**What is exponential smoothing and why is it used in Time Series Forecasting?**

Exponential Smoothing (ES) is a technique for smoothing time series data using the exponential window function. Whereas in a simple moving average, the past observations are weighted equally, exponential function are used to assign exponentially decreasing weights over time. The model’s base assumption is that the future is roughly similar to the recent past. The other assumption is that prediction is a weighted linear sum of past observations, thus the effect of the past is taken into account by using weights. It is one of the many window functions commonly used to smooth data by removing high-frequency noise.

It was first suggested by Robert G. Brown in 1956, then expanded by Charles Holt in 1957.

The simplest form of exponential smoothing is given by –

S(t) = alpha x(t) + (1 – alpha) S(t-1)

Where alpha is between 0 and 1 (with 0 and 1 both inclusive).

There is no formally correct procedure for choosing alpha. One can use formal experience or judgment or one can estimate it using the method of least squares. From the above, it’s clear S(t) is a simple weighted average of the current observation x(t) and the previous smoothed statistic S(t-1). Exponential smoothing parameters can be adjusted to change how much or how little the previous observations will be allowed to affect the current smoothed statistic S(t). This technique also does not require a minimum number of observations to be made before it can produce smoothing, but that being said -

There are three types of Exponential Smoothing –

1. Simple or Single Exponential Smoothing – this is the method of time series forecasting used with univariate data with no trend and no seasonal pattern.
2. Double Exponential Smoothing -
3. Triple Exponential Smoothing

**What is stationarity? What is seasonality? Why Is Stationarity Important in Time Series Forecasting?**

A stationary process is a stochastic process whose mean and variance does not change over time or to put it in another words, doesn’t change over time. Overall, the trend isn’t up or down. Neither is the variance increasing or decreasing. A common example of a stationary process is white noise.

The most common violation of stationarity is a trend in the mean, which can be because of external factors or ‘shock’ to the time series system. After the ‘shock’ the mean does not return to normal.

Stationarity is important assumption in Time Series Forecasting because it’s an important assumption and many statistical tests and models rely on it. If we visualize a times series and can see big noticeable and irregular trends (specifically, changing mean trends) or increasing variance, we can guess the time series is not stationary.

A seasonality effect is the presence of variations in data that occur at regular intervals – such as weekly or monthly or quarterly. This may be due to various factors such as vacations or holidays, or seasons. The characterizing factor is that this is periodic and repetitive and regular in the time series.

**How is seasonality different from cyclicality? Fill in:**Seasonal is predictable, whereas cyclical is not.

Cyclical behavior is different from seasonal behavior. If economic recessions are followed by economic expansions and though these are expected cycles, it is not possible to know the cycles or their arrival or their frequency or the overall length of the fluctuations in advance.

On the other hand, “seasonal” is connected with some aspect of weather (ice-cream sells less in winter and coats and jackets sell less in summer) or calendar (say holidays, like shopping during Thanksgiving is higher than any other time of the year, but this isn’t unexpected). These are seasonal fluctuations and are fixed and of a known period. Electricity consumption during summer months and winter months are high and this is known to be so. Home sales are normally higher before school year starts. This too is a known factor. Cyclical patterns however is generally not of a fixed period. Business cycles can fluctuate and do so, but it is not known beforehand how long it is going to be.

**How does the Prophet Algorithm differ from an LSTM?**

First of all, time series data is different from other sequential data like text or audio or protein sequences. The time series data has the time component, which often brings significant information when predicting the future. And although LSTM is a powerful RNN model, known for text, audio, or other sequential data prediction, its main task it to learn the key parts of the sequence and forget the less important ones using the gates. The gates have different functions – how to combine new input with the past, how to forget or not, how and what to predict for the next time step. To do this, an LSTM model must use cells to capture the structure of the sequence. Too few cells do not capture the time series, whereas too many overfits. It is a non-trivial task to consider how many parameters an LSTM needs to fit time series. So, LSTM requires careful tuning, especially of hyperparameters, to get good results. Furthermore, the LSTM needs massive datasets to understand complex patterns, so smaller time series datasets may not work to its advantage. LSTM is complex, massive algorithm too advanced for small datasets and prone to overfitting.

On the hand, Prophet Algorithm developed by Facebook was specifically designed for in-house business applications in time series prediction. It is targeted to certain domains and not to others. Its advantage is that it doesn’t need complicated tuning like an LSTM model. It is an additive model

Y(t) = g(t) + s(t) + h(t) + e(t)

Where –

Y(t) = time series target prediction at time (t)

g(t) = Represents the trend when the objective is to capture the general trend of the series. If the sale of donuts is expected to increase in Austin, TX during Austin City Live and South by Southwest, how many donuts are expected to be sold during those specific months out of the year? What is the trend?

s(t) = Seasonality component – the two big festivals in Austin are during specific seasons – spring and autumn. This is important to take into account as during summer months, it is blazing hot and most of the folks stay indoors.

h(t) = Holiday component.

e(t) = Error terms that follows a normal distribution N(0, s) with zero mean and an unknown variance that has to be derived from the data.

**Why does an LSTM have poor performance against ARIMA and Prophet for Time Series?**

ARIMA (Auto-Regressive Integrative Moving Average) is a class of time series prediction models. The core of ARIMA is a mathematical model that represents the time series values using its past values. There are two key features of ARIMA models are –

* Past Values: Which are good predictors of future values. The issue is the number of past values to use. The model uses last p time series values as features, where p is a number to be determined.
* Past Errors: The model can use the information on how well it has performed in the past. Thus, we add as features the most recent q errors the model has made. Like p above, q is also a hyperparameter.

The time series is standardized so that the model is independent from seasonal or temporal trends, that is, the model is to be trained on a stationary time series. This means (as stated before) that the properties of the process generating the time series do not change over time. The series changes over time but the way it changes does not change over time.

ARIMA is defined by three parameters – p, d, and q that describe the three main parts of the model. The parameter p gives us how many past values to consider for the expression of the current value. The past number of forecast errors to be considered is given by q. And finally, the number of differences needed to achieve stationarity is given by the parameter d.

LSTM usually performs badly compared to ARIMA and Prophet because although it is excellent in learning from sequential data, time series is only a special case of it. For smaller datasets LSTM are not able to detect patterns (as stated before). The LSTM do not rely on assumptions about the data like stationarity, but the hyperparameters also need tuning.