR(P), I(D), MA(Q) and m parameters at the seasonal level, e.g. SARIMA(p, d, q)(P, D, Q)m where **"m" is the number of time**

steps in each season (the seasonal period). A SARIMA model can be used to develop AR, MA, ARMA and ARIMA models. The method is suitable for univariate time series with trend and/or seasonal components.

6. **SARIMAX** - The Seasonal Autoregressive Integrated Moving-Average with Exogenous Regressors 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 (e.g. as an AR, MA, etc. process).
7. **VAR** – The Vector Autoregression (VAR) method models the next step in each time series using an AR model. It is the generalization of AR to multiple parallel time series, e.g. multivariate time series.
8. **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 multiple parallel time series, e.g. multivariate time series.
9. **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.
10. **SES** - The Simple Exponential Smoothing (SES) method models the next time step as an exponentially weighted linear function of observations at prior time steps.