Facebook Prophet

Parameter name	Options	Description
changepoint_prior_scale	default: 0.05 float	if it is too small, the trend will be underfit and variance that should have been modeled with trend changes will instead end up being handled with the noise term. If it is too large, the trend will overfit and in the most extreme case you can end up with the trend capturing yearly seasonality. The default of 0.05 works for many time series, but this could be tuned
seasonality_prior_scale	default : 10.0 float	This parameter controls the flexibility of the seasonality. Similarly, a large value allows the seasonality to fit large fluctuations, a small value shrinks the magnitude of the seasonality. The default is 10., which applies basically no regularization. That is because we very rarely see overfitting here. A reasonable range for tuning it would probably be [0.01, 10]
holidays_prior_scale	default : 10.0 float	This controls flexibility to fit holiday effects. Similar to seasonality_prior_scale, it defaults to 10.0 which applies basically no regularization, since we usually have multiple observations of holidays and can do a good job of estimating their effects. This could also be tuned on a range of [0.01, 10] as with seasonality_prior_scale
seasonality_mode	default: 'additive' options: ['additive', 'multiplicative']	Default is 'additive', but many business time series will have multiplicative seasonality. This is best identified just from looking at the time series and seeing if the magnitude of seasonal fluctuations grows with the magnitude of the time series (see the documentation here on multiplicative seasonality), but when that isn't possible, it could be tuned.
yearly_seasonality	Default : 'auto' Options : ['auto', True, False]	By default ('auto') this will turn yearly seasonality on if there is a year of data, and off otherwise. Options are ['auto', True, False]. If there is more than a year of data, rather than trying to turn this off during HPO, it will likely be more effective to leave it on and turn

		down seasonal effects by tuning seasonality_prior_scale.
weekly_seasonality daily_seasonality	Same as yearly_seasonality	Same as yearly seasonality

Neural Prophet

Parameter name	Options	Description
n_forecasts	default: 1 int	The size of the forecast horizon. The default value of 1 means that the model forecasts one step into the future.
n_lags	default : 0 int	defines whether the AR-Net is enabled (if n_lags > 0) or not. The value for n_lags is usually recommended to be greater than n_forecasts, if possible since it is preferable for the FFNNs to encounter at least n_forecasts length of the past in order to predict n_forecasts into the future. Thus, n_lags determine how far into the past the auto-regressive dependencies should be considered. This could be a value chosen based on either domain expertise or an empirical analysis
learning_rate	default : None float	NeuralProphet is fit with stochastic gradient descent - more precisely, with an AdamW optimizer and a One-Cycle policy. If the parameter learning_rate is not specified, a learning rate range test is conducted to determine the optimal learning rate.
loss_func	default: 'Huber' options: any other PyTorch torch.nn.modules.loss loss function	Loss function

num_hidden_layers	default : 0 int	defines the number of hidden layers of the FFNNs used in the overall model. This includes the AR-Net and the FFNN of the lagged regressors. The default is 0, meaning that the FFNNs will have only one final layer of size n_forecasts. Adding more layers results in increased complexity and also increased computational time, consequently. However, the added number of hidden layers can help build more complex relationships especially useful for the lagged regressors.
normalize_y	default:'auto' users can turn this off or select another normalization	About scaling the time series before modelling. By default, NeuralProphet performs a (soft) min-max normalization of the time series. Normalization can help the model training process if the series values fluctuate heavily. However, if the series does not such scaling, users can turn this off or select another normalization.
impute_missing		İmputing the missing values in a given series. Similar to Prophet, NeuralProphet too can work with missing values when it is in the regression mode without the AR-Net. However, when the autocorrelation needs to be captured, it is necessary for the missing values to be imputed, since then the modelling becomes an ordered problem. Letting this parameter at its default can get the job done perfectly in most cases.
yearly_seasonality weekly_seasonality daily_seasonality		Seasonal components to be modelled. For example, if you use temperature data, you can probably select daily and yearly. Using number of passengers using the subway would more likely have a weekly seasonality for example. Setting these seasonalities at the default auto mode, lets NeuralProphet decide which of them to include depending on how much data available. For example, the yearly seasonality will not be considered if less than two years data available. In the same manner, the weekly seasonality will not be considered if less than two weeks available etc... However, if the user if certain that the series does not include yearly, weekly or daily seasonality, and thus the model should not be distorted by such components, they can explicitly turn them off by setting the respective components to False. Apart from that, the parameters yearly_seasonality,

		weekly_seasonality and daily_seasonality can also be set to number of Fourier terms of the respective seasonalities. The defaults are 6 for yearly, 4 for weekly and 6 for daily. Users can set this to any number they want.