# **Facebook’s Prophet**

* One of the drawbacks of the SARIMAX Model is that one cannot have multiple seasonality and can only select one value.
* To overcome this disability, we're studying an open-source tool/library called prophet which is developed by Facebook

**The following are the features of Facebook's Prophet:**

* Provides intuitive parameters which can be **easily tuned**
* It is **robust to missing data and shifts in the trend**, and typically **handles outliers** well.
* It can account for **multiple seasonalities**. This is possible because, under the hood, the math of seasonalities is based on **Fourier transforms**, which help incorporate this.
* The Prophet uses a decomposable time series model with three main model components
* They are combined in the following equation:

**y(t)= g(t) + s(t) + h(t) + εt**

* **g(t):** piecewise linear or logistic growth curve for modeling non-periodic changes in time series (**trend**)
* **s(t):** periodic changes (e.g. weekly/yearly **seasonality**)
* **h(t):** effects of **holidays** (user provided) with irregular schedules
* **εt:** **error term** accounts for any unusual changes not accommodated by the model.
* For using Prophet, the data set should contain only two columns with column names as ‘ds’ and ‘y’
* ‘ds’ should always be in ‘date-time’ format. And ‘y’ represents a feature that you want to forecast.
* To install prophet to your environment, you can use the following environments:

!pip install pystan~=2.14

!pip install fbprophet

→ from fbprophet import Prophet

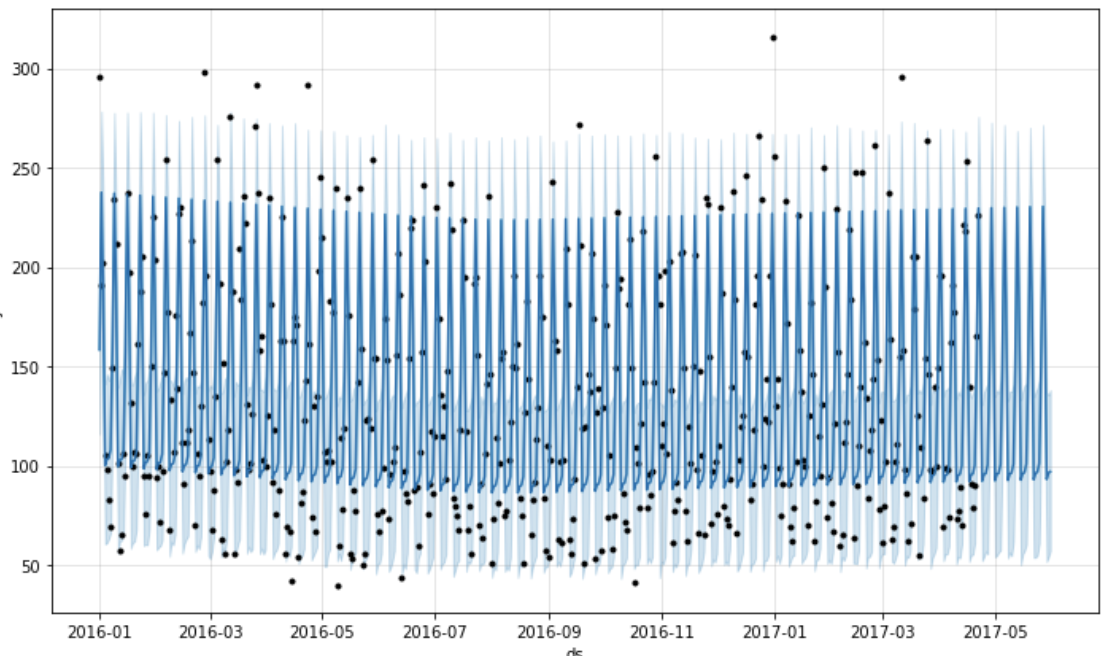
m = Prophet()

m.fit(df[['ds', 'y']][:-39]) #here we are leaving last 39 observations because we will predict it in 'future'

future = m.make\_future\_dataframe(periods=39,freq="D")

forecast = m.predict(future)

fig = m.plot(forecast)



* In the plot, black dots are actual visits, deep blue lines are the predicted visits and light blue lines is the 95% confidence interval around the prediction.
* You can see that the lines are flat and the model is not able to capture the seasonality properly so it is not a good fit.
* Here light blue lines are 95% confidence intervals around the predictions.
* Here we also didn't do anything explicitly for NaNvalues it was handled by the prophet.
* We can use ‘**add\_regressor’** for adding features to the prophet model.
  + There is another interesting parameter here: ‘**changepoint\_prior\_scale’**
* If the trend changes are being overfit (too much flexibility) or underfit (very less flexibility)
* We can adjust the strength of sparse prior using this parameter.
* Default value: 0.05
* Increasing this will make the trend more flexible.

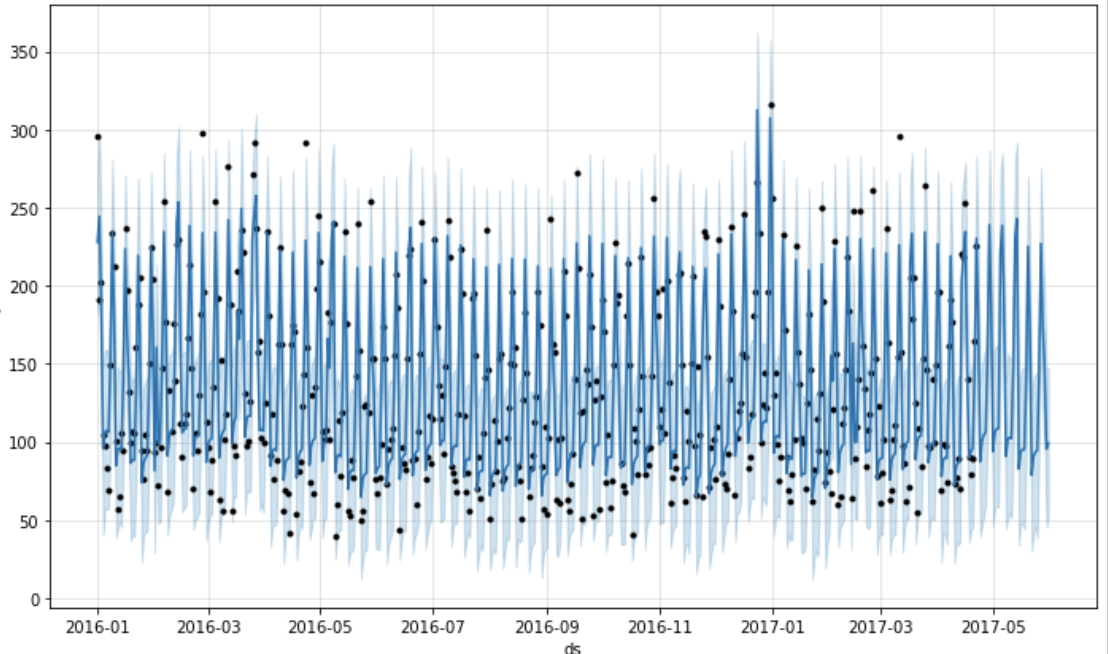
→ model2=Prophet(interval\_width=0.95, yearly\_seasonality=True, weekly\_seasonality=True,changepoint\_prior\_scale=4)

model2.add\_regressor('holiday') #adding holidays data in the model3

model2.fit(df[:-39])

forecast2 = model2.predict(df)

fig = model2.plot(forecast2)



### 

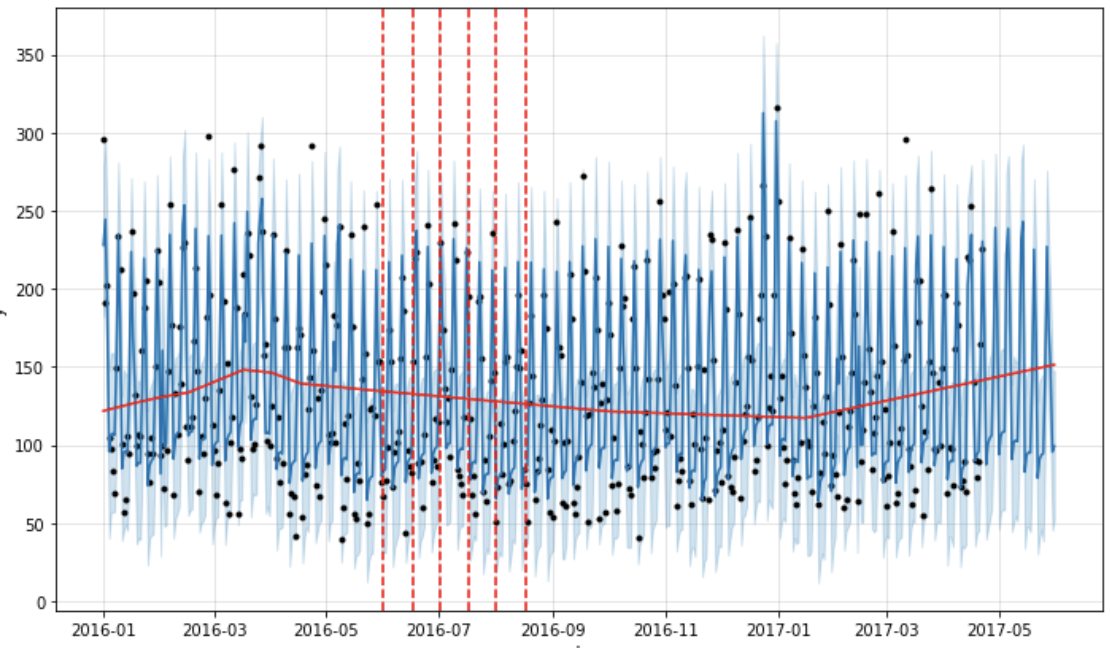
### **Changing Trends**

* Prophet automatically detects the changepoints and will allow the trend to adapt appropriately.
* Prophet detects changepoints by first specifying a large number of potential changepoints at which the rate is allowed to change.

→ from fbprophet.plot import add\_changepoints\_to\_plot

fig = m.plot(forecast2)

a = add\_changepoints\_to\_plot(fig.gca(), m, forecast2)



* The vertical lines in this figure indicate where the potential change-points were placed.
* Facebook prophet provides automated methods to forecast.
* The model has easily interpretable parameters that can be changed by the analyst to impose assumptions on the forecast.

### **Benefits of Prophet**

* The prophet is a simple library and is great for beginners.
* It works best with time series that have strong seasonal effects.
* Prophet is robust to missing data and shifts in the trend, and typically handles outliers well.
* We can add multiple regressors or exogenous variables.
* It forms a very good baseline model because almost no feature engineering is required.
* Interpretability is one of the key advantages of the Prophet.
* If your time series follows some business cycles, you can obtain very decent performance quickly.
* It can also be helpful while detecting change points.