TADA

PROPHET

Forecasting at Scale

Hannah Do | 10.29.2022

Contents

1. Introduction

PROPHET

PROPHET : Forecasting at Scale

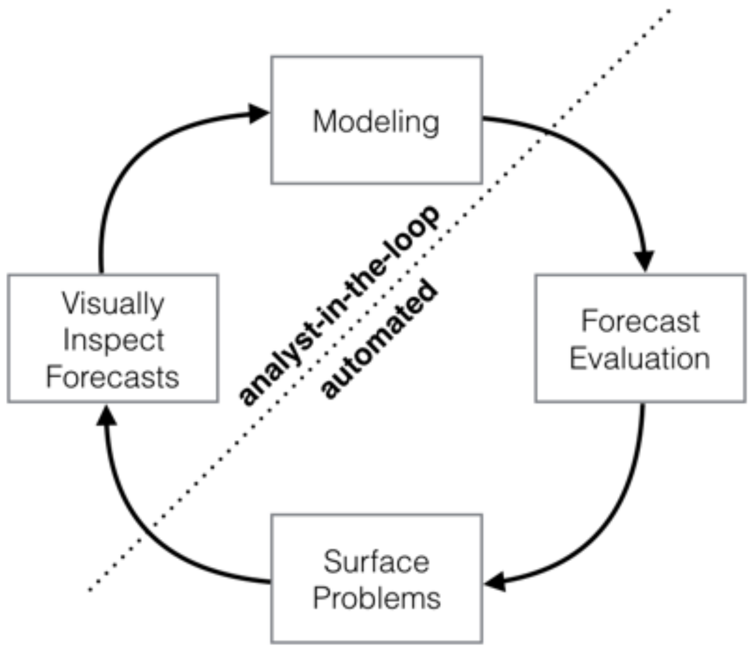
- Introduced by Facebook (Taylor & Letham, 2018)
- Used for forecasting univariate time series by decomposing the time series into pieces
- Made easy to use, especially for domain experts.

1. Introduction

Business forecasting methods should be suitable for

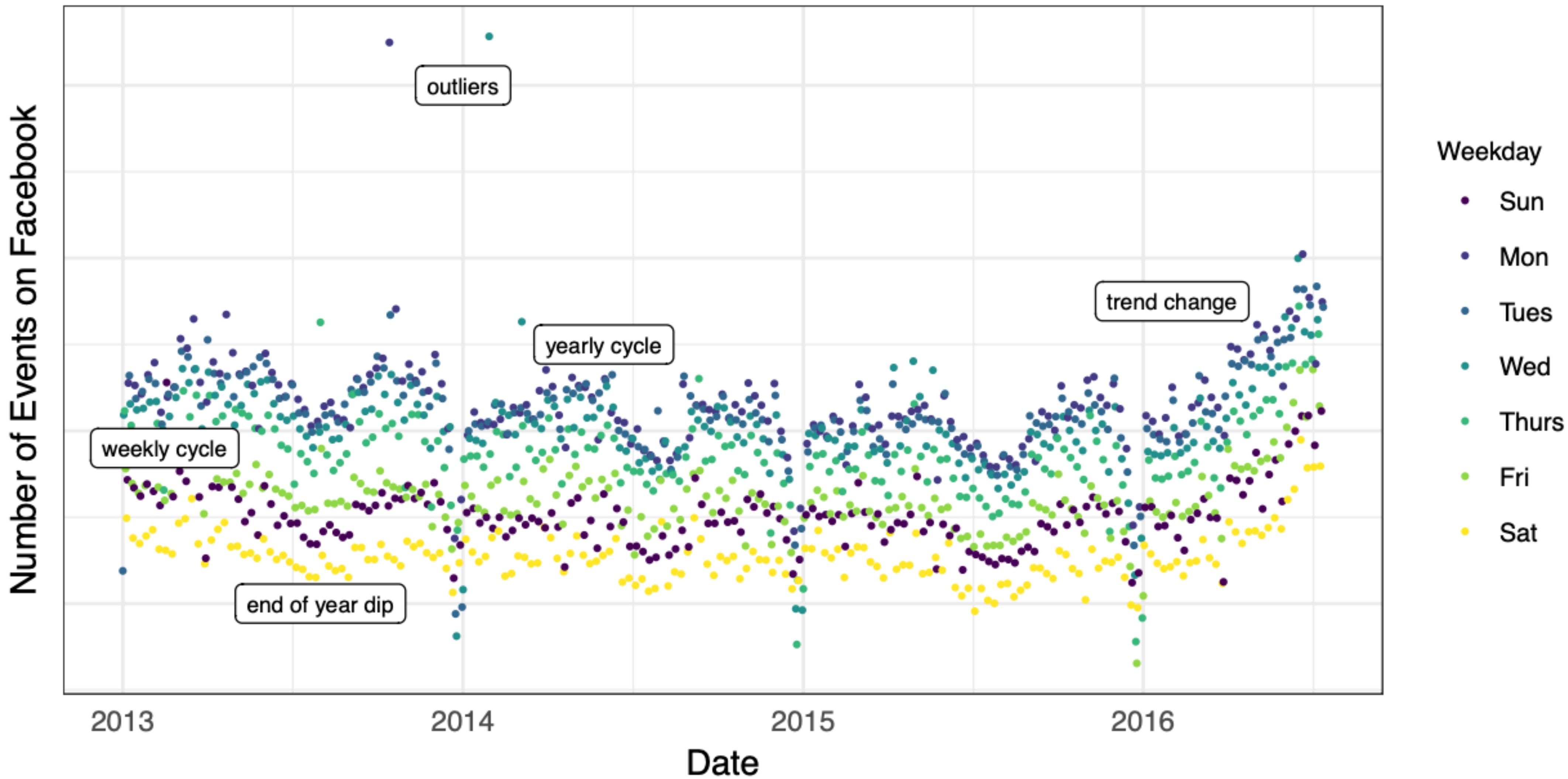
1. a large number of people making forecasts, possibly without training in time series methods
2. a large variety of forecasting problems with potentially idiosyncratic features

→ A large number of forecasts should be created, necessitating efficient, automated means of evaluating and comparing the forecasts



Prophet : Forecasting at ***Scale***

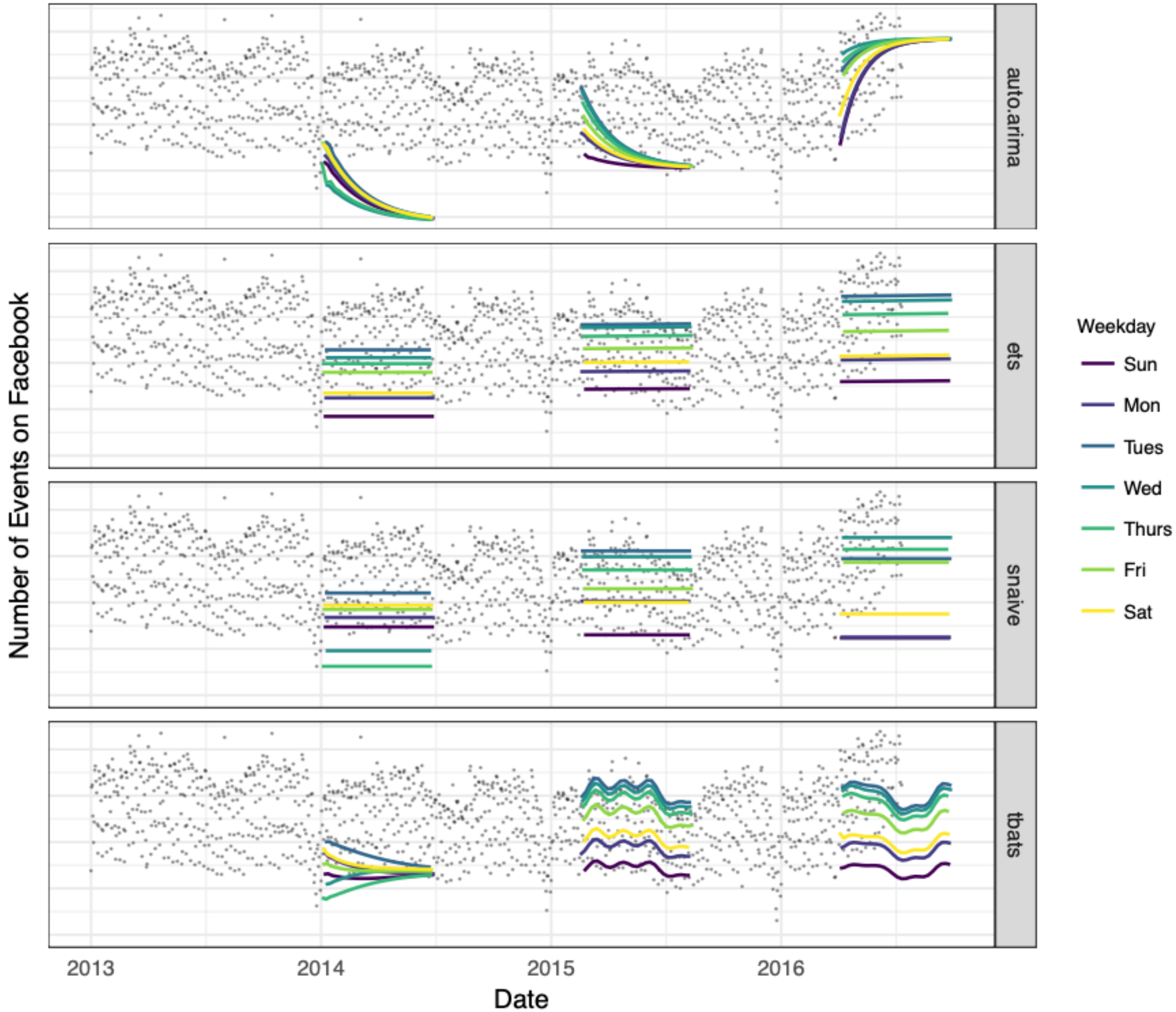
1. Introduction



The number of events created on Facebook.
Points for each day, color-coded by day-of-week to show weekly cycle.

Several seasonal effects are visible in this time series : weekly and yearly cycles.
These types of effects naturally arise and can be expected in time series generated by human actions.

1. Introduction



Forecasts on the time series from the previous figure using automated procedures from the forecast package in R

- **auto.arima** : fits a range of ARIMA models and automatically selects the best one
- **ets** : fits a range of exponential smoothing models and selects the best
- **snaive** : random walk model that makes constant predictions with weekly seasonality (seasonal naive)
- **tbats** : TBATS model with both weekly and yearly seasonalities

**Forecasts were made at three illustrative points in the history, each using only the portion of the time series up to that point. Outliers were removed for visualization.*

2. Abstract

Prophet is based on a **decomposable time series model** (Harvey & Peters, 1990) with three main model components : trend, seasonality, and holidays.

$$y(t) = g(t) + s(t) + h(t) + \epsilon_t.$$

g(t) : Trend function that models non-periodic changes in the value of time series

s(t) : Represents periodic changes (weekly and yearly seasonality)

h(t) : Effects of holidays which occur on potentially irregular schedules over one or more days

ϵ_t : Error term that represents any idiosyncratic changes which are not accommodated by the model, assumption is made that ε_t is normally distributed.

This results in creating a model that is similar to a generalized additive model (GAM), a class of regression models with potentially non-linear smoothers applied to the regressors.

Using GAM formulation allows easier decomposition and accommodating new components - a new source of seasonality is identified. Prophet is actually closer to a **curve-fitting exercise**, inherently different from time series models that explicitly account for the temporal dependence structure in data.

2. Abstract

Prophet is based on a **decomposable time series model** (Harvey & Peters, 1990) with three main model components : trend, seasonality, and holidays.

$$y(t) = g(t) + s(t) + h(t) + \epsilon_t.$$

g(t) : L1-regularized trend shifts

s(t) : Fourier series

h(t) : Dummy variables

ϵ_t : noise

This results in creating a model that is similar to a generalized additive model (GAM), a class of regression models with potentially non-linear smoothers applied to the regressors.

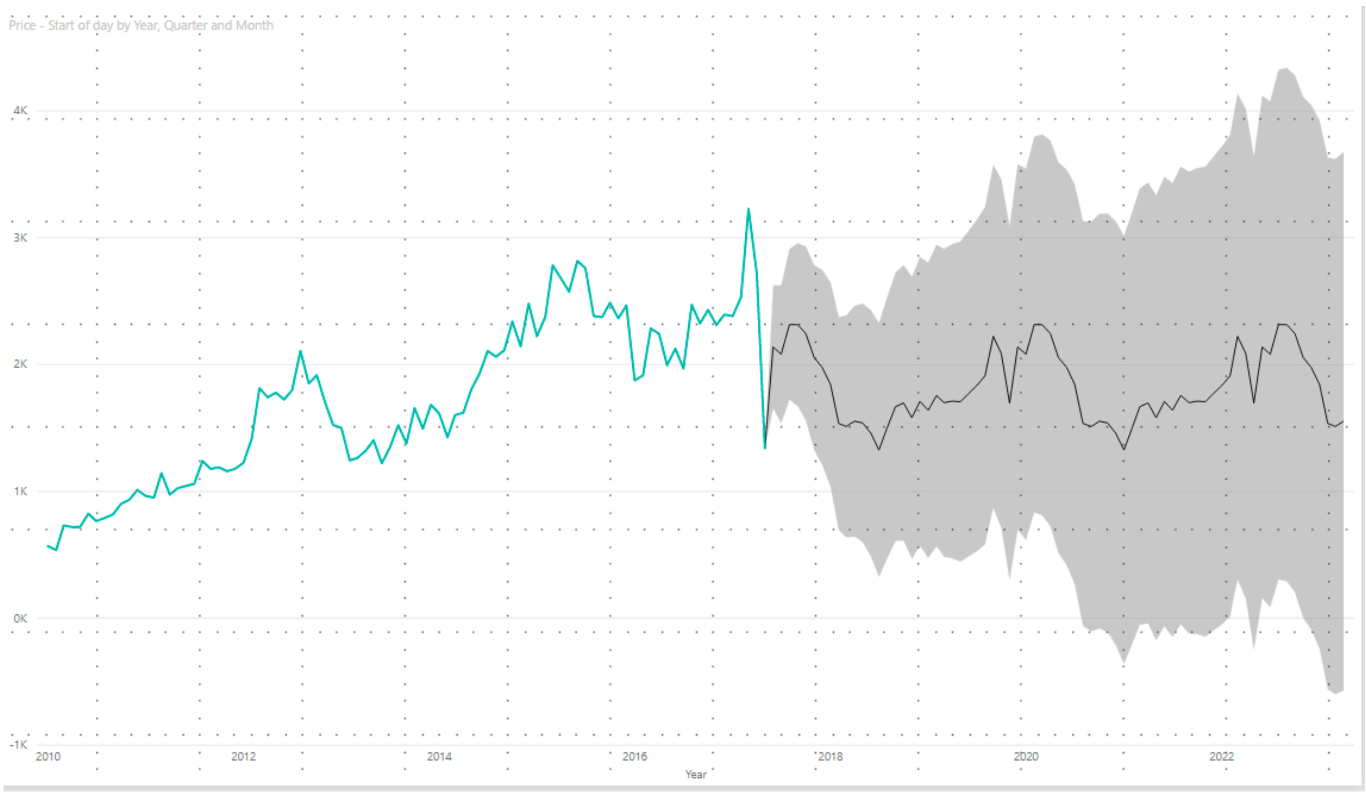
Using GAM formulation allows easier decomposition and accommodating new components - a new source of seasonality is identified. Prophet is actually closer to a **curve-fitting exercise**, inherently different from time series models that explicitly account for the temporal dependence structure in data.

3. g(t) : Trend

<u>Trend models in Prophet</u>

Trend line (time) is used as regressors in the model, there are two trend models implemented for Prophet :

A. Saturating Growth model : Uses logarithmic trend, similar to dampened trend approach in exponential smoothing
B. Piecewise Linear model : Regression is broken into different pieces of data using knots - user can specify knots / automatically chosen



https://www.analyticsvidhya.com/blog/2020/07/time-series-forecasting-using-microsoft-power-bi/

3. g(t) : Trend

A. Saturating growth model

Modeling growth at Facebook is similar to population growth in natural ecosystems - nonlinear growth that saturates at a carrying capacity.
This sort of growth was modeled by a following logistic growth model :

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

C : Carrying capacity - is actually not constant (proportional to the # of people who have access to Internet increases)

- replaced with time-varying capacity C(t)

k : Growth rate - needs to be incorporated with varying rate to fit historical data

- Assuming S changepoints at times s_j, rate at any time t becomes the base rate k, plus all of the adjustments up to that point (δ_j as the change in rate that occurs at time s_j) : $k + \sum_{j:t>s_j} \delta_j$

m : Offset parameter - adjusted to connect endpoints of the segments

$$g(t) = \frac{C(t)}{1 + \exp(-(k + \mathbf{a}(t)^\intercal \boldsymbol{\delta})(t - (m + \mathbf{a}(t)^\intercal \boldsymbol{\gamma})))}.$$

3. g(t) : Trend

B. Piecewise Linear model

For forecasting problems that do not exhibit saturating growth.
k : growth rate
δ : rate adjustments
m : offset parameter
γ_j : $-s_j\delta_j$ (to make the function continuous)

$$g(t) = (k + \mathbf{a}(t)^\intercal \boldsymbol{\delta})t + (m + \mathbf{a}(t)^\intercal \boldsymbol{\gamma}),$$

C. Automatic Changepoint Selection

Changepoints or knots can be specified by the analyst or automatically selected.
If automatically selected, a sparse prior on δ is set to ~ Laplace(0, τ). The parameter τ directly controlling the flexibility of the model.

4. s(t) : Seasonality

Seasonality

Uses Fourier transform :

Fourier shows that a series of sine and cosine terms of the right frequencies approximate periodic series. The different frequencies are added up to account for the seasonal pattens - weekly, yearly.

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

P : regular period
(P=365 for yearly data, P=7 for weekly data)

N : required to construct a matrix of seasonality vectors :
$\beta = [a_1, b_1, \ldots, a_N, b_N]^T$

N=10 and N=3 seemed to work well for yearly and weekly seasonality (increasing N allows for fitting seasonal patterns that change more quickly → increased risk of overfitting)

ex) N = 10

$$X(t) = \left[\cos\left(\frac{2\pi(1)t}{365.25}\right), \ldots, \sin\left(\frac{2\pi(10)t}{365.25}\right)\right].$$

$$s(t) = X(t)\boldsymbol{\beta}.$$

$$\boldsymbol{\beta} \sim \text{Normal}(0, \sigma^2)$$

5. h(t) : Holidays

Holidays

Uses Point (Pulse) Intervention as a binary indicator for a certain day

Holidays such as Thanksgiving, Super Bowl, or Christmas can be taken into account

Holiday	Country	Year	Date
Thanksgiving	US	2015	26 Nov 2015
Thanksgiving	US	2016	24 Nov 2016
Thanksgiving	US	2017	23 Nov 2017
Thanksgiving	US	2018	22 Nov 2018
Christmas	*	2015	25 Dec 2015
Christmas	*	2016	25 Dec 2016
Christmas	*	2017	25 Dec 2017
Christmas	*	2018	25 Dec 2018

$$Z(t) = [\mathbf{1}(t \in D_1), \ldots, \mathbf{1}(t \in D_L)]$$

An indicator is added whether time t is during holiday i, each holiday is assigned a parameter κ_i as the corresponding change in the forecast.

$$h(t) = Z(t)\boldsymbol{\kappa}. \qquad \boldsymbol{\kappa} \sim \mathrm{Normal}(0, \nu^2).$$

6. Model Fitting

Stan’s L-BFGS

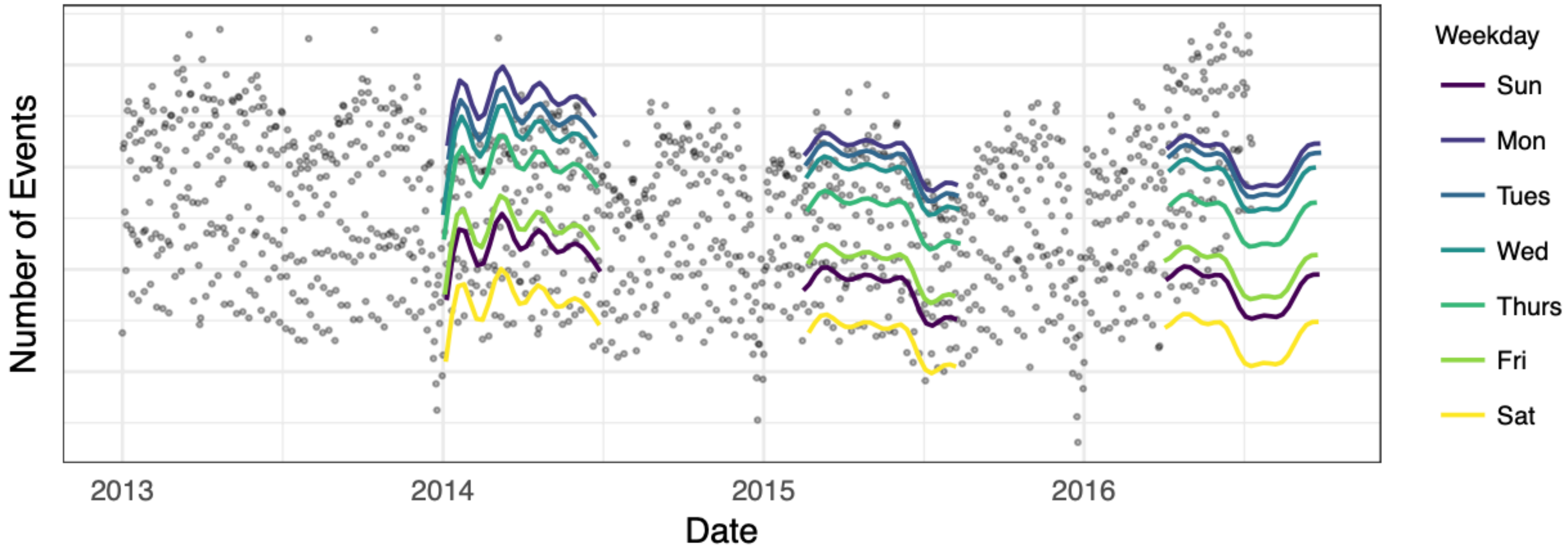
Assuming seasonality and holiday features are combined into a matrix X and the change point indicators a(t) in matrix A, the entire model can be expressed as the following :

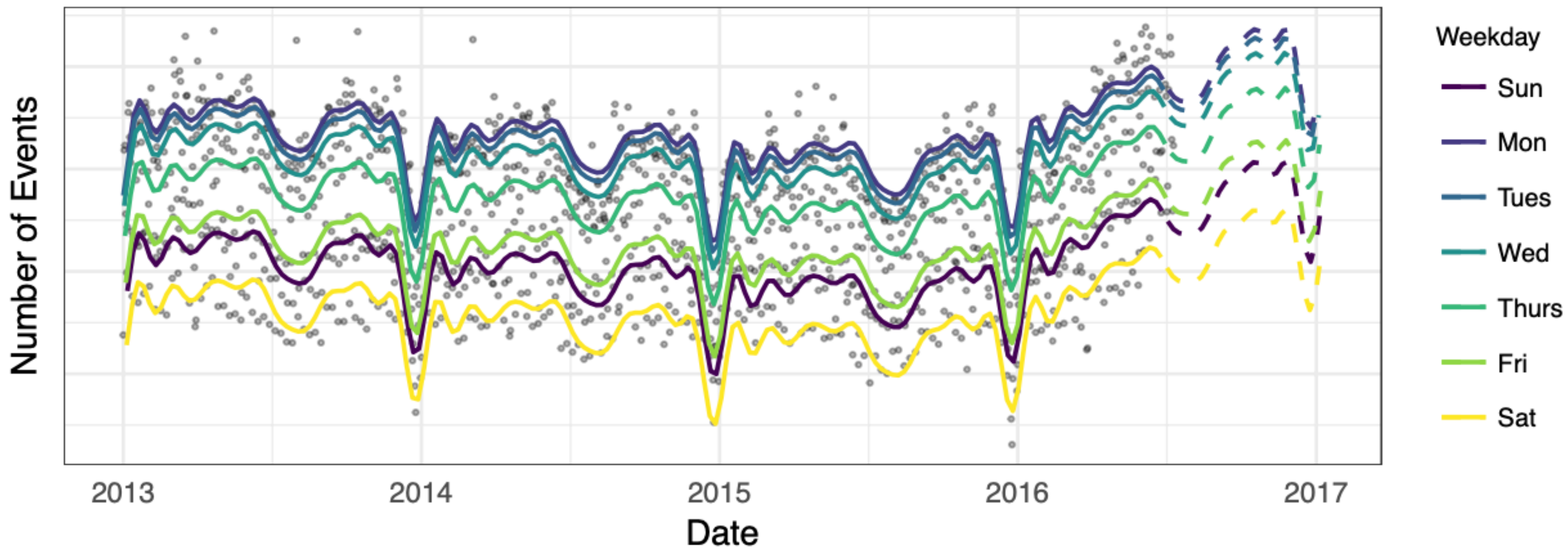
```
model {
  // Priors
  k ~ normal(0, 5);
  m ~ normal(0, 5);
  epsilon ~ normal(0, 0.5);
  delta ~ double_exponential(0, tau);
  beta ~ normal(0, sigma);

  // Logistic likelihood
  y ~ normal(C ./ (1 + exp(-(k + A * delta) .* (t - (m + A * gamma)))) +
             X * beta, epsilon);
  // Linear likelihood
  y ~ normal((k + A * delta) .* t + (m + A * gamma) + X * beta, sigma);
}
```

For model fitting, Stan’s L-BFGS is used to find the maximum a posteriori estimate, but a full posterior inference can be used to include model parameter uncertainty in the forecast uncertainty.

7. Results

Prophet Forecast

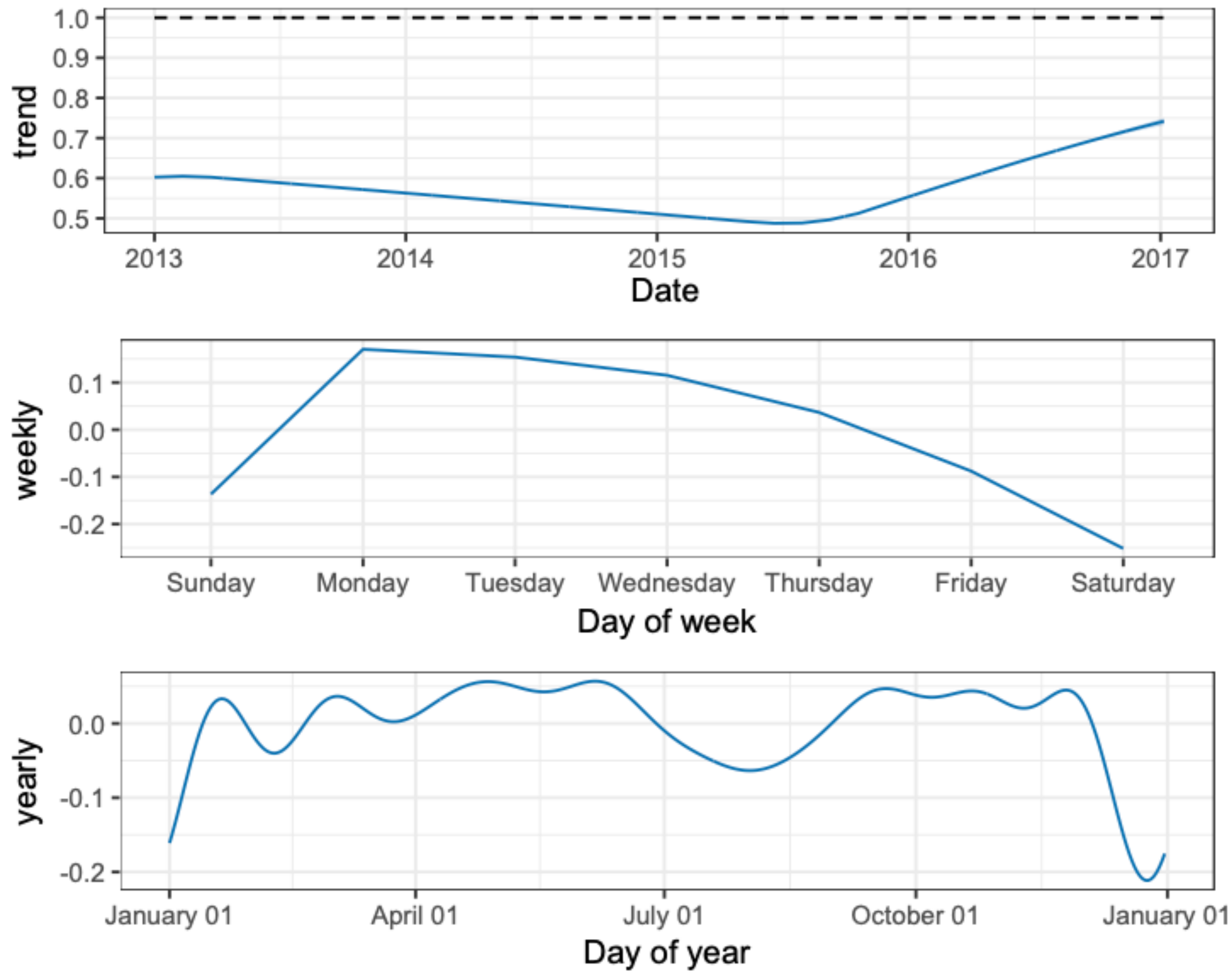




(Top) Prophet forecasts corresponding to the other time-series forecasting models from slide 6.
(Bottom) Prophet forecast using all available data, including the interpolation of historical data.

7. Results

Prophet Forecast

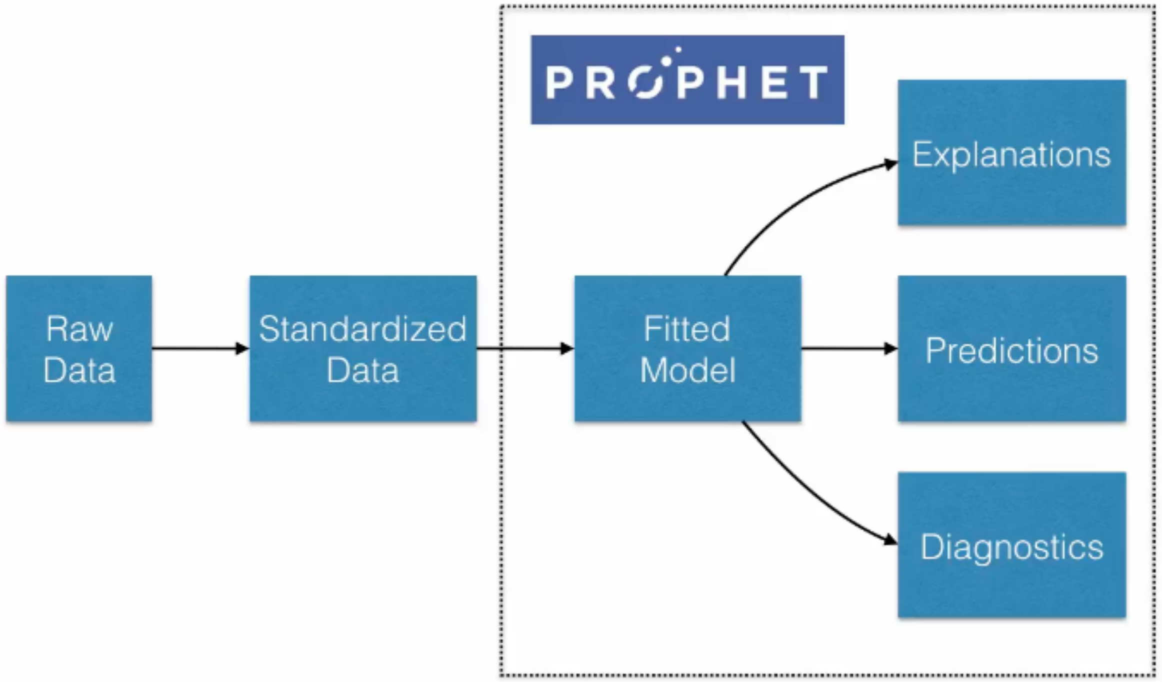


Decomposed linear graph for each variable :

Piece wise linear trend can be observed - local trend change causing extrapolation

Seasonal characteristics shown in yearly forecast graph

8. Application



Parameters in Prophet

- Priors on seasonality parameters
- Prior on how often we expect changepoints
- Functional form for growth (piecewise linear and logistic)
- Covariates and holidays
- Custom seasonalities
- MAP or full posterior

8. Application

Fairly easy to use

1. Install pystan & fbprophet

```bash
# bash
# Install pystan with pip before using pip to install fbprophet
$ pip install pystan
$
$ pip install fbprophet
```

2. Load Prophet model, set parameters, fit (train) & predict (make_future_dataframe, predict)

```python
m = Prophet(interval_width=0.95, daily_seasonality=True)
model = m.fit(df)
```

```python
future = m.make_future_dataframe(periods=100, freq='D')
forecast = m.predict(future)
```

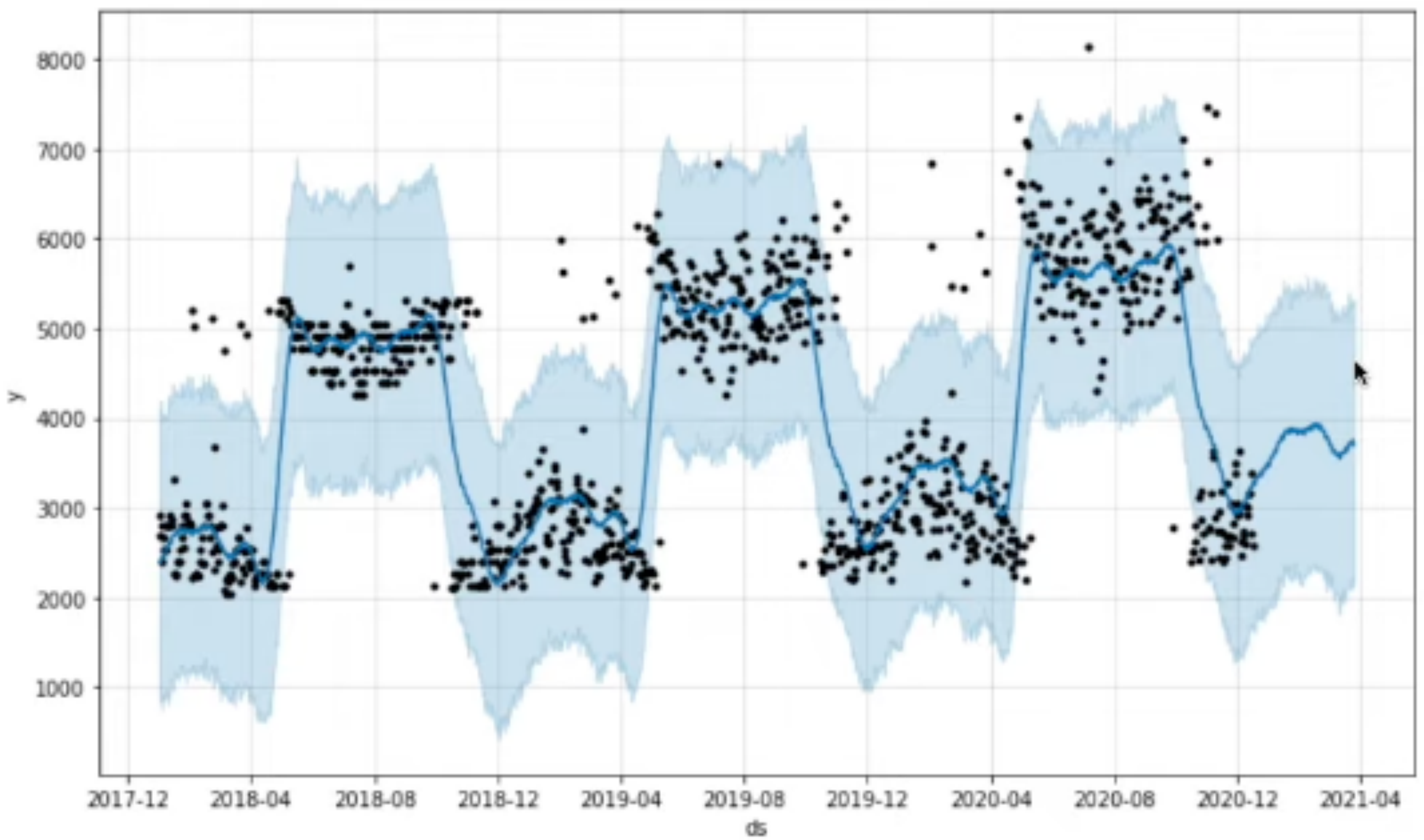
datetime datatype as* ***ds**, prediction variable as* ***y***
periods *: # of forecasts in addition to historical data*

8. Application

3. plot() function for visualizations

- black dots : actual values
- blue lines : predictions
- shaded area : different bounding boxes

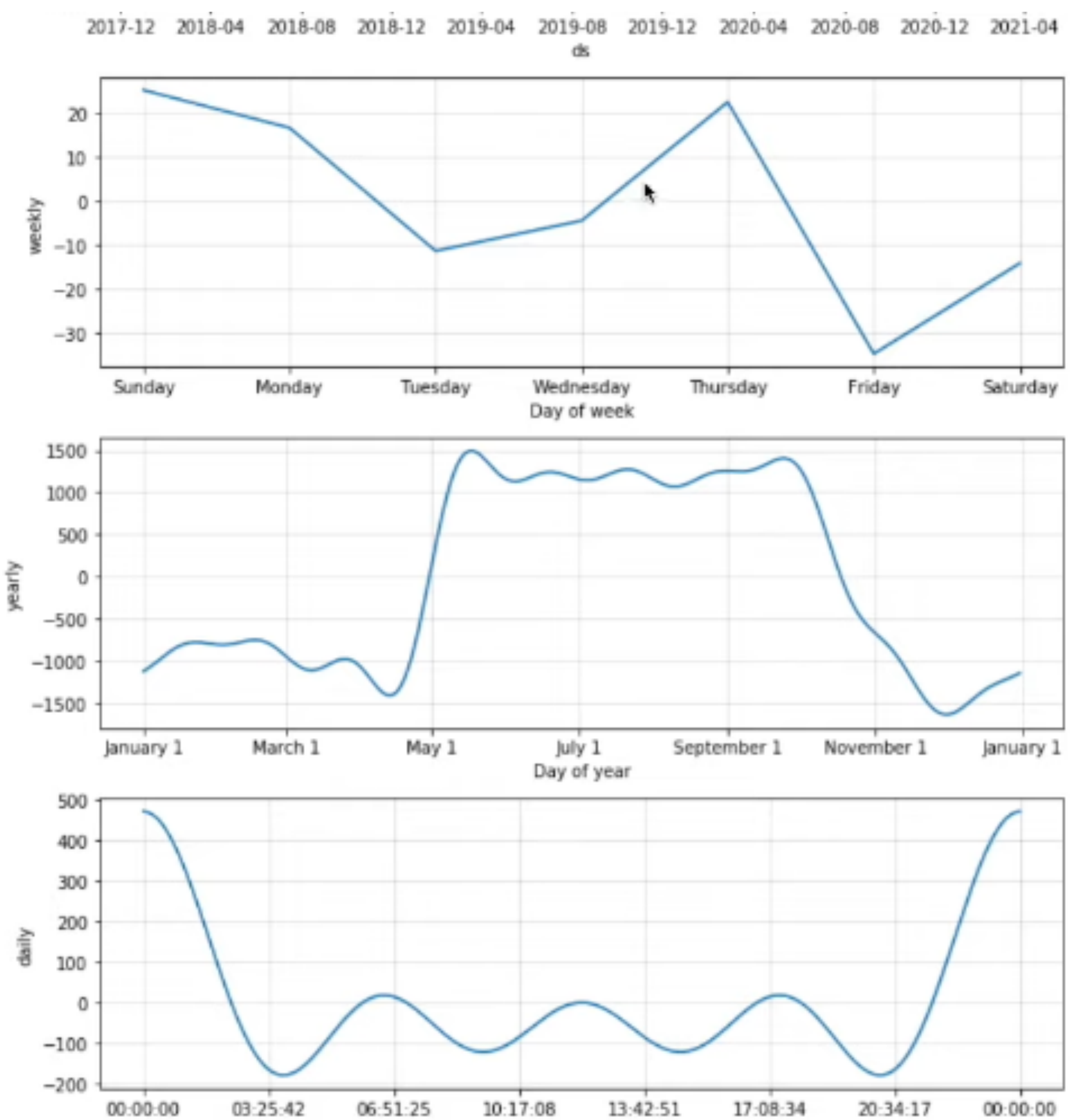
```
plot1 = m.plot(forecast)
```



8. Application

4. plot_components() function for visualizations - seasonal trends (yearly, weekly, daily, …) available for view

```
plot2 = m.plot_components(forecast)
```



9. Conclusion

"A major theme of forecasting at scale is that analysts with a variety of backgrounds must make more forecasts than they can do manually. Prophet uses 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.

...

Simple, adjustable models and scalable performance monitoring in combination allow a large number of analysts to forecast a large number and a variety of time series – **what we consider forecasting at scale.** "

10. References

1. Taylor, S.J. & Letham, B. (2017) Forecasting at Scale
    - https://peerj.com/preprints/3190.pdf

2. LaBarr, Aric (Feb 17, 2022) What is the Prophet Model, Time Series
    - https://www.youtube.com/watch?v=2XFro0nIHQM&ab_channel=AricLaBarr

3. Lander Analytics (Aug 2, 2018) Forecasting at Scale: How and Why We Developed Prophet for Forecasting at Facebook
    - https://www.youtube.com/watch?v=pOYAXv15r3A&ab_channel=LanderAnalytics

4. Renotte, Nicholas (Dec 17, 2020) Time Series Forecasting with Facebook Prophet and Python in 20 Minutes
    - https://www.youtube.com/watch?v=KvLG1uTC-KU&ab_channel=NicholasRenotte