Lecture Notes

Time Series Forecasting Using Smoothing

In the previous sessions, you looked at two methods commonly used for analysing time series, i.e. classical decomposition and ARIMA modelling.

In this session, you learnt how smoothing techniques can be used to make forecasts for a time series.

Time Series Forecasting - A Quick Overview

In this session, you started by getting an overview of the various techniques that are used to make a forecast. The different techniques used for this purpose are -

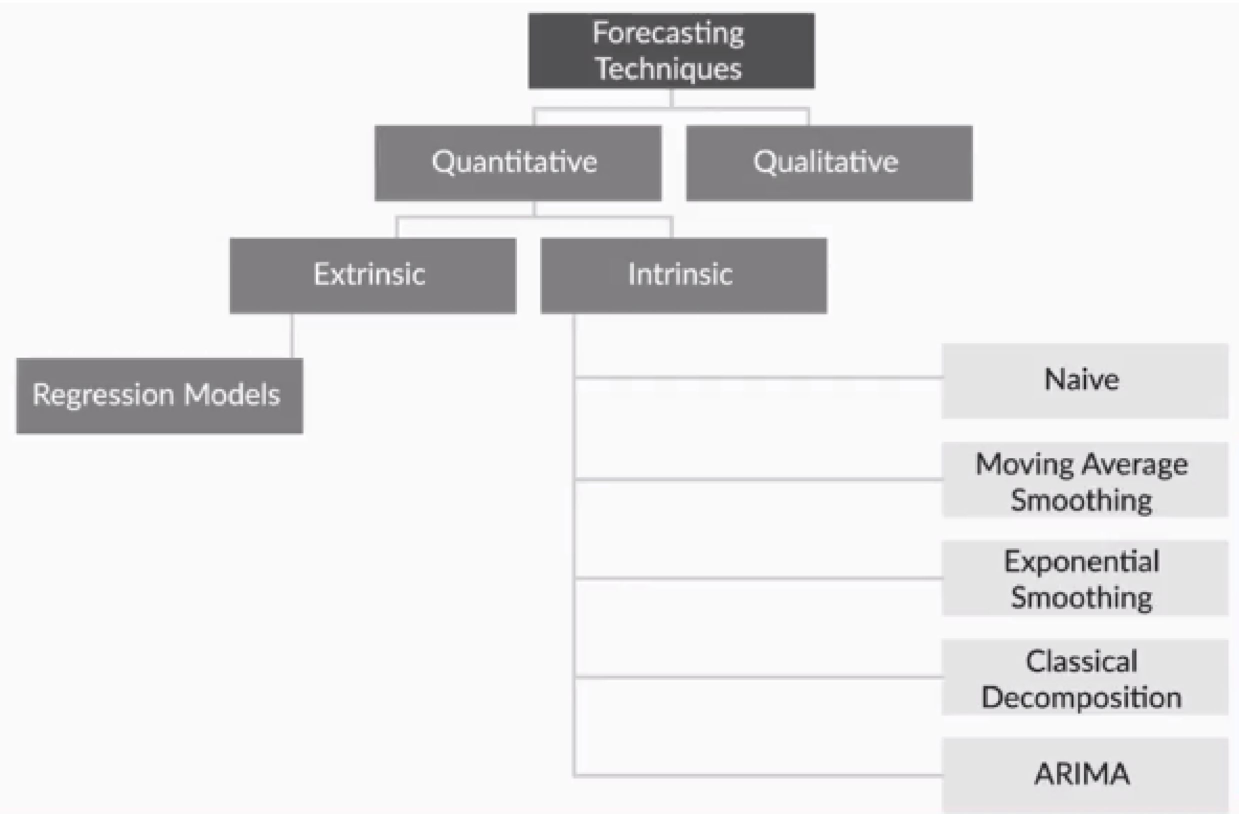

Figure 1 - Forecasting Techniques - An Overview

Then, you saw when in the product cycle. which of these techniques should be used.

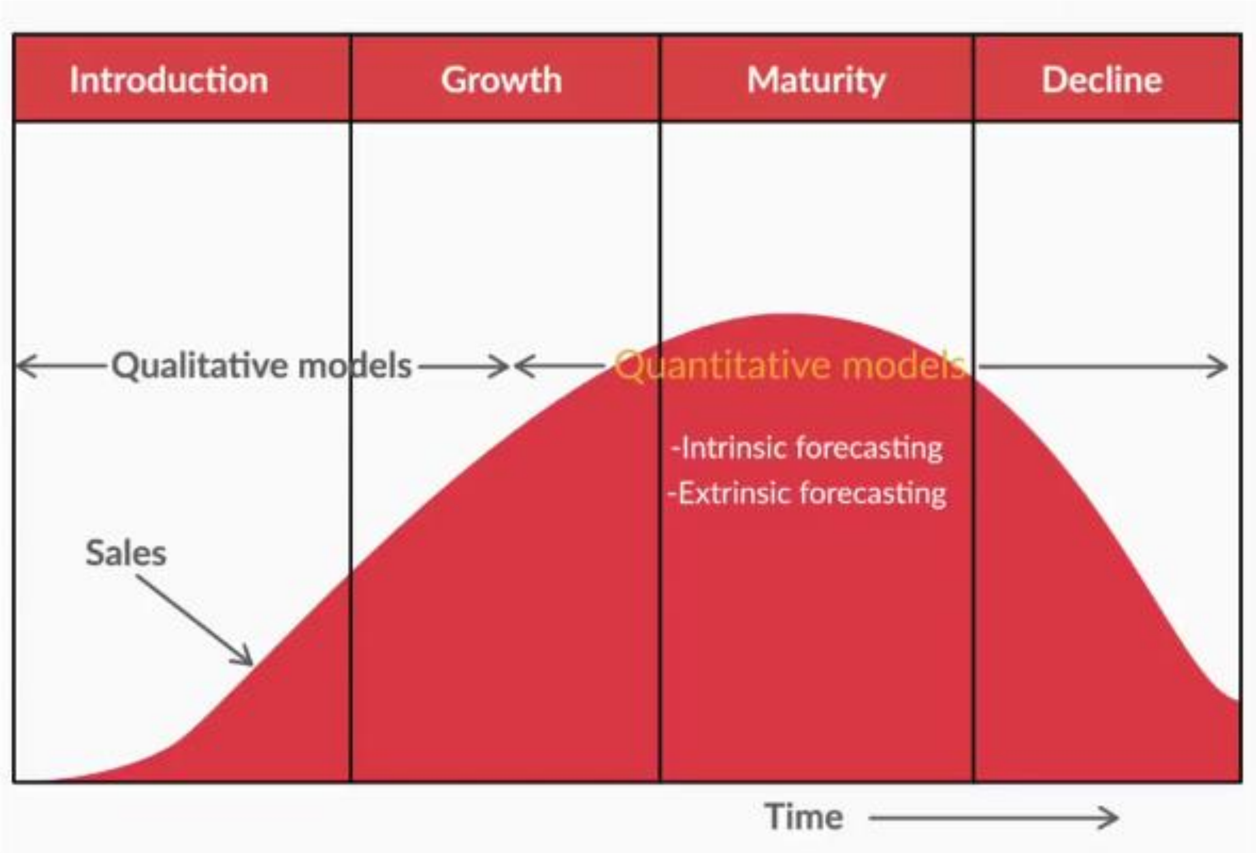

Figure 2 - Forecasting Techniques According to Uses in Product Cycle

Quite simply, the **qualitative models** are used when there is very **little data** to make a forecast. This usually happens early on in a product's lifecycle. In an **intermediate stage**, you would have information about the product, like competitor's pricing etc., but you would still not have enough past data. In that case, you would use **extrinsic forecasting** to make a forecast. However, once you have **enough past data**, you can use **intrinsic methods** of forecasting, such as ARIMA or classical decomposition to make a pretty good forecast.

Naïve and Average Methods for Time Series Forecasting

Then you saw how smoothing techniques can actually be used to make forecasts. The different techniques that can be used to make forecasts are -

1. **Naive Method** -
   - Forecast = Last month's sales
2. **Average Methods** -
   - **Simple Average Method** -
     - Forecast = Average of all past months' sales

- **Moving Average Method** -
  - Forecast = Average of last 3 (or 4, or n) months' sales
- **Weighted Average Method** -
  - Forecast = Weighted average of last 3 (or 4, or n) months' sales

3. **Exponential Smoothing Methods** -
   - **Simple Exponential Smoothing Method** -
     - Exponential smoothing used to forecast level
   - **Holt's Model** -
     - Exponential smoothing used to forecast level and trend
   - **Holt Winters Model** (optional) –
     - Exponential smoothing used to forecast level, trend and seasonality

Simple Exponential Smoothing

Exponential smoothing is basically a **weighted average** smoothing model with a slightly different equation:

$$F_{t+1} = \alpha d_t + (1-\alpha)F_t$$

For example, let's take a model with α = 0.3. Now, say you are trying to make a forecast for the fourth timestamp. The forecast would be –

$$F_4 = (0.3)d_3 + (0.7)F_3$$

$$F_4 = (0.3)d_3 + (0.21)d_2 + (0.49)F_2$$

$$F_4 = (0.3)d_3 + (0.21)d_2 + (0.147)d_1 + (0.343)F_1$$

For any given data set, you would need to find the best value of α. You can decide what the **best value of α** will be, by using evaluation measures such as **MAPE** or **RMSE**. Usually, for large data sets, a value of **0.1** is considered ideal for α.

The **advantages** this model offers are -

1. The forecast is influenced by all past values
2. Weightage given to recent data is much more than to past data

However, using simple exponential smoothing, you cannot model data that has trend or seasonality in it. But for data that has **trend**, you can use the **Holt's model** to make forecasts.

Holt's Model

The equations for Holt's Model are -

$$S_t = \alpha d_t + (1 - \alpha) F_t$$

$$b_t = \beta (S_t - S_{t-1}) + (1 - \beta) b_{t-1}$$

$$F_{t=1} = S_t + b_t$$

Where, S_t is the **level estimate**, b_t is the **trend estimate** and F_t refers to the **overall forecast**.

Since Holt's model estimates both level and trend, it can give better estimates for data that has trend in it. For our own sales data example, you saw how Holt's model was able to capture the trend, but exponential smoothing wasn't.

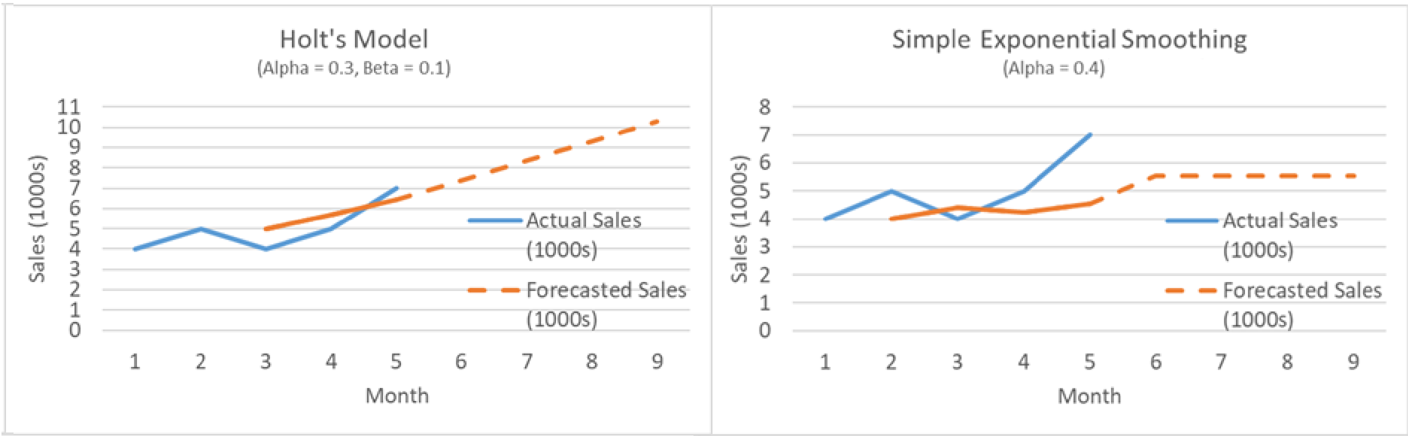

Figure 3 - Holt's Model vs Simple Exponential Smoothing

Lastly, for data that has level, trend and seasonality, you can use **Holt Winters Model**. Due to complexity and time constraints, it was not covered in the main module. However, you can learn it in the optional session of this module.