# Documentation ETS Model

## Basic Idea

ETS is an abbreviation for exponential smoothing. These models divide a time series into different components which are update or smoothed as new information becomes accessible over time. The basic decomposition of a time series in an ETS Model is into a level, trend and seasonality.

Each time period, the estimates of these components are combined to calculate the current one step ahead forecast. After the next periods true value becomes available, the forecasting error is used to update the three component estimates in order to improve the next forecast. The degree of adjustment of each component is determined by its individual smoothing component which is fixed a priori to the algorithm running threw the time series.

Note that there are multiple different ways of combining the basic components. The trend, seasonality and error term can be additive or multiplicative. Furthermore, it is not uncommon to dampen the trend as this often results in better long-term forecasts. These different combinations are all summarized under the term ETS Models and are typically distinguished by a three-letter combination. An ETS(A,N,M) Model for example, has an additive error term, no trend and a multiplicative seasonality. This illustrates how the letters denote the specification of the error, trend and seasonality in that order. The letter A stands for an additive component, Ad for additive dampened, M for multiplicative and N for an exclusion of the component.

For more information I refer you to „Forecasting with Exponential Smoothing: the State Space Approach“ (Hyndman,Koehler,Ord,Synder, 2008)[[1]](#footnote-1). This textbook provided the theoretical foundation on which this project was based.

## Initial Model exploration

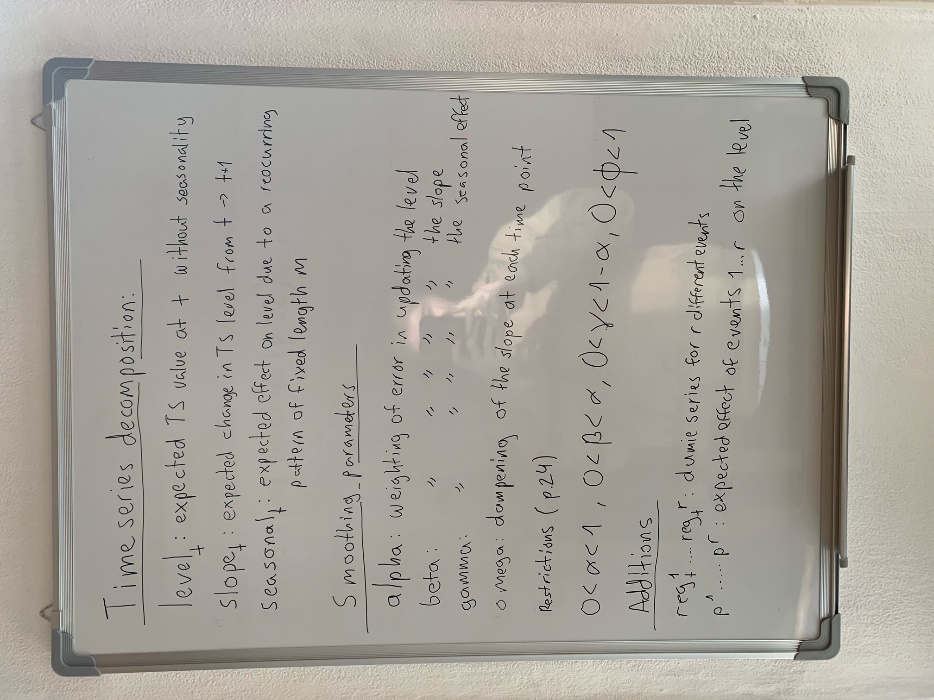
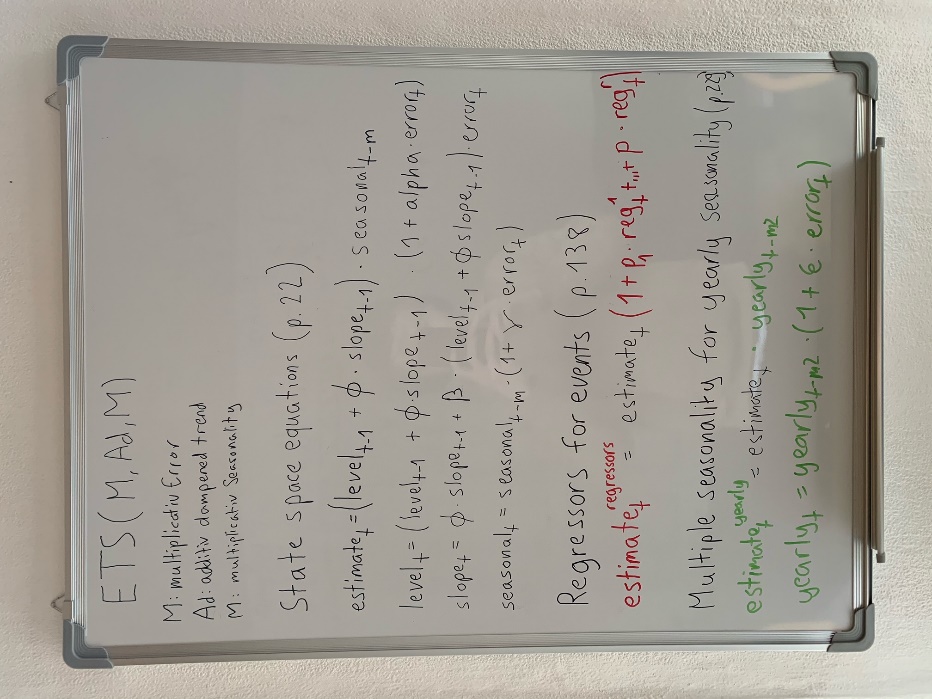
In our use-case we tried various combinations of ETS models as well as the SARIMAX and Prophet algorithms in our initial modelling stage. Note that in this stage of the project we focused exclusively on the time series of aggregated daily food sales for store number one in California.

We found that an ETS(M,Ad,M) model had the best forecast. Further it seemed the most promising from a theoretical point of view as it seems sensible that the forecasting error variance and weekly seasonality depend on the overall sales level. Hence a multiplicative error and seasonality would rather produce homoscedastic errors and a better forecast than an additive error and seasonality effect would. This intuition was supported by our initial comparison of different models.

## ETS(M,Ad,M) Model equations

After establishing why we choose to further pursue the ETS(M,Ad,M) model let us take a closer look at its equations.

The Whiteboard picture below shows the basic state space equations as well as how to add additional regressors and a second seasonality component. Note that further information can be found on pages 22,138,229 of the previously mentioned book.



### State space equations

The equations determine how the one step ahead forecasting estimate as well as the state updates are derived. Note that the smoothing parameters are time independent as they are set prior to the algorithms run though the time series. Thus, the parameters on which the algorithm is optimized are the states as well as the initial component estimators.

The large the smoothed estimates are, the more the current error is considered, thus the less strongly past observations are weighted in the forecast. In essence by setting the smoothing parameters the analyst makes a statement about how stable he believes the time series is and thus about how relevant he believes past information is for forecasting.

An important side note: The relative error is calculated as the absolute error divided by the estimate. Not to the base of the series because the relative error is used to update the estimate. I note this explicitly because this was an error I made in coding which made the model create large errors and took a long time to find.

### Regressors for events

In order to model the effect of events such as Holidays or food subsidiary days we chose a dummy variable approach. In essence each event is converted into a time series with a one on days of the event and a zero otherwise. Then we assume the effect of events to be multiplicative, thus, to depend on the general sales level. This is sensible as one can assume that sales rise higher on holidays in stores with higher average sales, thus a larger consumer base, than in a store with lower average sales. The effect on sales of our event is determined by the coefficient p. It represents the percentage change in sales due to the event. Finally note that we don’t directly multiply the effect of events but add 1 to the effect as otherwise p would have a different interpretation and our forecast would be zero for days without events.

### Multiple seasonality for yearly seasonality

Multiple seasonality is required in our use-case as we have effects of weekdays, which we model with seasonality, and effects of the time of the year, which we model with yearly. Both effects are multiplicative and are simply multiplied to the estimate.

### Initialization

Initialization is done according to p.23-24. A detailed description is provided in the comment to the initialization function in the script initialization.py.

### Structuring the model into an algorithm

To formalize the model into a script I divided the algorithm into subfunctions and combined these in a loop. The two main functions preformed in the loop in the following order are:

* calc\_error: The one-step ahead forecast is calculated with the current ETS components and compared to the true time series value to calculate the relative error.
* calc\_new\_estimates: With the new relative error and the smoothing parameters the ETS components are updated.

The loop starts with initial parameters for the ETS components as well as the smoothing parameters. It ends once the algorithm has passed threw the entire series.

Additional functions are:

* save\_estimates: it stores various important values such as the current ETS component estimates, one-step ahead forecast and error. The values are stored in list which are append each time the loop is passed.
* seasonal\_matrices: Here the transition matrices and updating vectors for the seasonal components are defined. These are required in updating the seasonal components

## Estimation of optimal parameters

### Monthly dummies and Fourier series for yearly seasonality

### Fixing Smoothing parameters

1. [Forecasting with Exponential Smoothing: the State Space Approach | Rob J Hyndman](https://robjhyndman.com/expsmooth/) [↑](#footnote-ref-1)