Time Series:

1. EDA with Time Series Data
2. Error Trend seasonality (ETS), EMWA, ARIMA, ACF, PACF, SARIMAX
3. fb prophet
4. machine learning project with time series
5. Deep Learning - Time Series
6. RNN, GRU, LSTM, Bidirectional LSTM
7. CNN LSTM Encoder Decode, CNN

Time Series Analysis:

Time	Temp
5 am	54°F
6 am	57°F
7 am	58°F
8 am	58°F
9 am	60°F
10 am	62°F
11 am	64°F
12 PM	66°F

⇒ we generally deal with two types of time series:

1. univariate time series
2. Multivariate time series

Time	Temperature	cloud cover	dew point	humidity	wind
5:00 am	59 °F	97%	51 °F	74%	8 mph SSE
6:00 am	59 °F	89%	51 °F	75%	8 mph SSE
7:00 am	58 °F	79%	51 °F	76%	7 mph SSE
8:00 am	58 °F	74%	51 °F	77%	7 mph S
9:00 am	60 °F	74%	51 °F	74%	7 mph S
10:00 am	62 °F	74%	52 °F	70%	8 mph S
11:00 am	64 °F	76%	52 °F	65%	8 mph SSW
12:00 pm	66 °F	80%	52 °F	60%	8 mph SSW

→ inside any time series model, we will have time stamps such as

sec - sec
min - min
hr - hr
Day - Day
week - week
mnth - mnth
year - year

these can be considered to be independent variables.....

→ EDA is also an imp task even for this time series, followed by preprocessing and model building.

⇒ what models do we have in time series?

Statical

- ARIMA
- SARIMA
- SARIMAX
- VAR
- Exponential Smoothening
  - {ES}

w.r.t ML

- Xgboost

deeplearning

- RNN
- LSTM
- GRU
- N-beats
- deepAR

framework

- facebook (fbprophet)
- Google (temporal fusion transformer)

⇒ To evaluate a time series model:

- ⇒ AIC
- ⇒ BIC

} used for ARIMA mostly

→ MSE
→ RMSE
→ MAE

Note:

We dont have classification in time series, we only have regression.

EDA + Preprocessing ⇒ PACF, ACF

Partial Auto Correlation function – PACF
Auto Correlation function – ACF

Note:

→ We should only consider the integer values just like we do in ML

→ We neglect using obj data.

→ Time series analysis is extensively used to forecast company sales, product demand, stock market trends, agricultural production etc....

→ The fundamental idea for time series analysis is to decompose the original time series (Sales, stock market trend into several independent components. etc)

→ Typically, business time series are divided into the following four components:

Trend- overall direction of the series either upwards (or) downwards etc

Seasonality- montly (or) quarterly patterns

Cycle- long-term business cycles, they usually come after 5 (or) 7 years

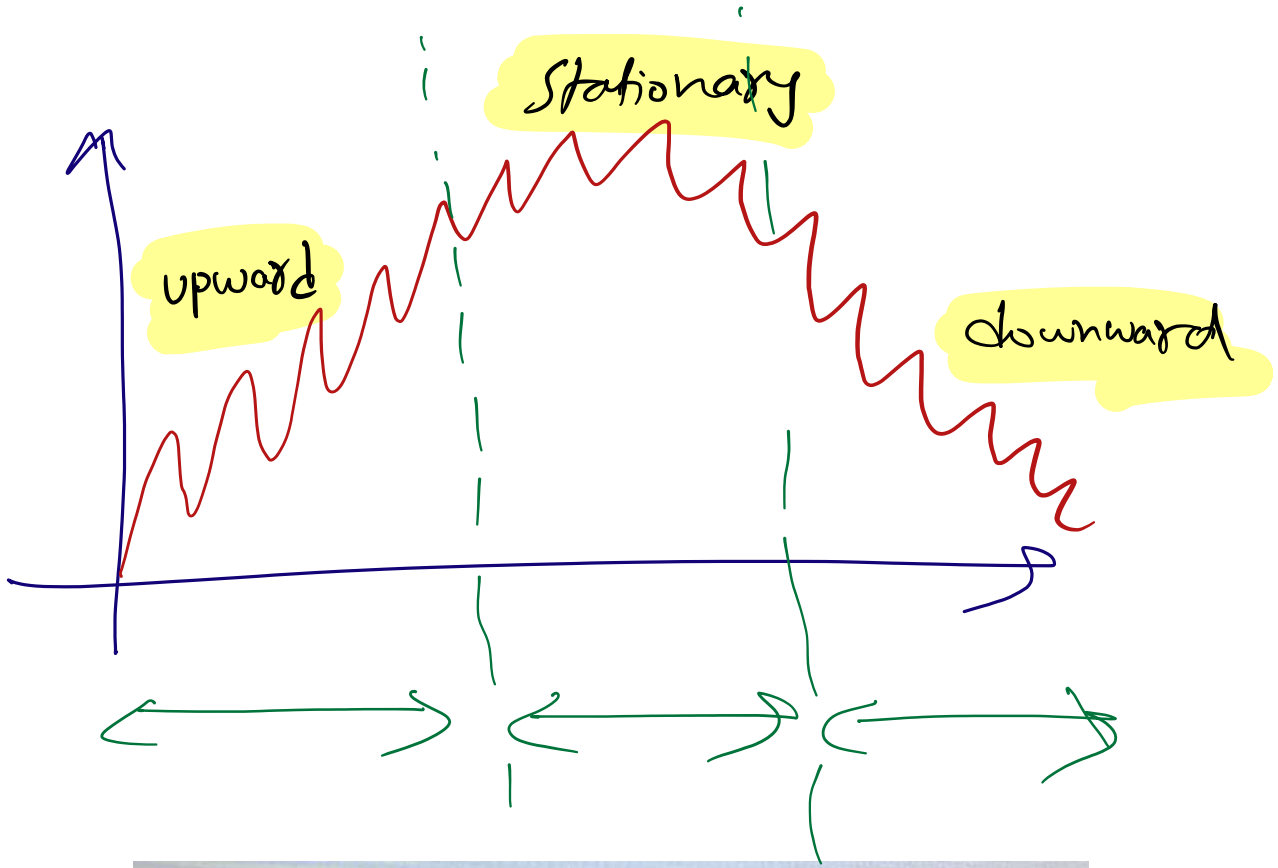
Irregular remainder- random noise left after extraction of all the components.

→ Interference of these components produces the final series.

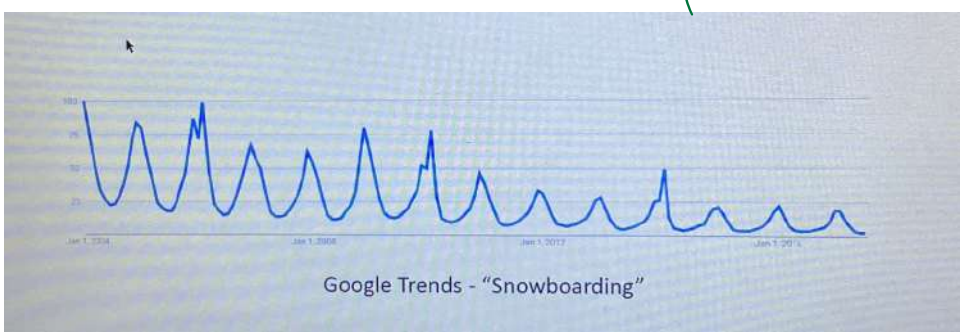
Why decomposing the original / actual time series into components?

⇒ It is much easier to forecast the individual regular patterns produced through decomposition of time series than the actual series.

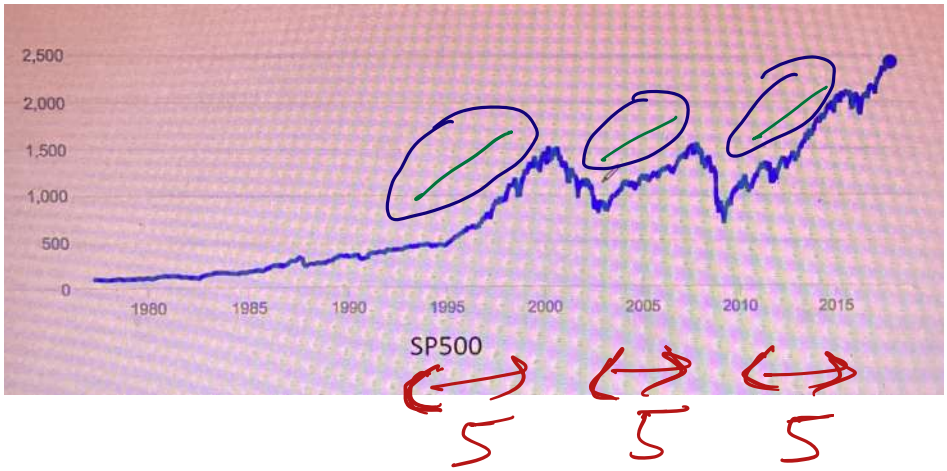
Trend:



Seasonality:



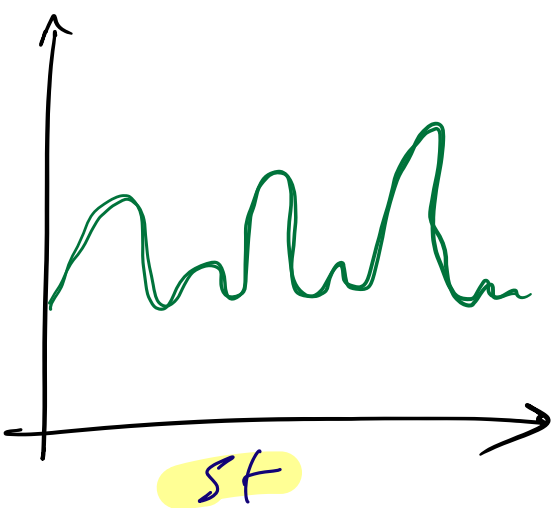
Cyclical - with no set repetition

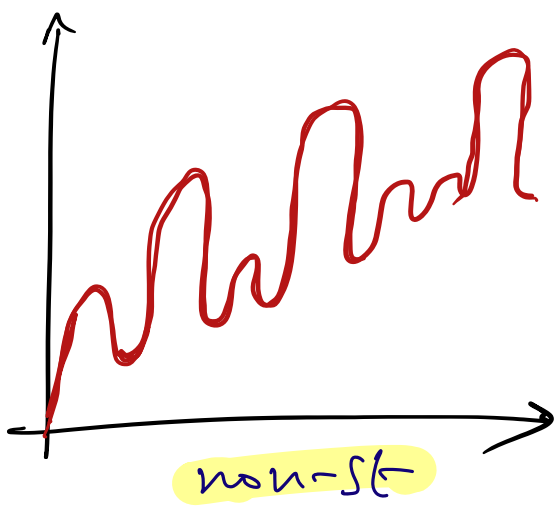


Stationary (vs) Non-Stationary Data:

→ To effectively use ARIMA, we need to understand stationary in our data.

→ A stationary series has **Constant mean** and **variance** over time.

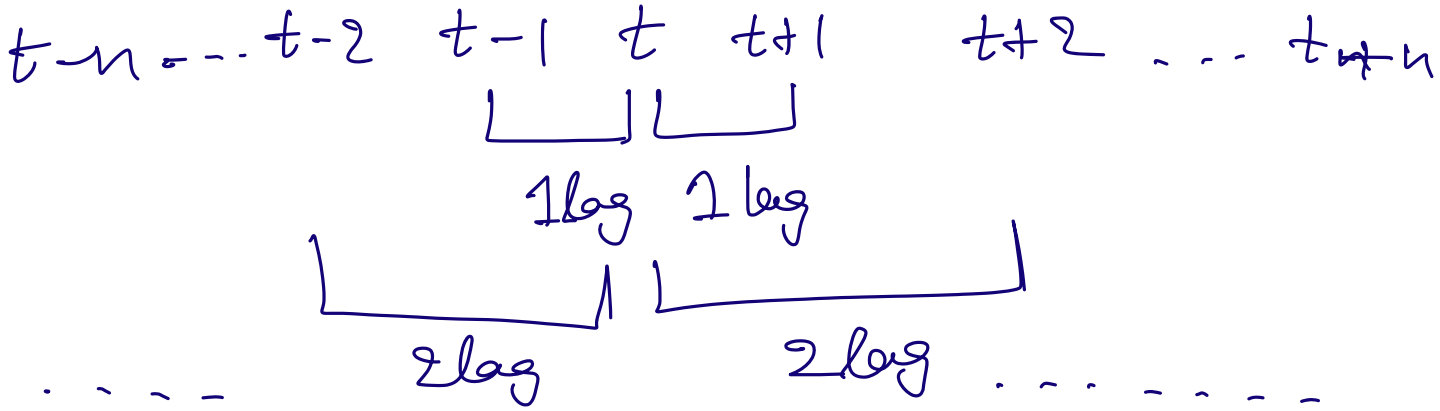




→ a stationary data set will allow our model to predict that the mean and variance will be the same in future.

→ we use a process called **differencing**

original data		First diff		2nd Diff	
Time 1	10	Time 1	NA	Time 1	NA
Time 2	12	Time 2	2	Time 2	NA
Time 3	8	Time 3	-4	Time 3	-6
Time 4	14	Time 4	6	Time 4	10



note: **lag tells us no. of steps**

→ you can continue differencing until you reach stationarity (which you can check both visually and mathematically)

→ we may loose some data

p	d	q
AR	I	MA

Trend Using MAs:

Moving Averages over time:

→ one way to identify a trend pattern is to use moving averages over a specific window of past observations

→ This smoothens the curve by averaging adjacent values over the specified time horizon (window)

The additive model is $Y(t)$

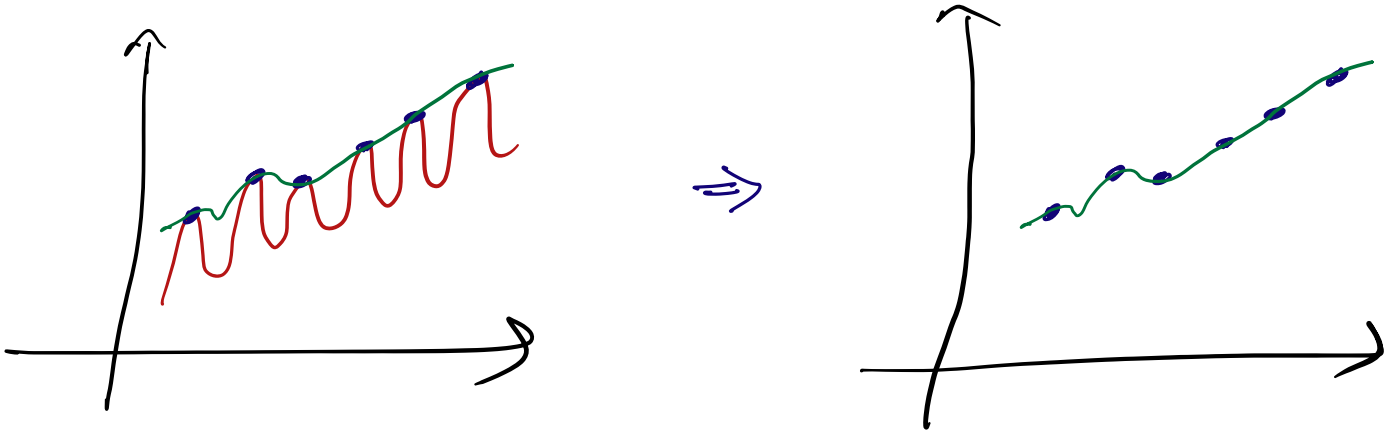
$$= T(t) + S(t) + e(t)$$

The multiplicative model is $Y(t)$

$$= T(t) * S(t) * e(t)$$

Rolling:

1. we prefer various window sizes, which will Consider that no. of values for the mean values
2. after we get the means, we try to connect them.
3. this will result us a smoothend line, for some particular window value.
4. we Consider that dataset



After rolling with specific mean of windows.

Augmented Dickey Fuller test:

ADF test is a common statistical test used to test whether a given time series is stationary or not.

→ it is one of the most commonly used statistical test when it comes to analyzing the stationary of a series.

→ there is a hypothesis testing involved with a null and alternate hypothesis and of a result a test static is computed and p-value get reported.

→ ADF test belongs to a category of tests called 'Unit Root Test', which is the proper method for

testing the stationary of a time series...

Unit Root ?

→ it is a characteristic of a time series which makes it non-stationary

for a eqn

$$Y_t = \alpha Y_{t-1} + \beta X_e + c$$

$\alpha = 1$ ⇒ called as unit root

Y_t = Value of time series at time 't'

X_e = exogenous variable

note:

→ if there is unit root means the series is non-stationary

Dicky-fuller test:

→ it is a null hypothesis where $\alpha = 1$ in the below eqn

$$y_t = c + \beta t + \alpha y_{t-1} + \phi \Delta y_{t-1} + e_t$$

y_{t-1} = lag 1 of time series

Δy_{t-1} = first diff of the series at time $(t-1)$

note:

So, there exists a null hypothesis, we need to neglect this in order to make it a stationary series..

ADF:

→ The ADF test expands the DF test eqn to include high order regressive process in model.

$$y_t = c + \beta t + \alpha y_{t-1} + \phi_1 \Delta y_{t-1} + \phi_2 \Delta y_{t-2} + \cdots + \phi_p \Delta y_{t-p} + e_t$$

→ adding more differencing terms, adds more thoroughness to the test.

note:

→ Since the null hypothesis assumes the presence of unit root, that is $\alpha = 1$ the p-value obtained should be less than the significance level (say 0.05) in order to reject the null hypothesis

→ thereby considering the series is stationary.

how to perform ADF in python?

→ Statsmodel package provides adfuller()

function in `statsmodels.tsa.stattools`

→ it returns the following o/p's

1. The p-value
2. The value of the test statistic
3. number of lags considered for the test
4. The critical value cut offs

Kpss test:

→ Kwiatkowski - phillips - schmidt - shin

→ is a type of unit root test that tells for the stationary of a given series around a deterministic trend.

→ it can't be used with interchanging ADF, cut there are few exceptions.

how to interpret kpss test results:

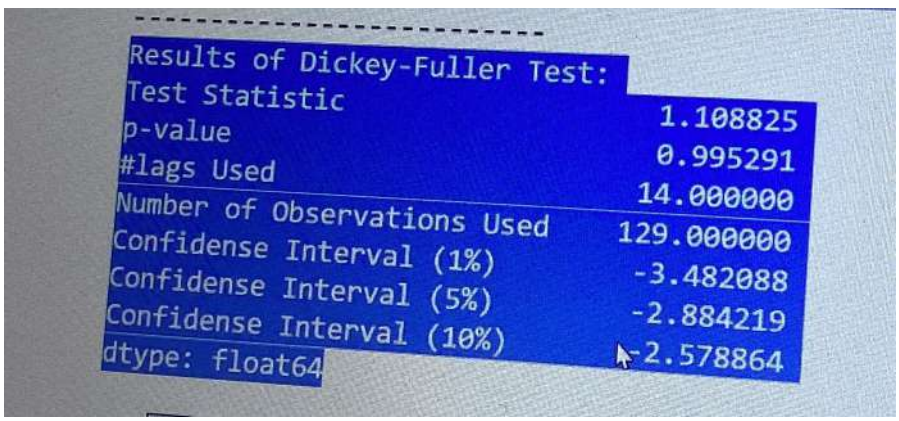
1. The KPSS Statistic
2. p-value
3. Number of lags used by the test
4. Critical values.

Diff b/w KPSS and ADF:

→ KPSS can check for stationality only in a deterministic trend.

→ The word deterministic implies the slope of the trend in the series does not change permanently.

→ even if the series goes through a shock, it tends to regain its original path.



note:

→ we know what is linear regression

$$y = mx + c$$ the best fit line

→ for multiple regression, we will be having

$$y = m_1x_1 + m_2x_2 + \cdots$$

→ now, what is AutoRegression (AR)

ex)

X

10 (t-9)
20 (t-8)
30 (t-7)
40 (t-6)
50 (t-5)
60 (t-4)
70 (t-3)
80 (t-2)
90 (t-1)
100 (t)

from these values we can draw few inferences....

if

10
20
30
40
50

(1)

= train data

⇒ 60 test data

(2)

20
30
40
50
60

train

70 test data

→ fly for other values

this will give us something like:

10	20	30	40	50	60
20	30	40	50	60	70
30	40	50	60	70	80
40	50	60	70	80	90
50	60	70	80	90	100

X | y

hypothesis func^n (or) line eq^n of ARIMA:

$$y_t = b + (t-1)m_1 + (t-2)m_2 + \cdots + (t-n)m_n$$

types of Moving Averages: (MA)

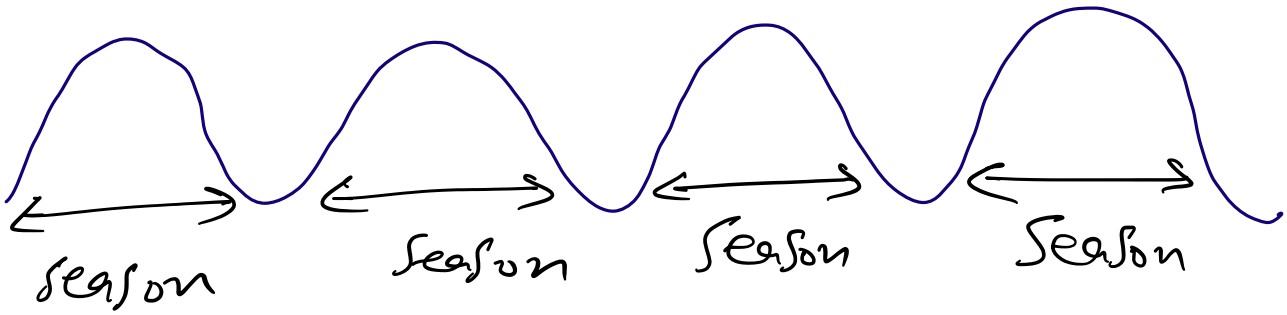
1. Simple Moving Average (SMA)
2. Exponential Moving Average (EMA)
3. Exponential weighted Moving Average (EwmA)

SMA ⇒ $y = \theta + c(\varepsilon_{t-1})$

const

SARIMA:

Seasonal ARIMA



↳ So, these seasons are being considered

as an additional factor, hence it is called SARIMA

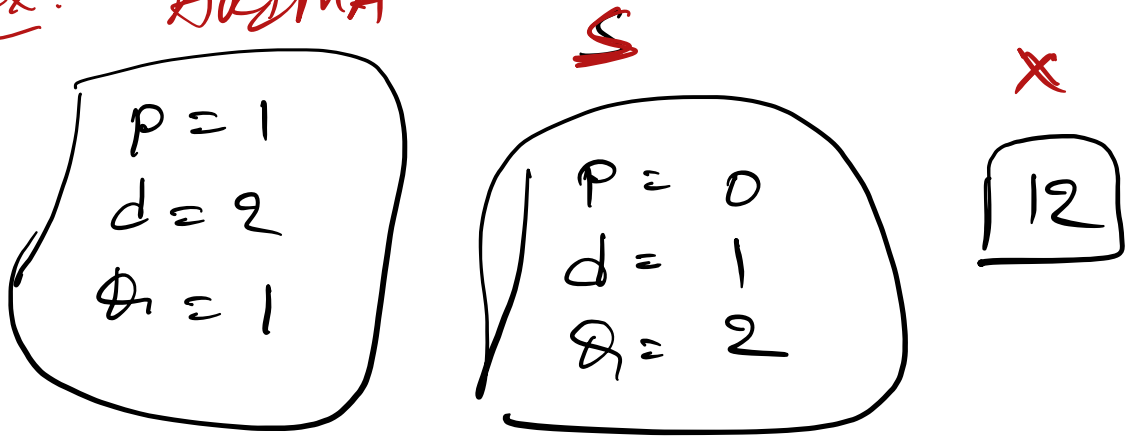
SARIMAX :

↳ along with the seasonal feature, we consider more features like, [multi] (or) any kind of classification given inside a season.

↳ This is called SARIMAX

X → frequency of season

ex: ~~ARIMA~~



⇒ roughly, (P,D,Q) $[P,D,Q,X]$

$= (1,2,1)\ [0,1,2,12]$

→ out of these we will select the respective permutation and combinat^{n} values.

ACF : [Auto correlation plot]

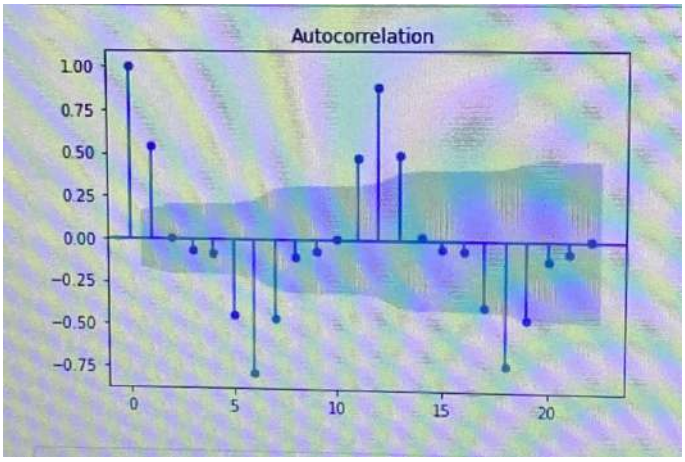
→ In EDA we find out correlation in a dataset

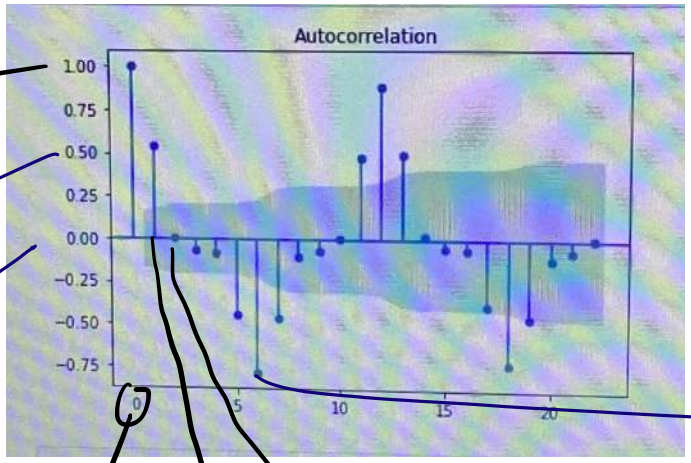
→ Auto correlation ~~means~~ finding correlation within the variables, just like we did for Auto Regressor among the values of a column

→ within a variable w.r.t time lag we find correlation.

ACF

Rough graph





1 correlation

0.5 correlation

almost '0' correlation

-ve Correlation

'0' time lag

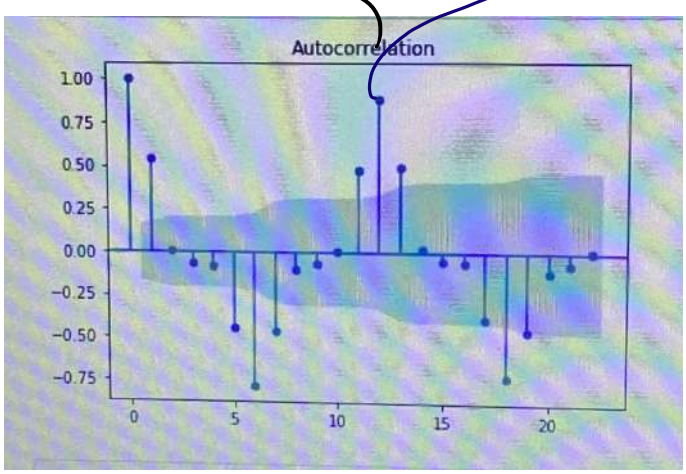
'1' time lag

'2' time lag

⇒ to perform this Autoregression

using the (p) value, we want best correlation value,

in this graph it is at point



this point is near to (1),

approx it is 0.80

So, for wt time lag do we have this corrln?

⟹ approx = 12

⇒ So, we consider

$p = 12$ roughly

⇒ We call this correlation graph as

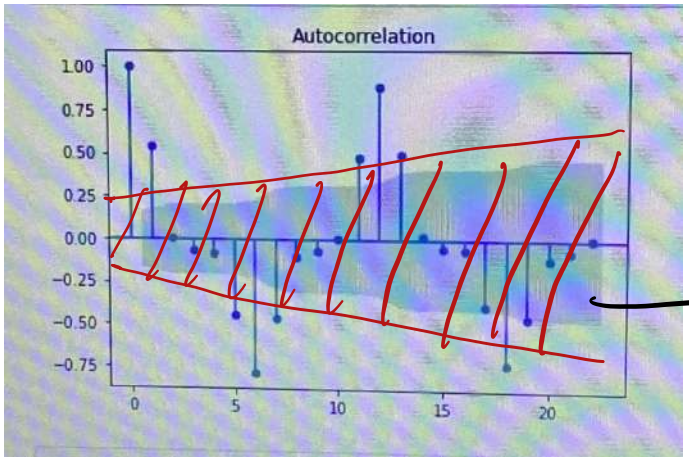
Corrologram

⇒ heat map is plotted for between the variables

$x_1 \longleftrightarrow x_2$

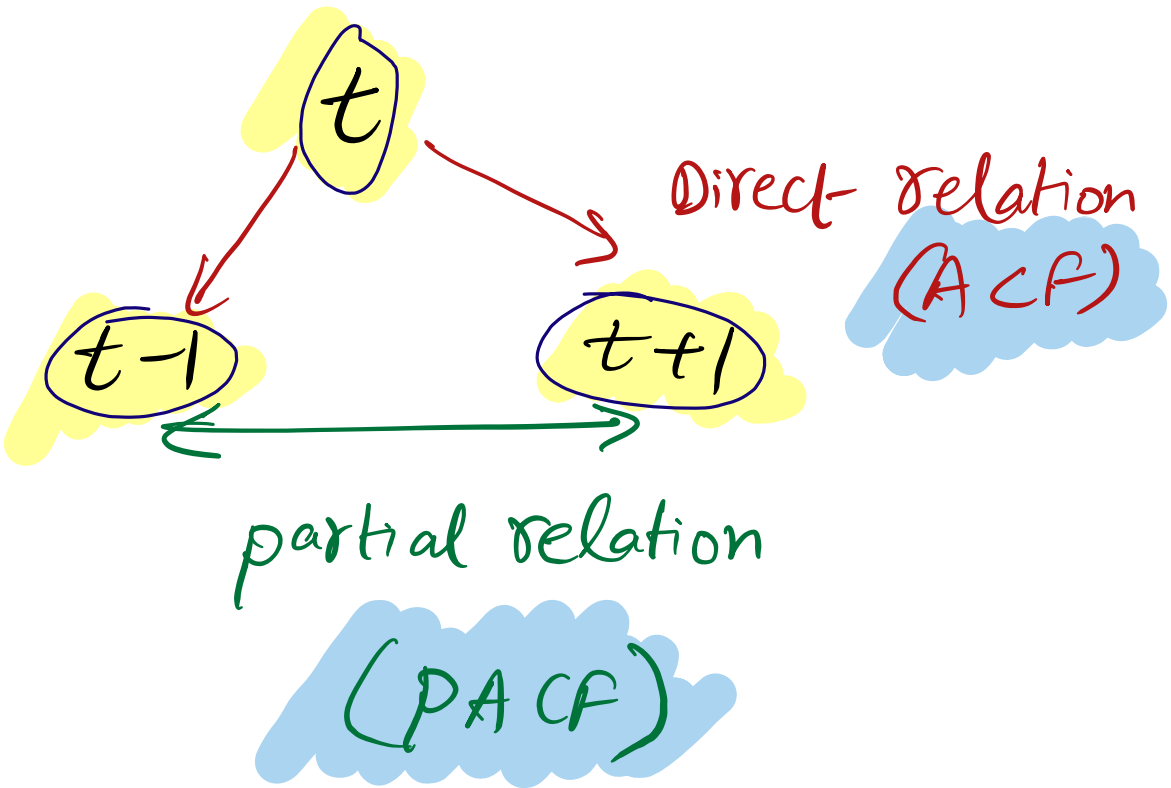
⇒ Corrologram is plotted within the variables

$x_1 \downarrow x_2 \downarrow x_3$

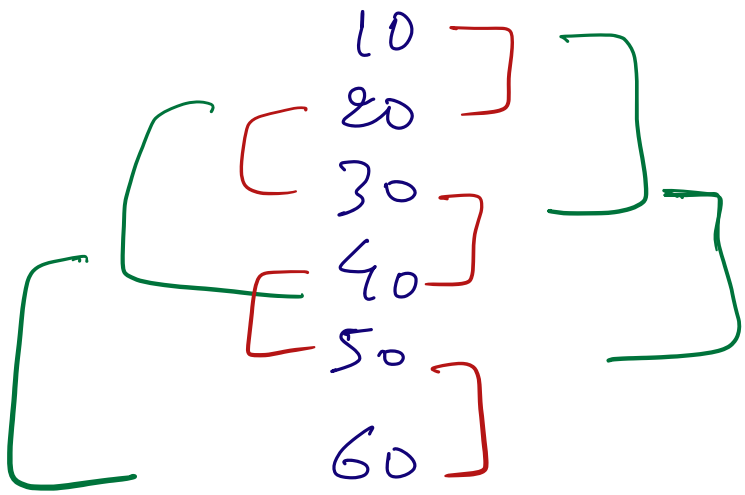


→ threshold region

⇒ that portion is increasing w.r.t increase in time lag.



w.r.t real world dataset:



AIC:

→ what is AIC?
→ when should you use it?
→ How should the results be interpreted?
→ pitfalls of AIC

(*) The Akaike information Criterion (AIC) is an estimator of out-of-sample prediction error there by relative quality of statistical models for a given set of data.

→ Given a collection of models for the data, AIC estimates the quality of each model, relative to each of the other models. Thus, AIC provides a means for model selection

⇒ In plain words, AIC is a single number score that can be used to determine which of multiple models is most likely to be best model for a given dataset.

⇒ they are only useful in comparison with other AIC scores for the same dataset.

⇒ A lower AIC score is better.

→ it works by evaluating the models fit on the training data, and adding a penalty term for the complexity

of the model (similar fundamentally to regularization)

$$AIC = -2\ln(L) + 2K$$

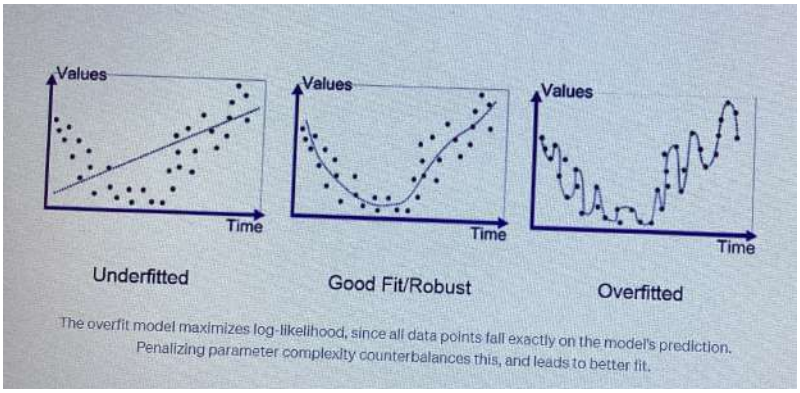
L = likelihood

K = # of parameters

ln = log-likelihood

→ AIC is low for models with high log-likelihoods (the model fits the data better, which is wt we want), but adds a penalty term for models with higher parameter complexity, since more parameters means a model is more likely to overfit to the training data.

⇒ refer the below image.



⑧ when?

→ AIC is typically used when you don't have access to out-of-sample data and want to decide b/w multiple diff model types (00) for time convinience.

⑧ Assumptions

⇒ use the same data b/w models

⇒ measuring the same outcome variable b/w models

$$p = \exp((AIC_{min} - AIC_i)/2)$$

"exp" means "e" to the power

BIC :

Bayesian information criterion (BIC)

→ When fitting models, it is possible to increase likelihood by adding parameter, but doing so may result in overfitting.

→ It resolves this problem by introducing a penalty term for the no. of parameters in the model.

→ The penalty term is larger in BIC than in AIC.

$$BIC = \ln(k) - 2\ln(L)$$

L = maximum likelihood

n = no. of data points

k = no. of free parameters to be estimated

→ models can be tested using corresponding BIC values.

[lower BIC value indicate lower penalty terms hence a better model]

Note:

BIC considers the no. of observations in the formula, which AIC does not.

Though BIC is always higher than AIC, lower the value of these two measures better the model.

→ AIC is best for prediction as it is asymptotically equivalent to Cross-validation

→ BIC is best for explanation as it

allows Consistent estimation of the underlying data generating process.

note!

Holt-Winters !!

→ it is a model of time series behavior.

→ it is a way to model three aspects of the time series:

1. a typical value (avg)
2. a slope (trend) over time
3. and a cyclical repeating pattern (seasonality)

Diff b/w Holt and Holt-winters?

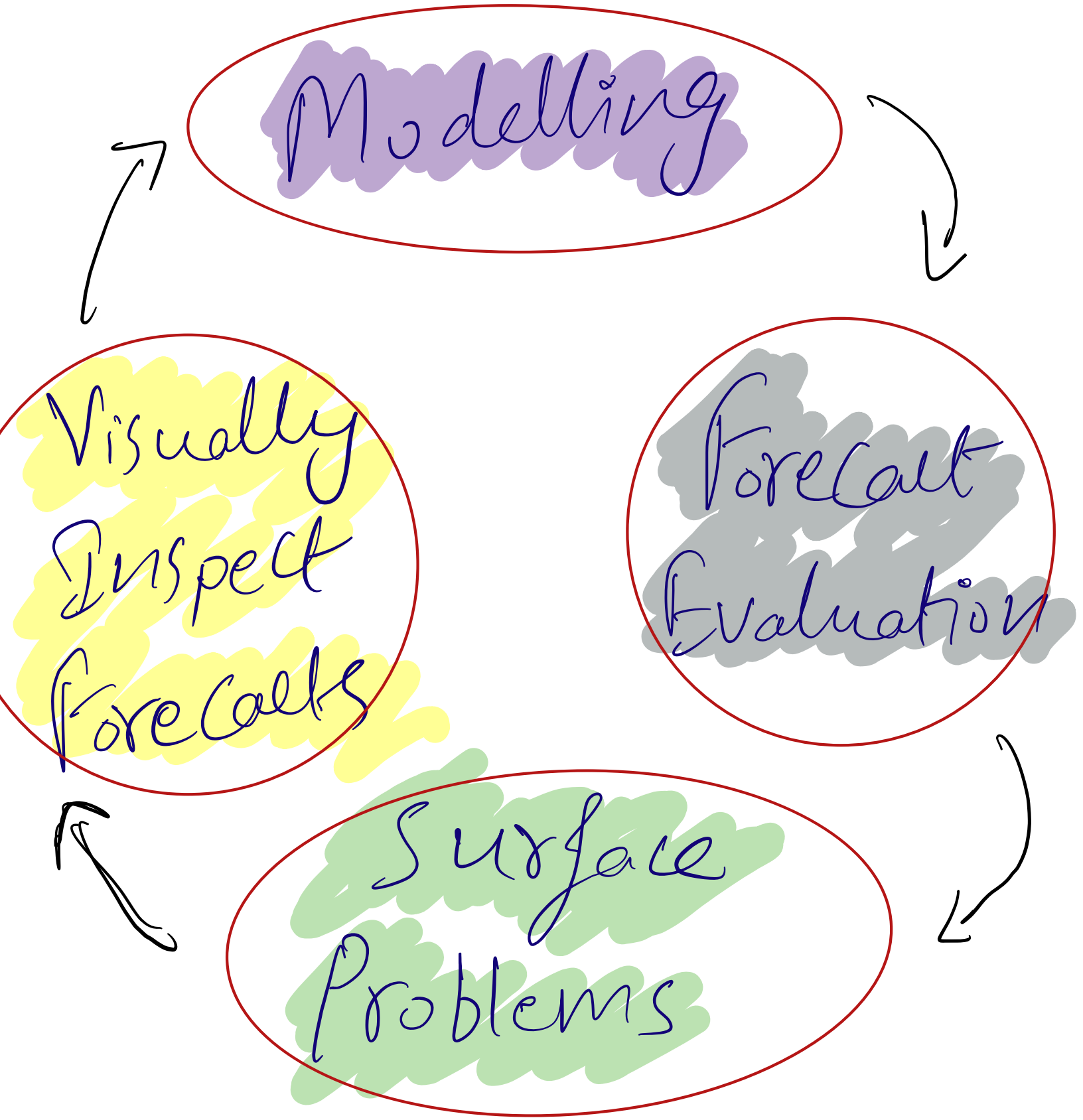
Holt: Exponential smoothing with a
→ trend component.
→ double exponential smoothing.

Holt-Winters:

Exponential smoothing with a trend component and a seasonal component.

→ triple exponential smoothing.

⇒ Holt-winters have extra parameters which ARIMA dont have.


Modelling
Forecast Evaluation
Surface Problems
Visually Inspect Forecasts

FBprophet:

→ it is an open source algorithm for generating time-series models that uses a few old ideas with some new twists.

→ it is particularly good at modeling time series that have multiple seasonalities and doesn't face some of the above drawbacks of other algorithms.

→ mainly used by FB to forecast

→ it is a procedure for forecasting time series data based on additive model where non-linear trends are fit with yearly, weekly, and daily seasonality, plus holiday effects.

→ It works best with time series that have strong seasonal effects and

Several seasons of historical data.

prophet is robust to missing data and shifts in the trend, and typically handles outliers well.

→ prophet uses '3' features trend, seasonality, holidays. to decompose the time series.

$$y(t) = g(t) + s(t) + h(t) + e(t)$$

$g(t)$ = trend models non-periodic changes

$s(t)$ = Seasonality presents periodic change

$h(t)$ = ties in effects of holidays

$e(t)$ = covers idiocentric changes not accommodated by the model.

The eqn can also be written as,

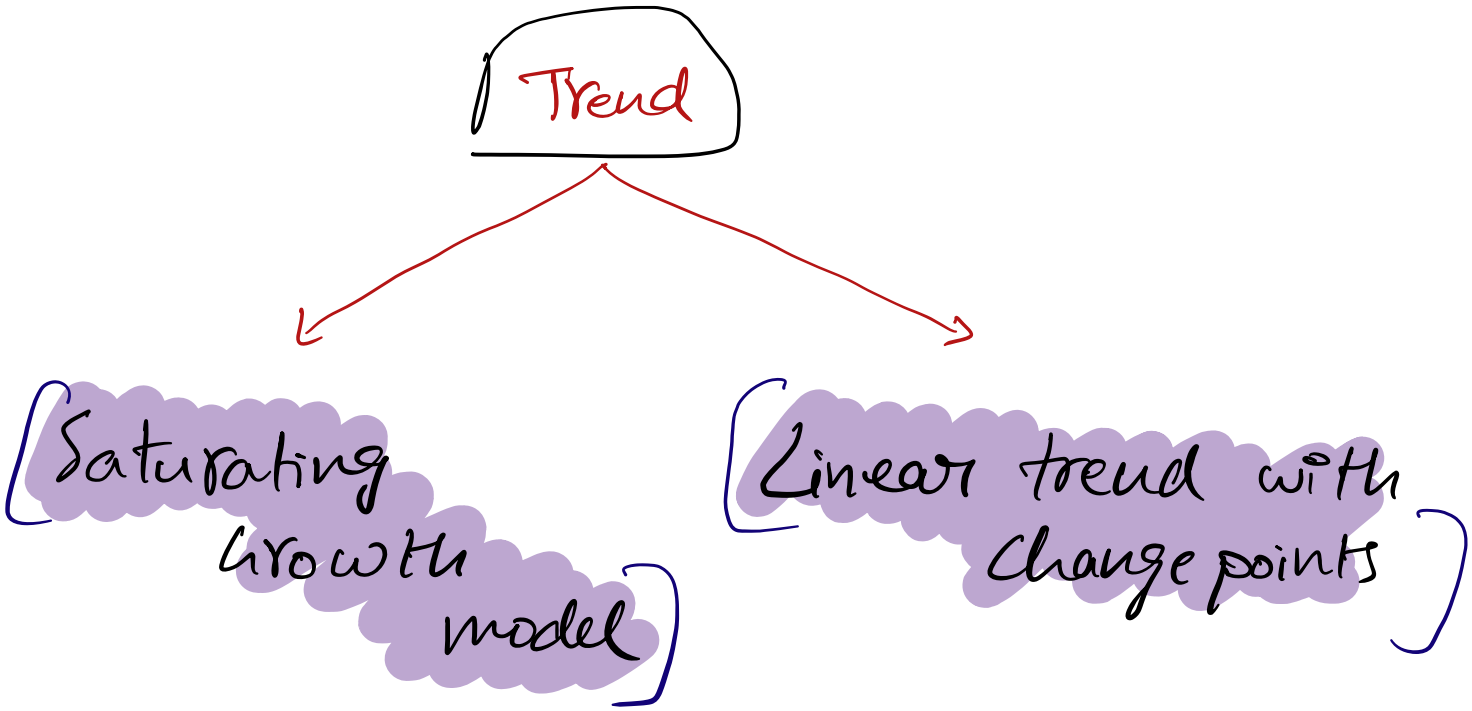
$$y(t) = \text{piecewise\_trend}(t) + \text{seasonality}(t) + \text{holiday\_effects}(t) + \text{i.i.d noise.}$$

Note:

modelling seasonality as an additive component is the same approach taken by exponential smoothing

GAM formulation has the advantage that it decomposes easily and accommodates new components as necessary, for instance when a new source of seasonality is identified.

⇒ prophet is essentially "framing the forecasting problem as a curve-fitting excercise" rather than looking explicitly at the time based dependence of each observation.



Saturating Growth model:

$$g(t) = \frac{C}{1 + \exp(-k(t-m))}$$

C is the carrying capacity
K is the growth rate
m is an offset parameter

→ Carrying capacity is not constant as the number of people in the world who have access to the internet increases, so does the growth ceiling.

$(C_t) \longrightarrow (C)$

Varying Capacity Constant Capacity

→ Growth rate is also not constant cuz, as the time passes, the learning also increase.

→ offset parametre, gives us a sigmoid curve so that they dont exceed limits like logistic regression

Linear trend with change points:

$$g(t) = (k + a(t)^T \delta)t + (m + a(t)^T \gamma)$$

k is the growth rate
δ has the rate adjustments
m is the offset parameter

⇒ The uncertainty intervals are, however a useful indication of the level of uncertainty, and especially an indicator of over fitting.

→ changepoints - tells us about over fitting
changepoint_prior_scale - can be used to adjust trend flexibility, we want higher value

Seasonality:

→ The seasonal component $S(t)$ provides an adaptibility to the model by allowing periodic changes based on sub-daily, daily, weekly and yearly seasonality.

→ Business time series often have multi-period seasonality as a result of the human behavior they represent.

→ prophet relies on **Fourier Series** to provide a mellable model of periodic effects.

(P) is regular period of the time series.

→ approximate arbitrary smooth seasonal effects is therefore tied

in with a standard fourier series.

$$s(t) = \sum_{n=1}^{N} \left(a_n \cos\left(\frac{2\pi n t}{P}\right) + b_n \sin\left(\frac{2\pi n t}{P}\right) \right)$$

⇒ Truncating the series at N applies a low pass filter to the seasonality. So, with increased risk of overfitting, increasing N allows for fitting seasonal patterns that change more quickly.

Holidays and Events

→ Impact of a particular holiday on the time series is often similar

year after year, making it an important incorporation into the forecast.

→ To utilize this feature, the user needs to provide a custom list of events. Fusing this list of holidays into the model is made straightforward by assuming that the effects of holidays are independent.

→ It is often important to include effects for a window of days around a particular holiday.

⇒ To account for that we include additional parameters for the days surrounding the holiday.

⇒ Essentially treating each of the days in the window around the

holiday as a holiday itself.

Conclusion:

- Prophet was engineered to help analysts with a variety of backgrounds produce more forecasts with less time invested towards doing so.

- This was done by sticking to a relatively plain model

- We use a simple, modular regression model that often works well with default parameters, and that allows analysts to select the components that are relevant to their forecasting problem and easily make adjustments as needed.

Exponential Smoothing:

→ It is one of the most widely used time series forecasting methods for univariate data

→ It is often considered as an alternative of Box – Jenkins ARIMA class of methods for time series forecasting

→ It is similar to Simple Moving Average

diff?

Simple moving averages consider past observations equally, whereas exponential smoothing assigns exponentially decreasing weights overtime.

→ Means, {ES} places a bigger emphasis on more recent observations, providing a weighted average.

'3' types of {ES} methods:

① Single Exponential Smoothing:

→ It is used for time-series data with no seasonality (or) trend.

→ It required a single smoothing parametre that controls the rate of influence from historical observations.

→ if the values are closer to (1) that means the model pays little attention to past observations

② Double exponential smoothing:

→ it is used for time-series data with no seasonality - but with trend.

→ with an additional smoothing factor to control the influential decay of the change in trend,

supporting both linear and exponential trends. → (additive)

↳ (multiplicative)

@ Triple exponential smoothing:

→ AKA Holt-Winters Exponential smoothing, is used for time-series data with a trend and seasonal pattern.

→ This is built on previous two techniques with a third parameter that controls the influence on the seasonal component.

Single exponent smoothing:

$$S_{t+1} = \alpha y_t + (1-\alpha) s_t$$

S_{t+1} is the predicted value for the next time period

S_t is most recent predicted value

y_t is the most recent actual value

α is the smoothing factor b/w 0 and 1

Note:

when alpha is closer to 1 more emphasis is placed on the most recent actual value

when alpha is closer to 0 the level of smoothing is stronger and the prediction is less sensitive to recent changes.

Hyperparameters:

Double ES

(1) Alpha : Smoothing factor for the level
Beta: Smoothing factor for the trend
Trend type: Additive (or) multiplicative
Dampen type: Additive (or) multiplicative
phi: Damping coefficient.

Triple ES

(2) Alpha: smoothing factor for the level
Beta: Smoothing factor for the trend
Gamma: Smoothing factor for the seasonality
Trend type: Additive (or) multiplicative
Dampen type: Additive (or) multiplicative
phi: Damping co-efficient
Seasonality type: Additive (or) multiplicative
period: Time steps in seasonal period

VAR (Vector Autoregressive model)

→ Vector Autoregression (VAR) is a multivariate forecasting algorithm that is used when two (or) more time series influence each other.

→ they are diff from univariate autoregressive models because they allow feedback to occur b/w the variables in the model.

Reduced form VAR models:

Consider each variable to be a function of:

⇒ Its own past values
⇒ The past values of other variables in the model.

Recursive VAR models:

→ Contain all the components of the reduced form model, but also allow some variables to be function of other concurrent variables.

⇒ allows us to have some Structural shocks.

Structural VAR models:

→ Include restrictions that allow us to identify causal relationships beyond those that can be identified with reduced form or recursive models.