

```
In [1]: #importing important libraries
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
#importing the required libraries for forecasting using the facebook prophet algorithm
from fbprophet import Prophet
from fbprophet.plot import add_changepoints_to_plot
from fbprophet.diagnostics import cross_validation
from fbprophet.diagnostics import performance_metrics
from fbprophet.plot import plot_cross_validation_metric
```

```
In [11]: #importing the dataset of stocks of ebay in the city of new york
data = pd.read_csv('ebay_data.csv')
```

```
In [12]: #getting an idea of the datapoints of ebay stocks
data.head(5)
```

Out[12]:

	date	symbol	open	close	low	high	volume	close - open	eps ratio	pe r
0	24-11-2015	EBAY	28.420000	29.000000	28.420000	29.180000	7769400	0.580000	7.470000e-08	3908811
1	02-11-2015	EBAY	27.730000	28.500000	27.719999	28.520000	11305900	0.770000	6.810000e-08	4187587
2	04-12-2015	EBAY	28.719999	29.350000	28.590000	29.590000	10055900	0.630001	6.260000e-08	4723073
3	16-09-2015	EBAY	26.000000	26.740000	25.910000	26.750000	13030700	0.740000	5.680000e-08	4710421
4	12-10-2015	EBAY	24.040001	24.559999	23.980000	24.620001	9882300	0.519998	5.260000e-08	4678907

```
In [13]: #preparing the dataset to be put through the algorithm
#it is necessary that datetime be named ds and the value to be predicted named y
df = pd.DataFrame()
df['ds'] = pd.to_datetime(data['date'])
df['y'] = data['close']
```

```
In [14]: #feeding the dataset created to the prophet time series
m = Prophet()
m.fit(df)
```

```
INFO:numexpr.utils:NumExpr defaulting to 4 threads.
INFO:fbprophet:Disabling daily seasonality. Run prophet with daily_seasonality=True to override this.
```

Out[14]: <fbprophet.forecaster.Prophet at 0xlead5daa308>

```
In [15]: #making few prediction to see if everything is working, printing out the predicted
datapoints from the end of the dataset
future = m.make_future_dataframe(periods=12 * 6, freq='M')
future.tail()
```

Out[15]:

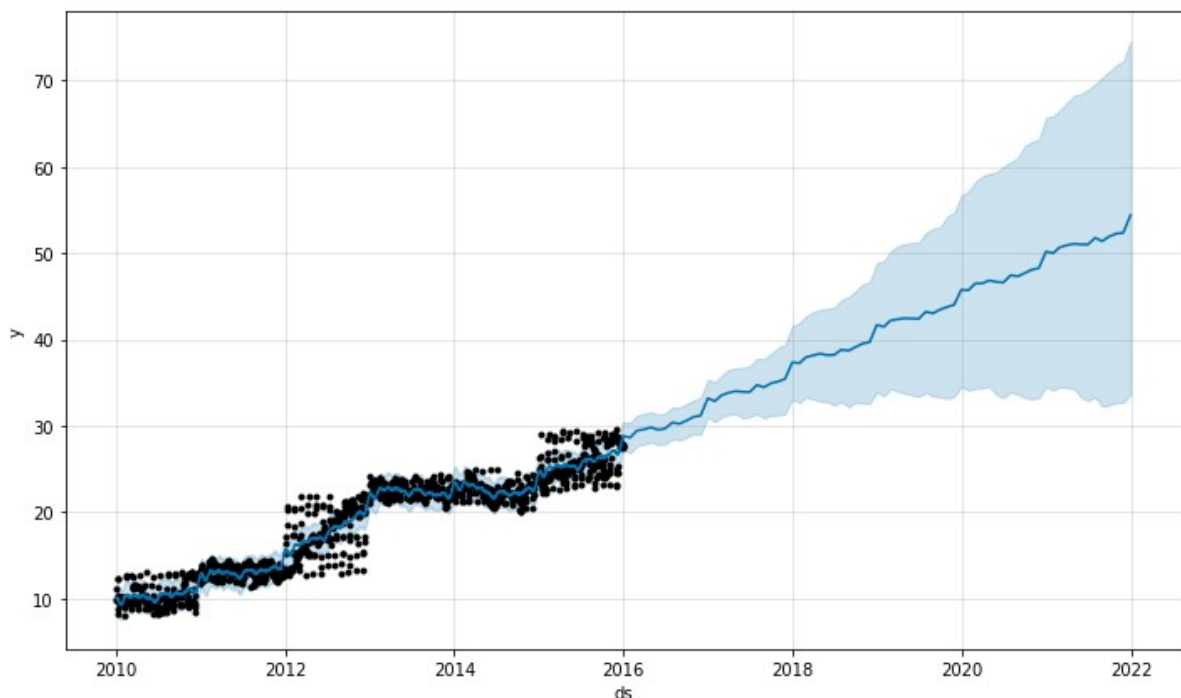
	ds
1567	2021-08-31
1568	2021-09-30
1569	2021-10-31
1570	2021-11-30
1571	2021-12-31

```
In [16]: #setting out the parameters to predict and forecasting
#yhat column is the predicted value
#yhat_lower and yhat_upper columns shows the uncertainty interval
#There are three sources of uncertainty in the forecast: uncertainty in the trend,
uncertainty in the seasonality estimates, and additional observation noise.
forecast = m.predict(future)
forecast[['ds', 'yhat', 'yhat_lower', 'yhat_upper']].tail()
```

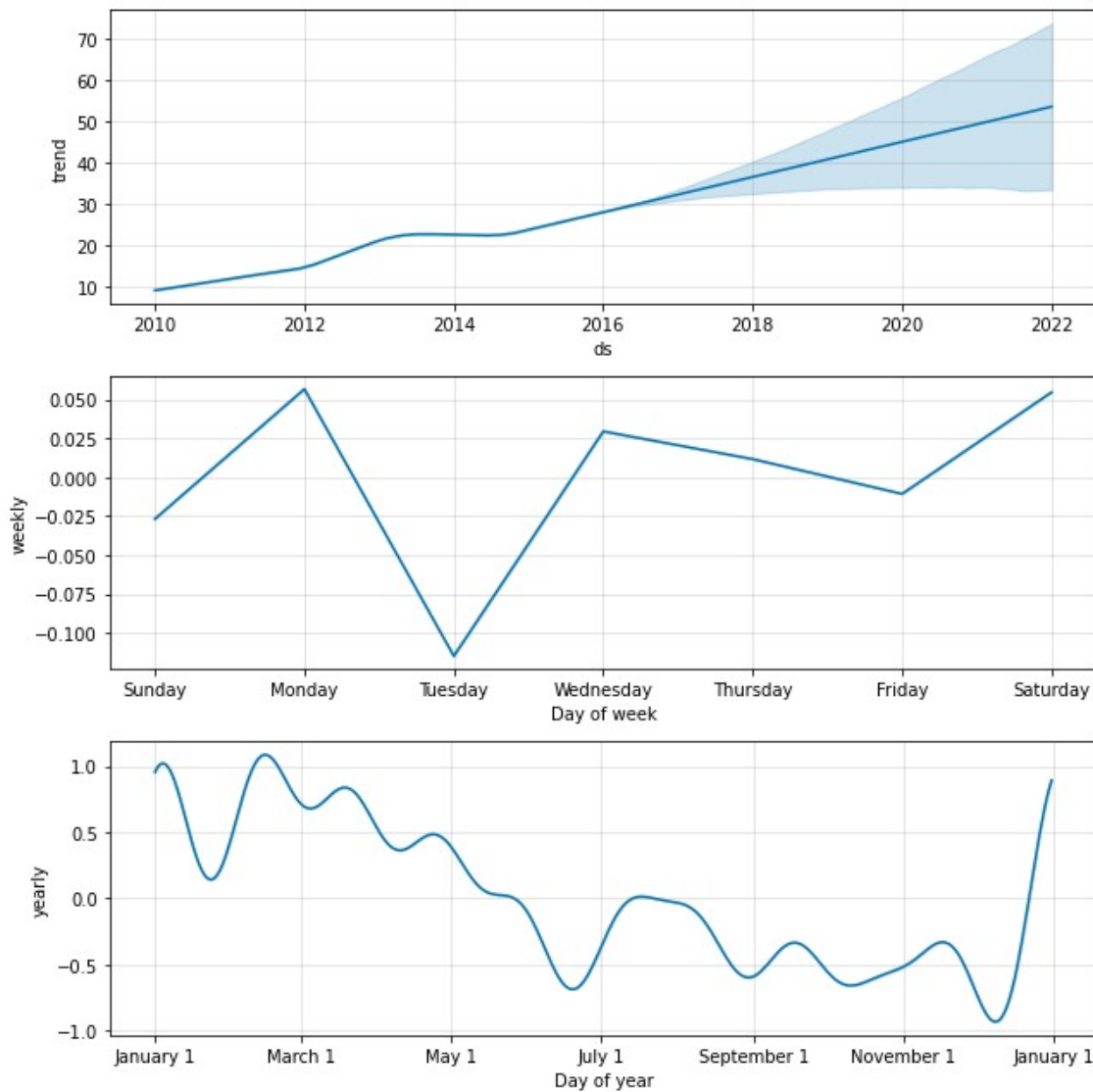
Out[16]:

	ds	yhat	yhat_lower	yhat_upper
1567	2021-08-31	51.403142	32.241986	70.387208
1568	2021-09-30	51.941725	32.410627	71.054065
1569	2021-10-31	52.274289	32.680658	71.810585
1570	2021-11-30	52.363842	32.731438	72.265999
1571	2021-12-31	54.419067	33.620227	74.614816

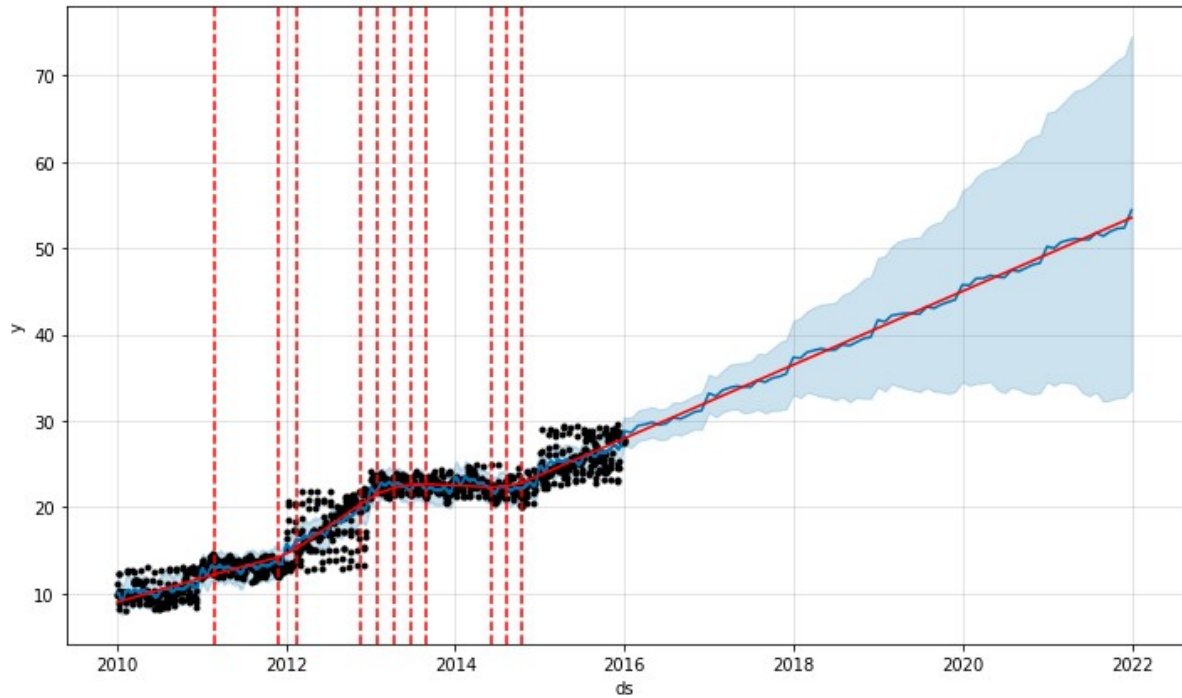
```
In [17]: #plotting out the results of forecast
fig1 = m.plot(forecast)
```



```
In [18]: #plotting out the components of the forecast namely, trend, weekly, yearly  
fig2 = m.plot_components(forecast)
```

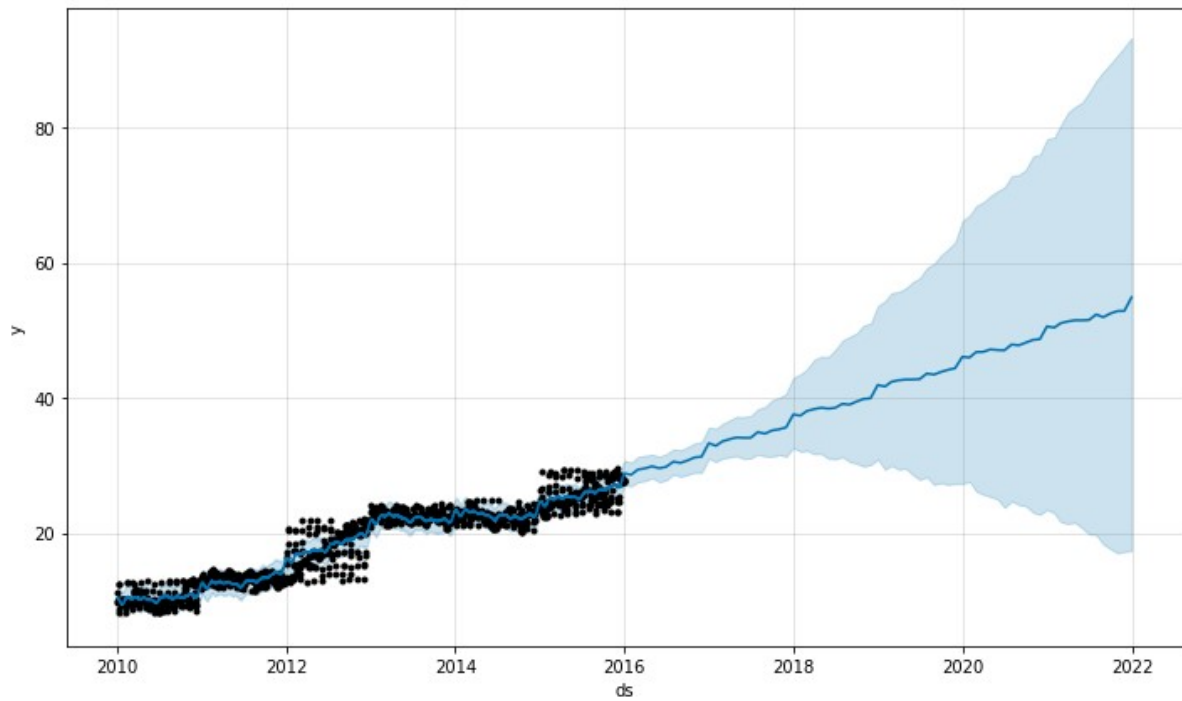


```
In [19]: #real time series frequently have abrupt changes in their trajectories.  
#By default, Prophet will automatically detect these changepoints and will allow th  
e trend to adapt appropriately.  
#However, if finer control over this process is required,  
#then there are several input arguments you can use.  
fig = m.plot(forecast)  
#adding changepoints  
a = add_changepoints_to_plot(fig.gca(), m, forecast)
```

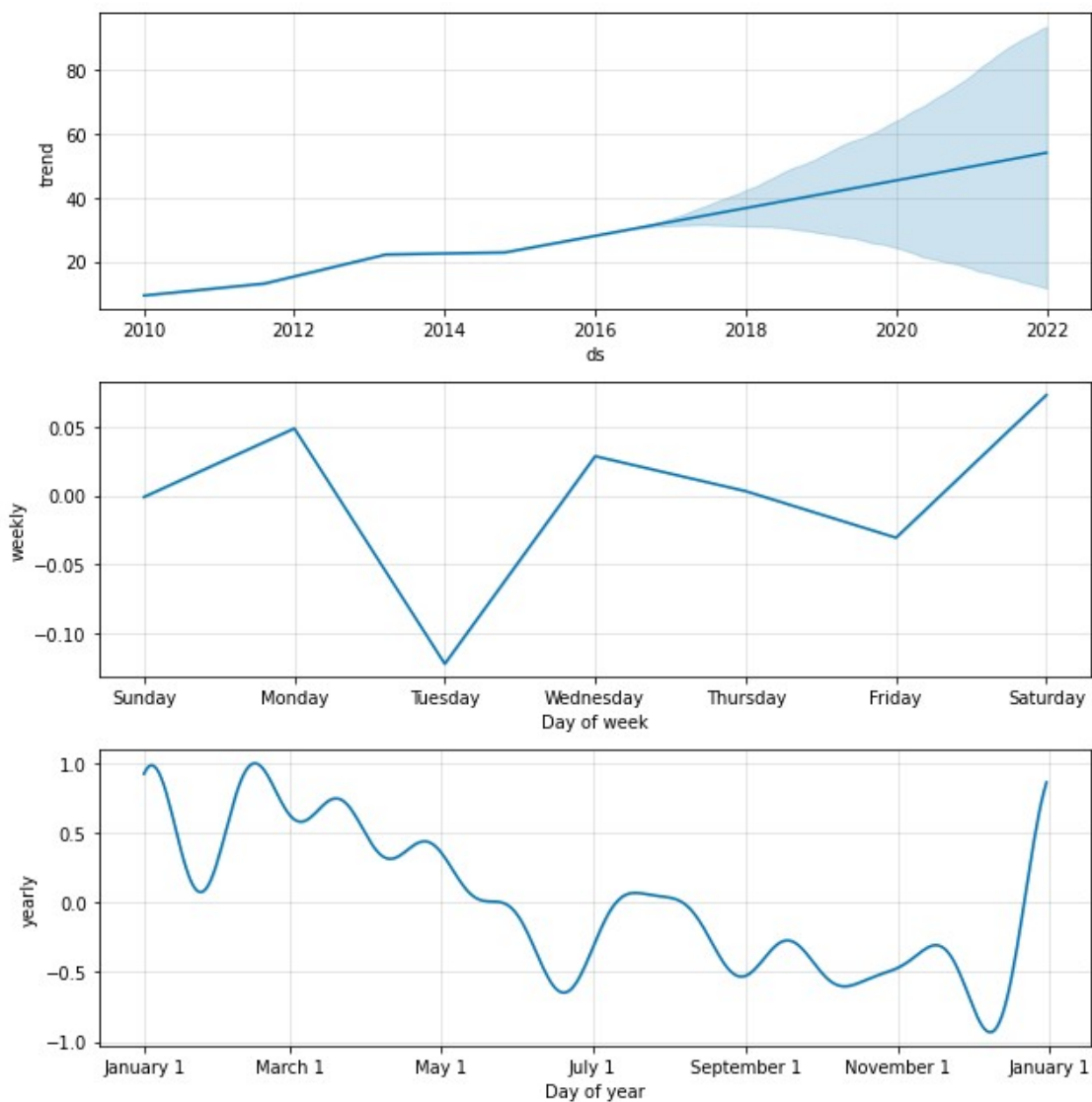


```
In [20]: #fine tuning the changepoints
m = Prophet(n_changepoints=3).fit(df)
#predicting for 6 years and each year include 12 months so the period is 12 * 6
future = m.make_future_dataframe(periods=12 * 6, freq='M')
#inputting the parameters created and forecasting on the basis of them
forecast = m.predict(future)
#plots out the forecast data
fig = m.plot(m.predict(future))
```

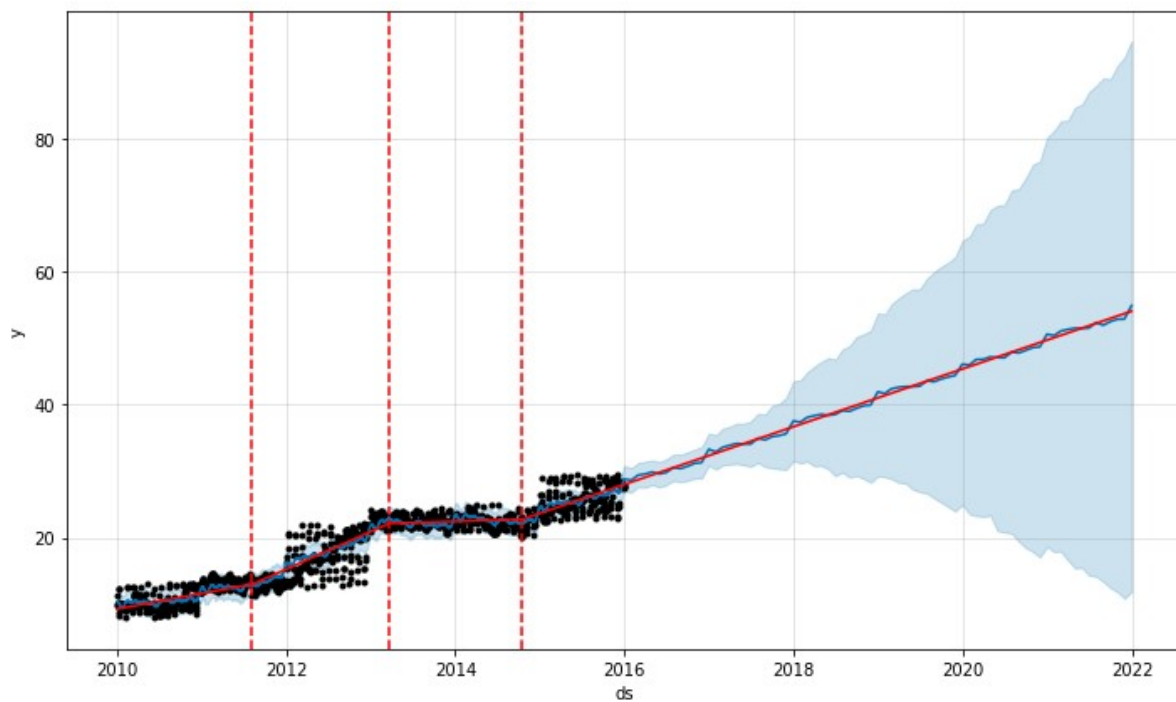
INFO:fbprophet:Disabling daily seasonality. Run prophet with daily_seasonality=T rue to override this.



```
In [21]: #plots out the different components and trends of the forecast done by the algorithm  
m  
fig2 = m.plot_components(forecast)
```

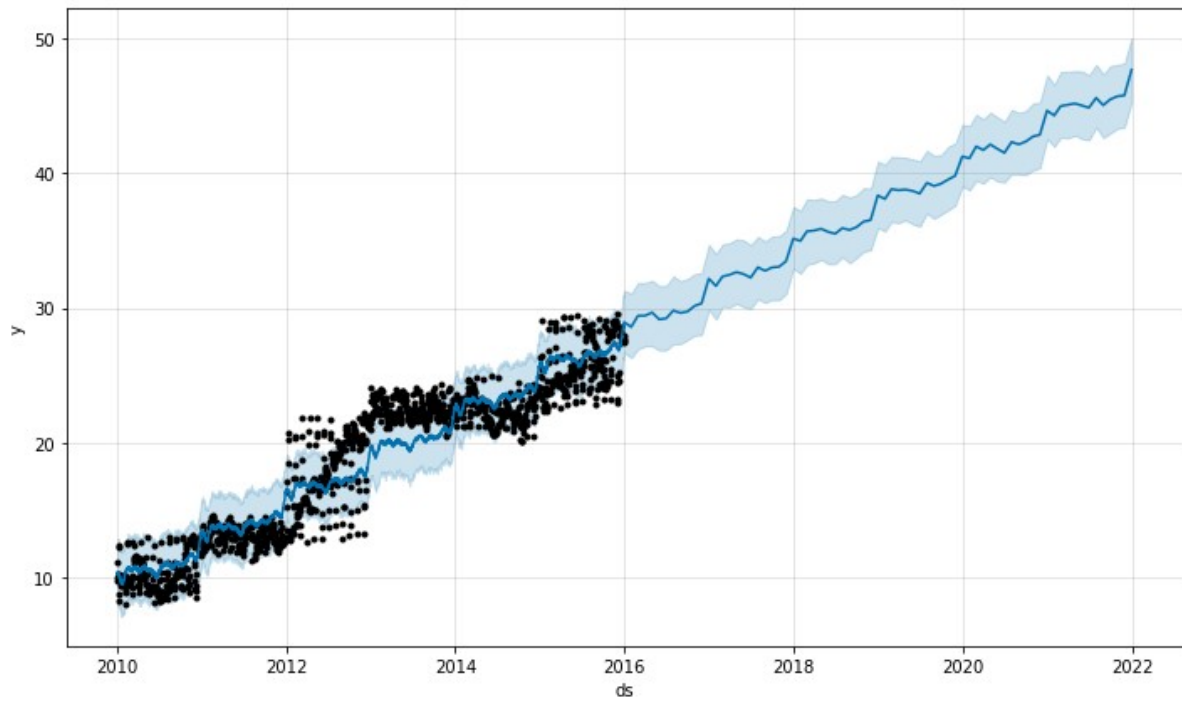


```
In [22]: #adding more changepoints so as to have a finer control over the abrupt changes in the time series
fig = m.plot(forecast)
a = add_changepoints_to_plot(fig.gca(), m, forecast)
```



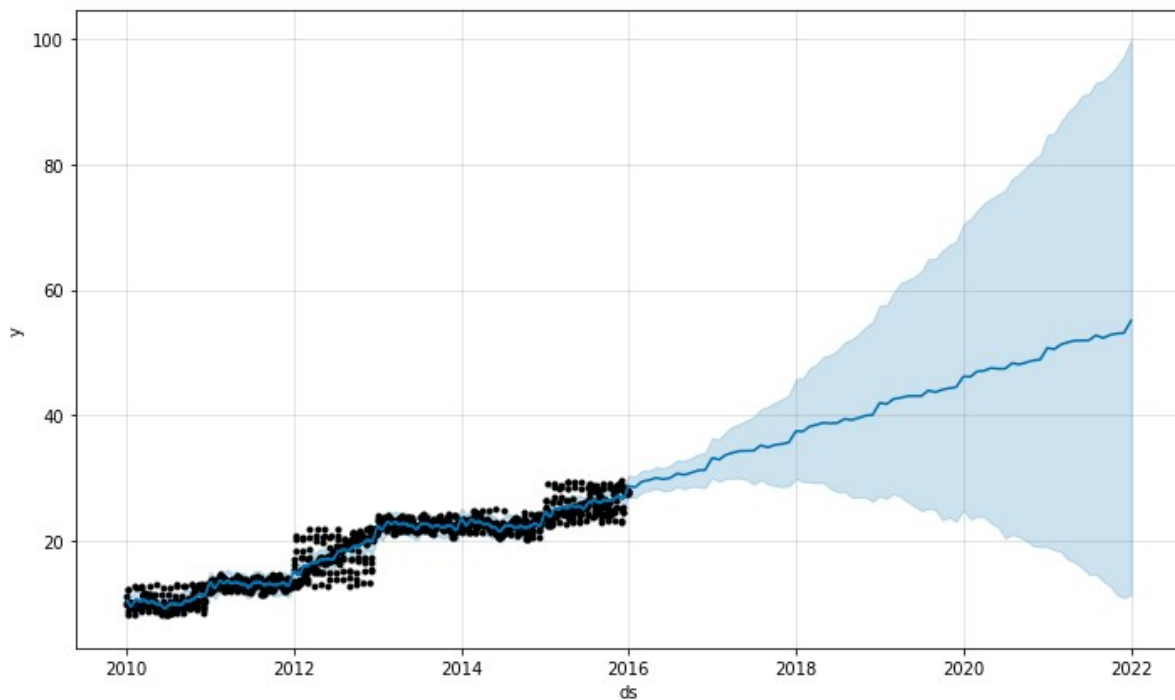
```
In [23]: #increasing the changepoints prior scale will decrease the uncertainty in the time series  
m = Prophet(changepoint_prior_scale=0.001).fit(df) #setting the prior scale to be 0.001  
future = m.make_future_dataframe(periods=12 * 6, freq='M')  
forecast = m.predict(future)  
#plotting the forecast plots  
fig = m.plot(forecast)
```

INFO:fbprophet:Disabling daily seasonality. Run prophet with daily_seasonality=True to override this.



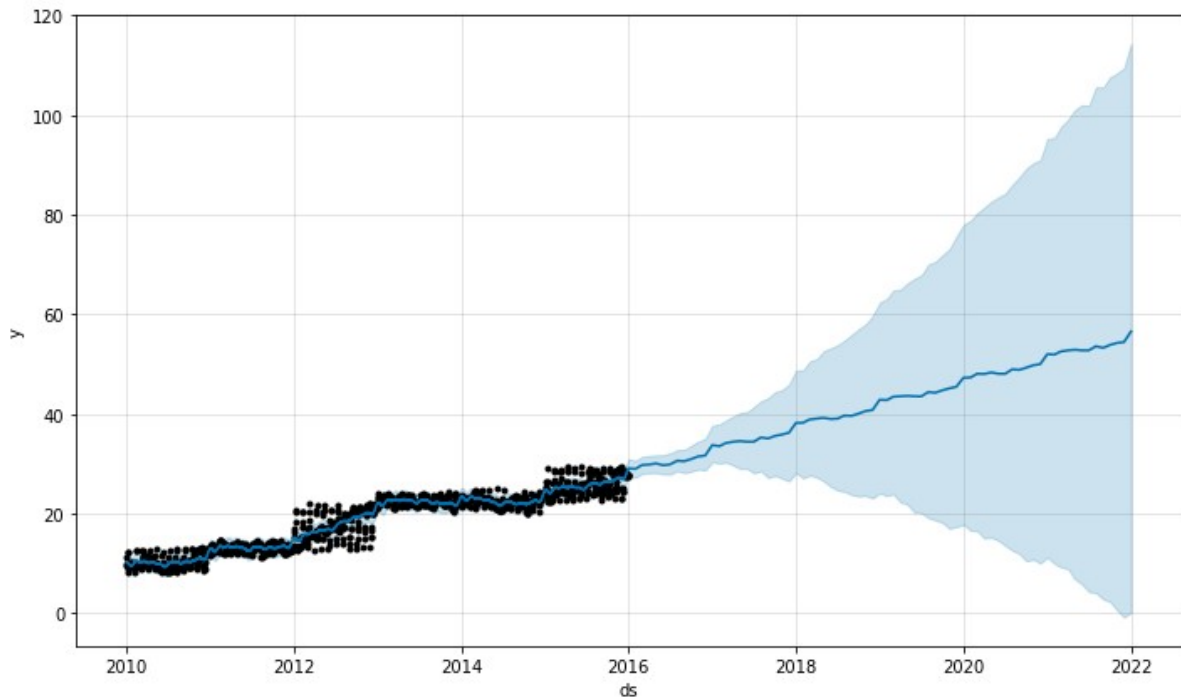

```
In [24]: m = Prophet(changepoint_prior_scale=0.5).fit(df) #increasing the prior scale to 0.5  
to see the changes in uncertainty  
future = m.make_future_dataframe(periods=12 * 6, freq='M') #M means monthly  
forecast = m.predict(future)  
fig = m.plot(forecast)
```

INFO:fbprophet:Disabling daily seasonality. Run prophet with daily_seasonality=T
true to override this.

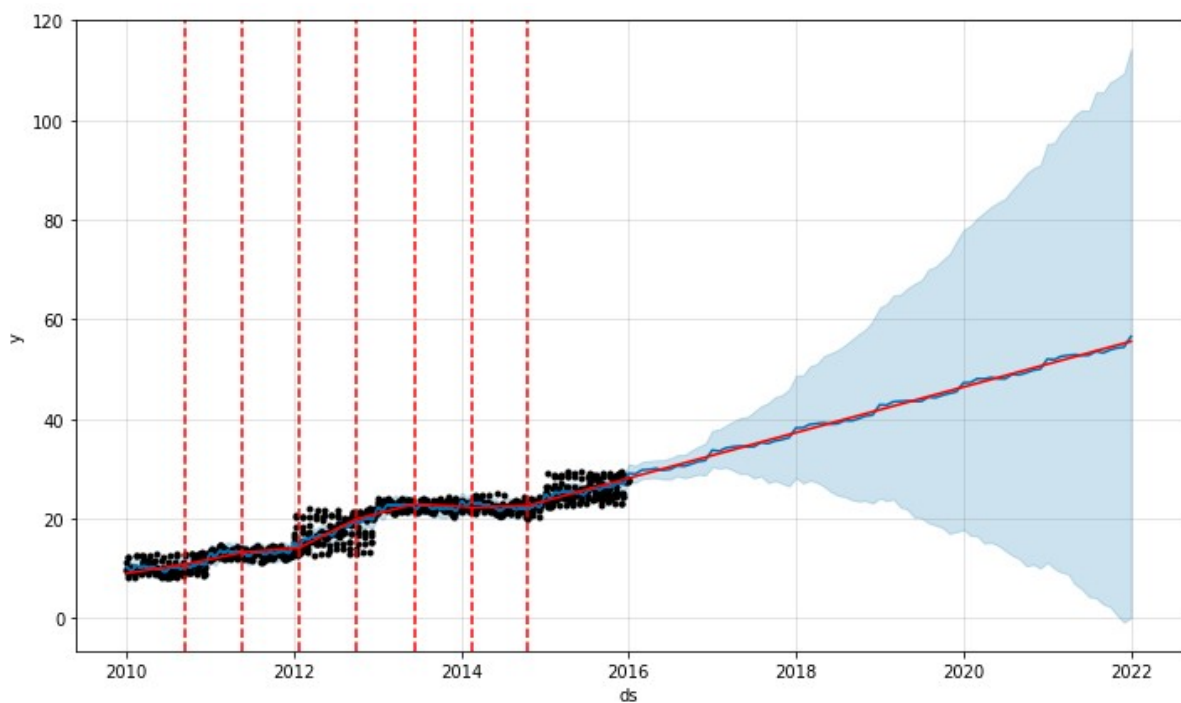


```
In [25]: m = Prophet(changepoint_prior_scale=0.5, n_changepoints=7).fit(df)
future = m.make_future_dataframe(periods=12 * 6, freq='M')
forecast = m.predict(future)
fig = m.plot(forecast)
```

INFO:fbprophet:Disabling daily seasonality. Run prophet with daily_seasonality=True to override this.



```
In [26]: #adding all the changepoints to the plot
fig = m.plot(forecast)
a = add_changepoints_to_plot(fig.gca(), m, forecast)
```



```
In [27]: #validating the dataset with a cutoff from the dataset
#validating the data from the error in the last 365 days predicted
#cutoff are the days that are removed from the dataset for cross validation
df_cv = cross_validation(m, initial='730 days', period='180 days', horizon = '365 d
ays')
df_cv.head()
```

INFO:fbprophet:Making 7 forecasts with cutoffs between 2012-01-16 00:00:00 and 2014-12-31 00:00:00

WARNING:fbprophet:Optimization terminated abnormally. Falling back to Newton.

Out[27]:

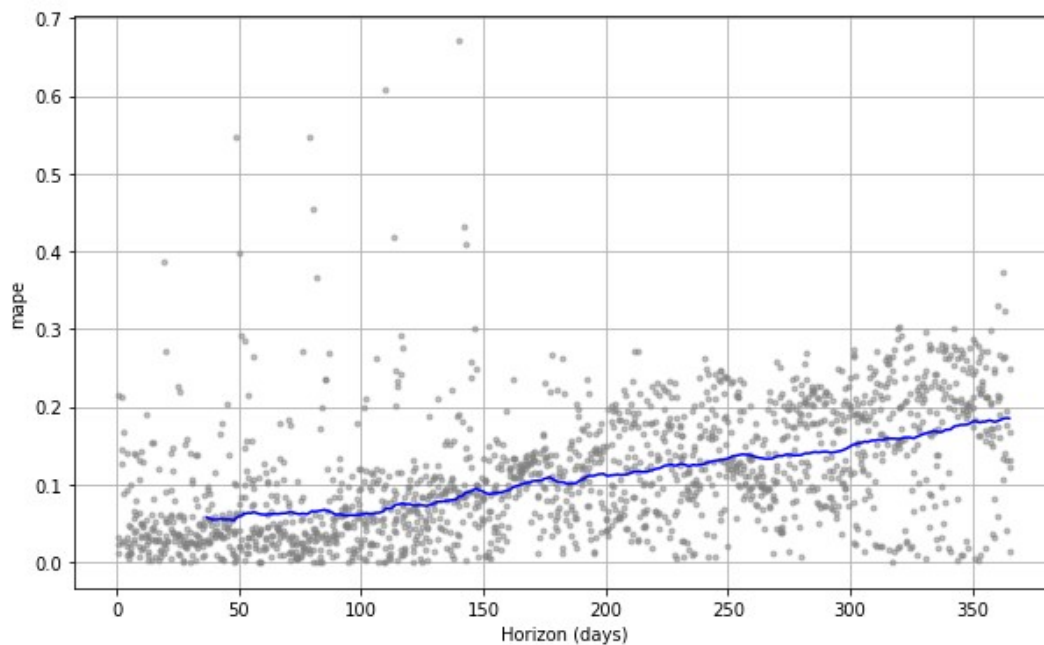
	ds	yhat	yhat_lower	yhat_upper	y	cutoff
0	2012-01-17	15.607826	14.269834	16.775655	12.849326	2012-01-16
1	2012-01-18	15.471241	14.218249	16.652972	12.769360	2012-01-16
2	2012-01-19	15.484002	14.122606	16.774869	13.261784	2012-01-16
3	2012-01-20	15.325889	14.063474	16.499539	13.438551	2012-01-16
4	2012-01-23	15.348673	14.017894	16.631261	13.463804	2012-01-16

```
In [28]: #getting the metrics like mean square error, root mean squared error
#these metrics define the performance of the prediction model created
df_p = performance_metrics(df_cv)
df_p.head()
```

Out[28]:

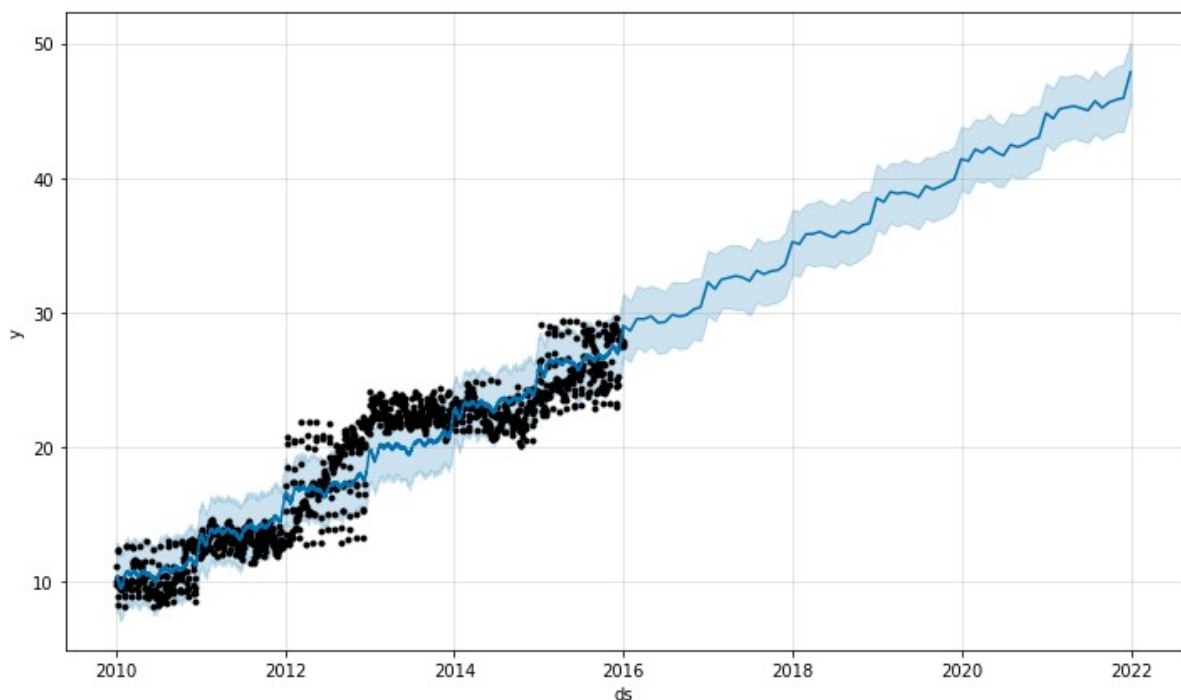
	horizon	mse	rmse	mae	mape	mdape	coverage
0	37 days	2.107013	1.451555	1.101506	0.057236	0.036574	0.777299
1	38 days	2.059845	1.435216	1.091294	0.056011	0.037122	0.784483
2	39 days	1.999956	1.414198	1.075286	0.054704	0.036881	0.795019
3	40 days	2.027633	1.423950	1.082736	0.054423	0.037416	0.797701
4	41 days	2.026547	1.423568	1.084951	0.054489	0.037416	0.798851

```
In [29]: #through mape plot we can see that the prediction error at first is less than 5% and then increases to about 5%
fig = plot_cross_validation_metric(df_cv, metric='mape')
```



```
In [30]: m = Prophet(changepoint_prior_scale=.001, n_changepoints=0).fit(df)
future = m.make_future_dataframe(periods=12 * 6, freq='M')
forecast = m.predict(future)
fig = m.plot(forecast)
```

INFO:fbprophet:Disabling daily seasonality. Run prophet with daily_seasonality=True to override this.
 WARNING:fbprophet:Optimization terminated abnormally. Falling back to Newton.

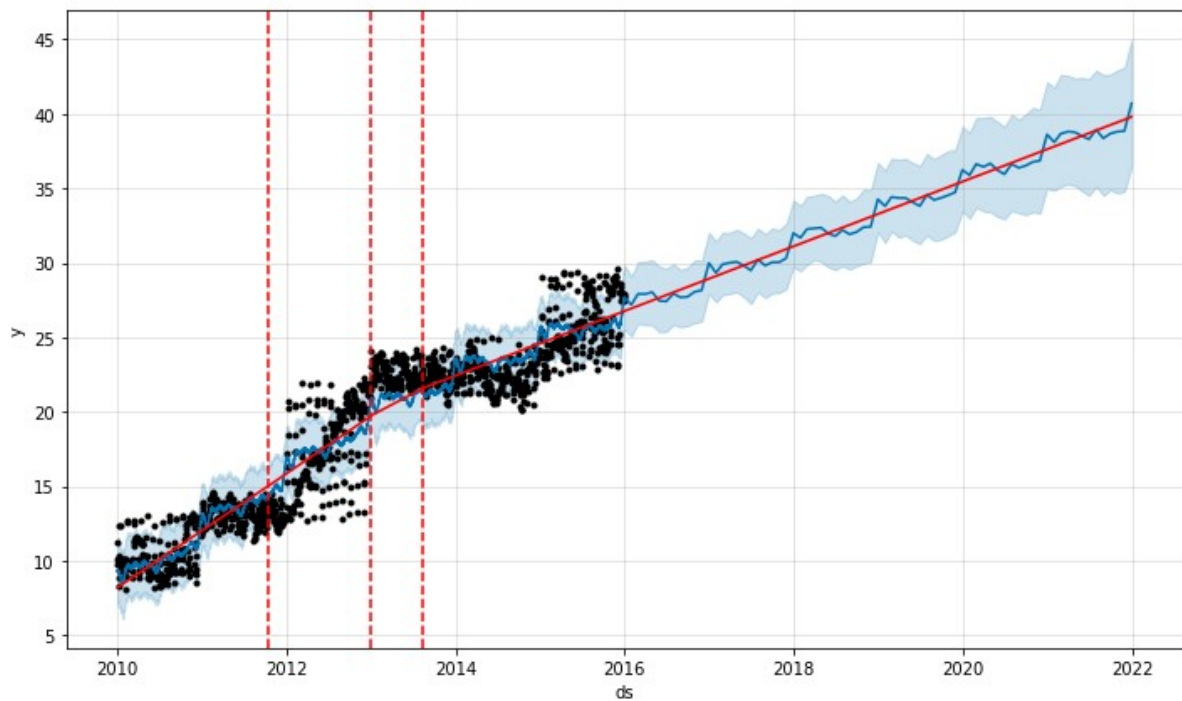


```
In [31]: #plotting the different changepoints scales and different numbers of changepoints
changepoint_prior_scales = [.005, .05, .5, 2]
n_changepoints = [8, 10, 15, 20, 25]
#creating an array of rmse
rmse = []
#for loop for looping through the differnt changepoint prior scales
for changepoint_prior_scale in changepoint_prior_scales:
    #for loop for looping through the number of changepoints
    for n_changepoint in n_changepoints:
        print('Changepoint Prior Scale:', changepoint_prior_scale) #prints which cha
nangepoint
        print('Number Changepoints:', n_changepoint) #prints the number of changepoi
nt
        m = Prophet(changepoint_prior_scale=changepoint_prior_scale, n_changepoints
=n_changepoint).fit(df) #fits the parameters to the algorithm
        future = m.make_future_dataframe(periods=12 * 6, freq='M') #defining the pre
diction periods
        forecast = m.predict(future)
        fig = m.plot(forecast) #plotting of the forecasting
        a = add_changepoints_to_plot(fig.gca(), m, forecast) #adding the changepoint
s to the gca plot
        plt.show()
        df_cv = cross_validation(m, initial='1095 days', period='180 days', horizon
= '365 days') #validates the prediction over the period of 365 days
        df_p = performance_metrics(df_cv) #aadding the performance metrics
        rmse.append((df_p['rmse'].mean(), {'changepoint_prior_scale': changepoint_p
rior_scale, 'n_changepoint': n_changepoint}))
        fig = plot_cross_validation_metric(df_cv, metric='mape') #plotting the cross
validation using the mape metric
        plt.show()
```

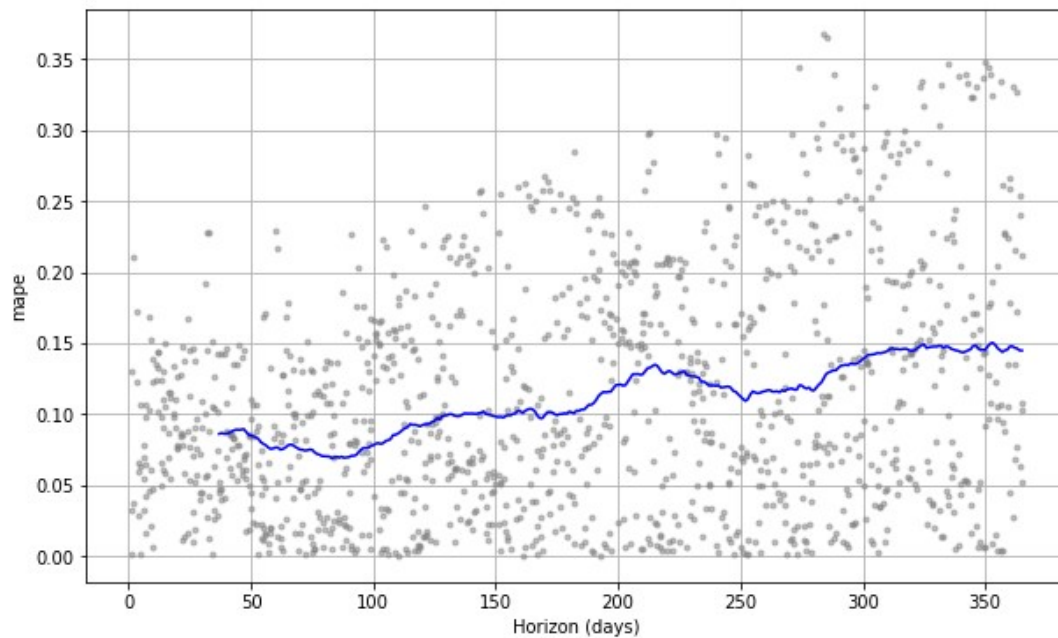
INFO:fbprophet:Disabling daily seasonality. Run prophet with daily_seasonality=True to override this.

Changepoint Prior Scale: 0.005

Number Changepoints: 8



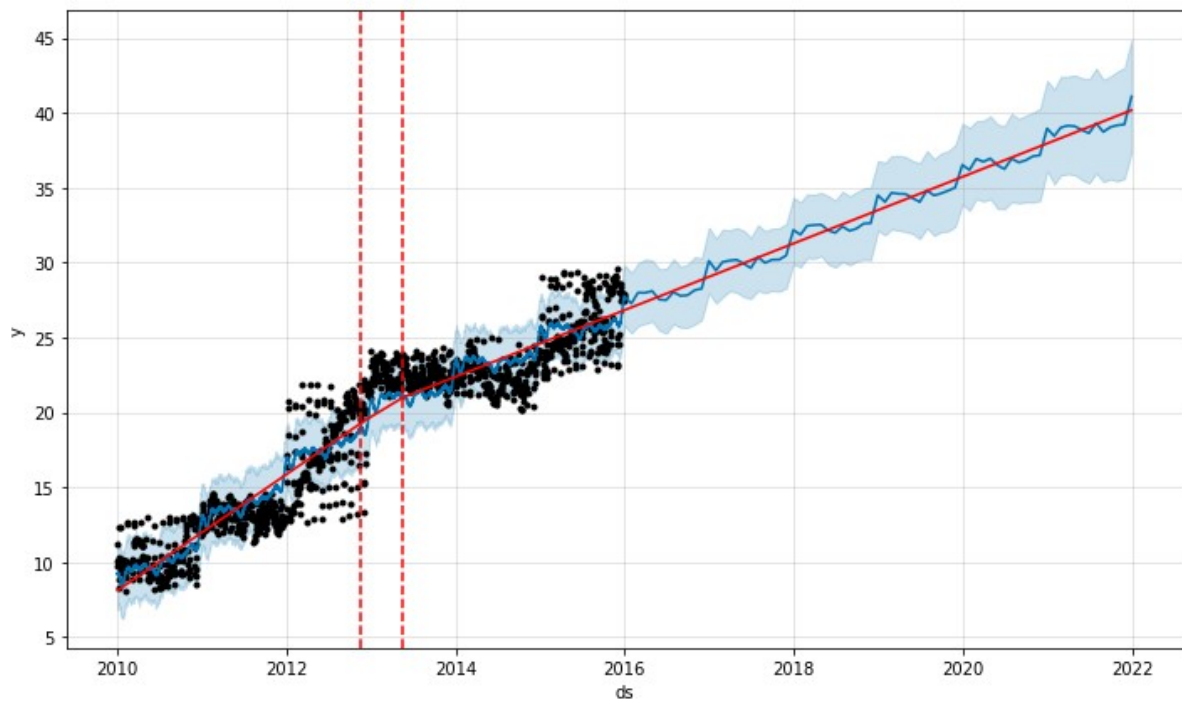
INFO:fbprophet:Making 5 forecasts with cutoffs between 2013-01-10 00:00:00 and 2014-12-31 00:00:00



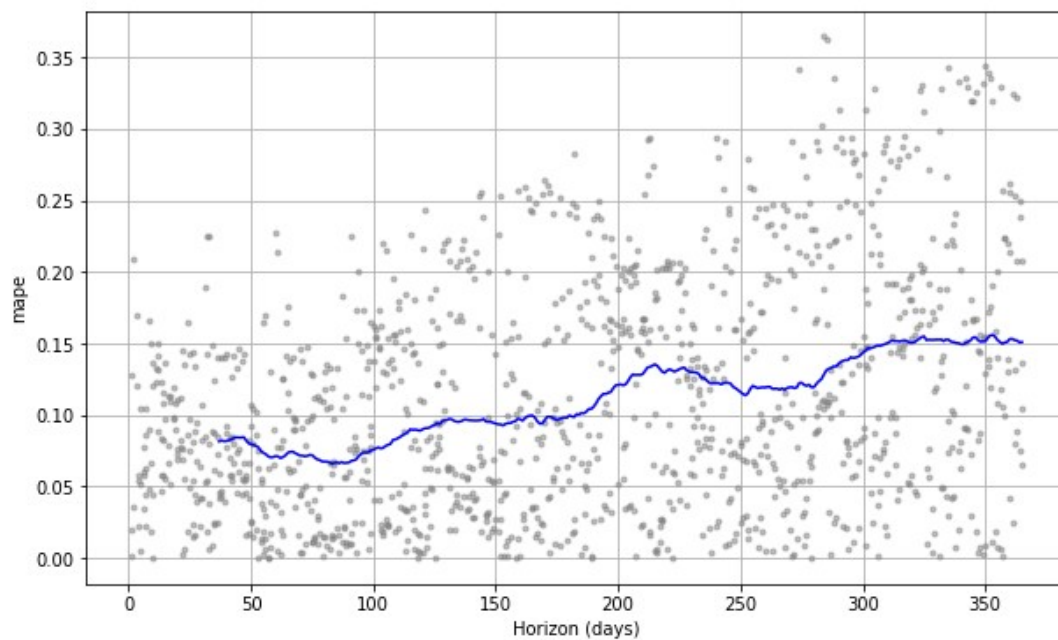
INFO:fbprophet:Disabling daily seasonality. Run prophet with daily_seasonality=True to override this.

Changepoint Prior Scale: 0.005

Number Changepoints: 10

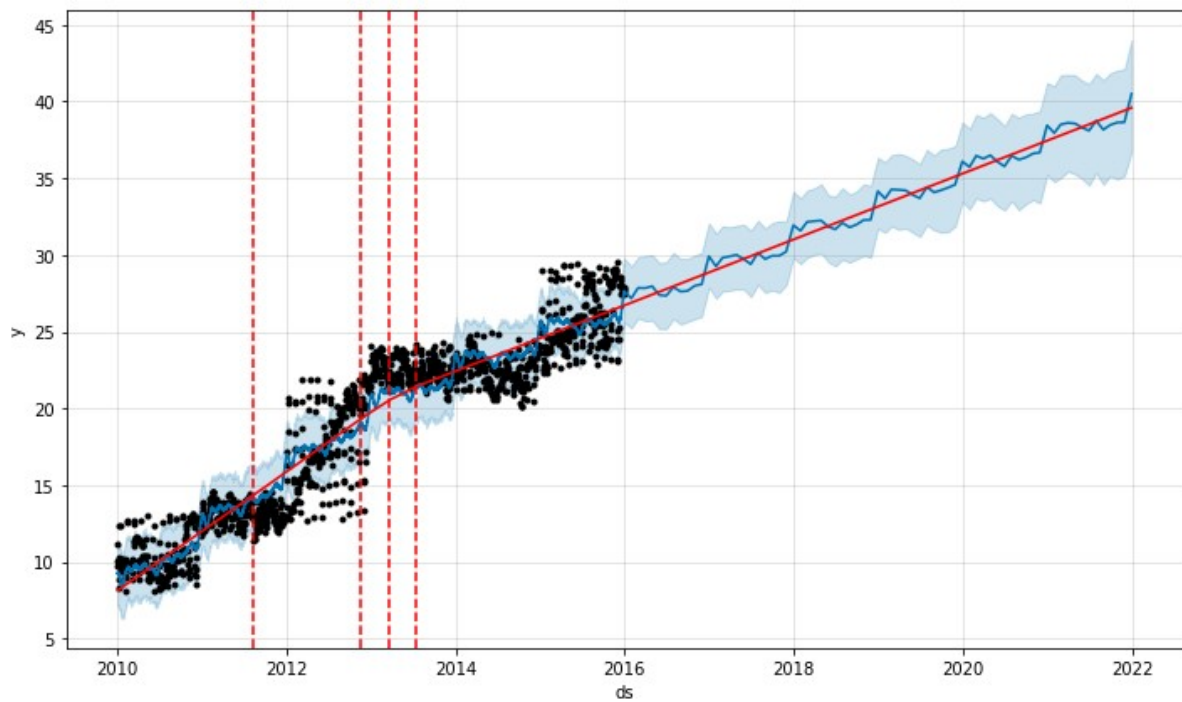


INFO:fbprophet:Making 5 forecasts with cutoffs between 2013-01-10 00:00:00 and 2014-12-31 00:00:00

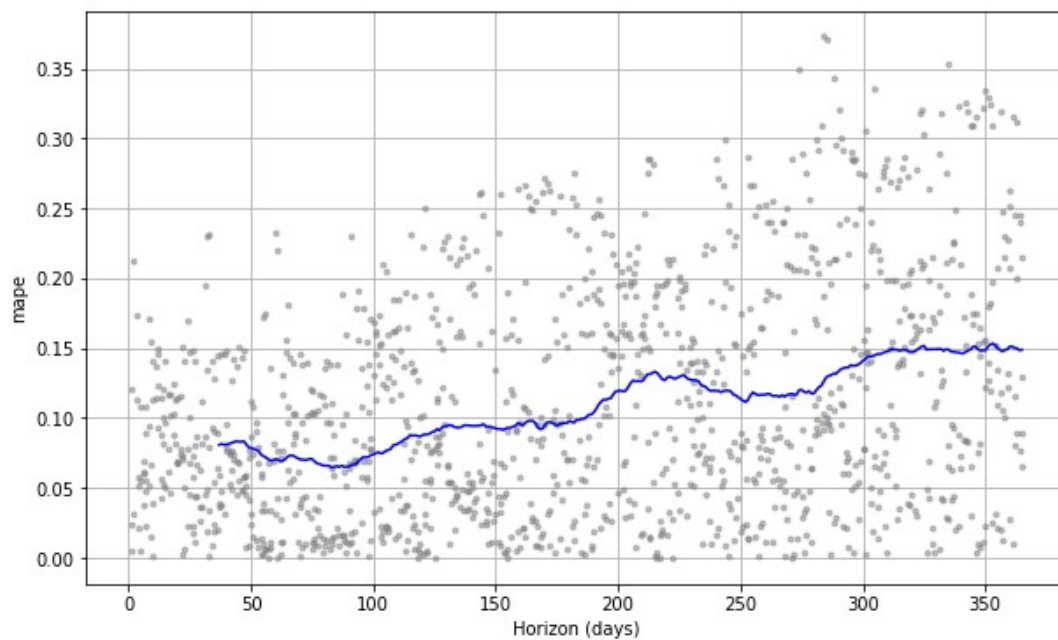


INFO:fbprophet:Disabling daily seasonality. Run prophet with daily_seasonality=True to override this.

Changepoint Prior Scale: 0.005
Number Changepoints: 15

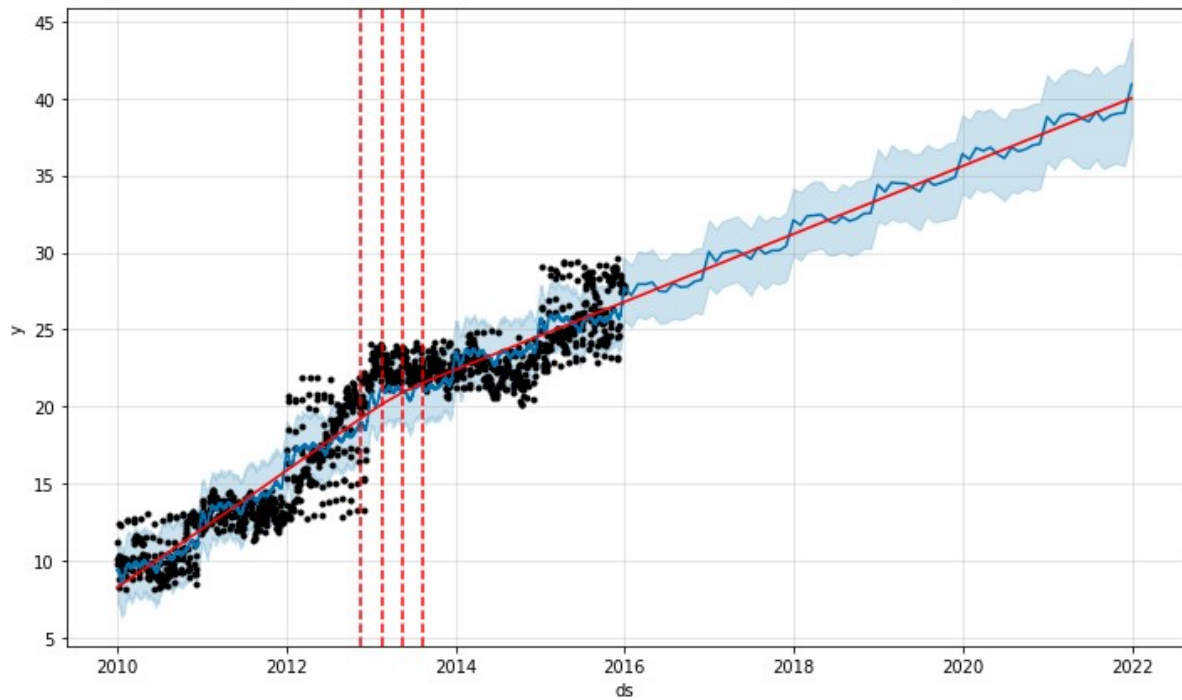


INFO:fbprophet:Making 5 forecasts with cutoffs between 2013-01-10 00:00:00 and 2014-12-31 00:00:00

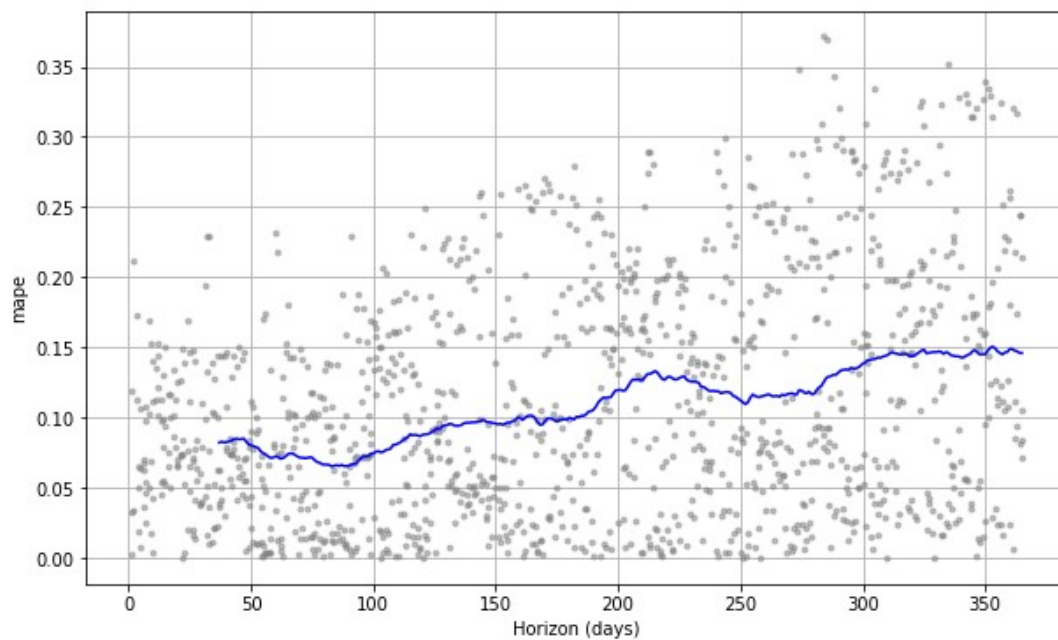


INFO:fbprophet:Disabling daily seasonality. Run prophet with daily_seasonality=True to override this.

Changepoint Prior Scale: 0.005
Number Changepoints: 20

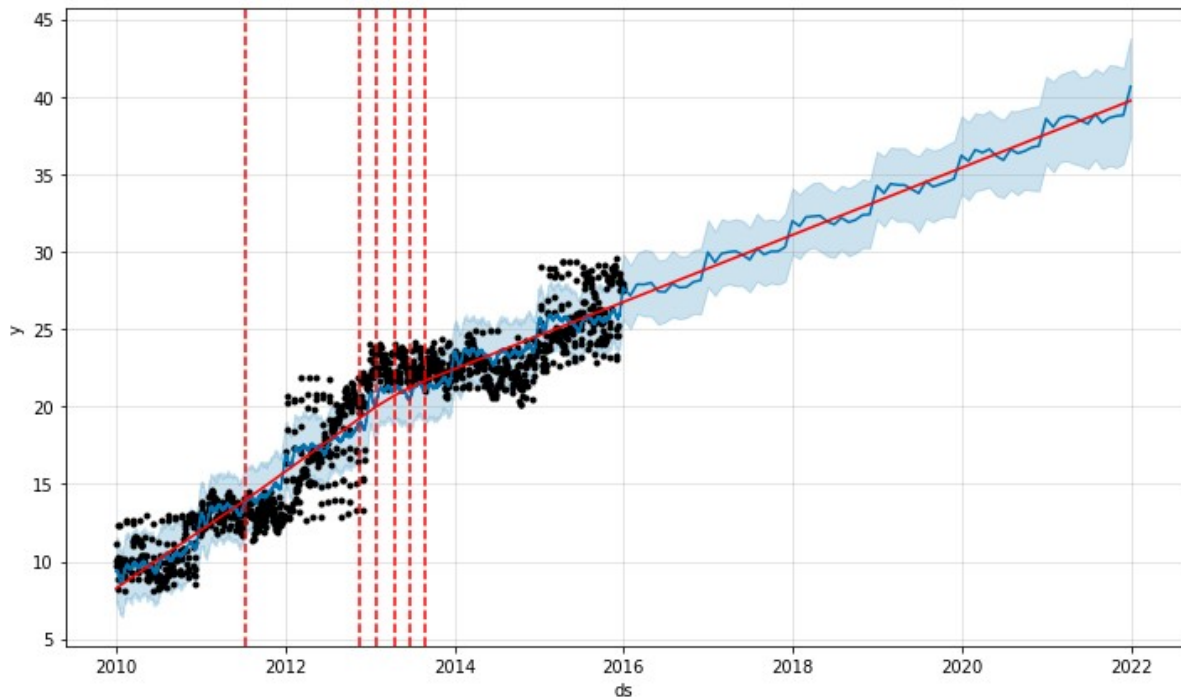


INFO:fbprophet:Making 5 forecasts with cutoffs between 2013-01-10 00:00:00 and 2014-12-31 00:00:00

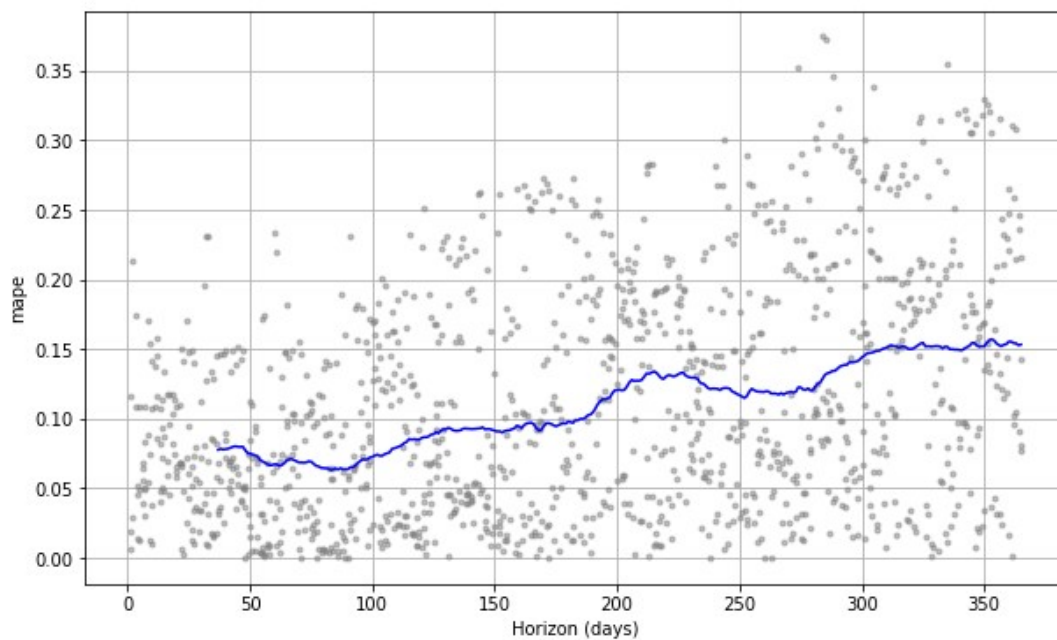


INFO:fbprophet:Disabling daily seasonality. Run prophet with daily_seasonality=True to override this.

Changepoint Prior Scale: 0.005
Number Changepoints: 25



INFO:fbprophet:Making 5 forecasts with cutoffs between 2013-01-10 00:00:00 and 2014-12-31 00:00:00

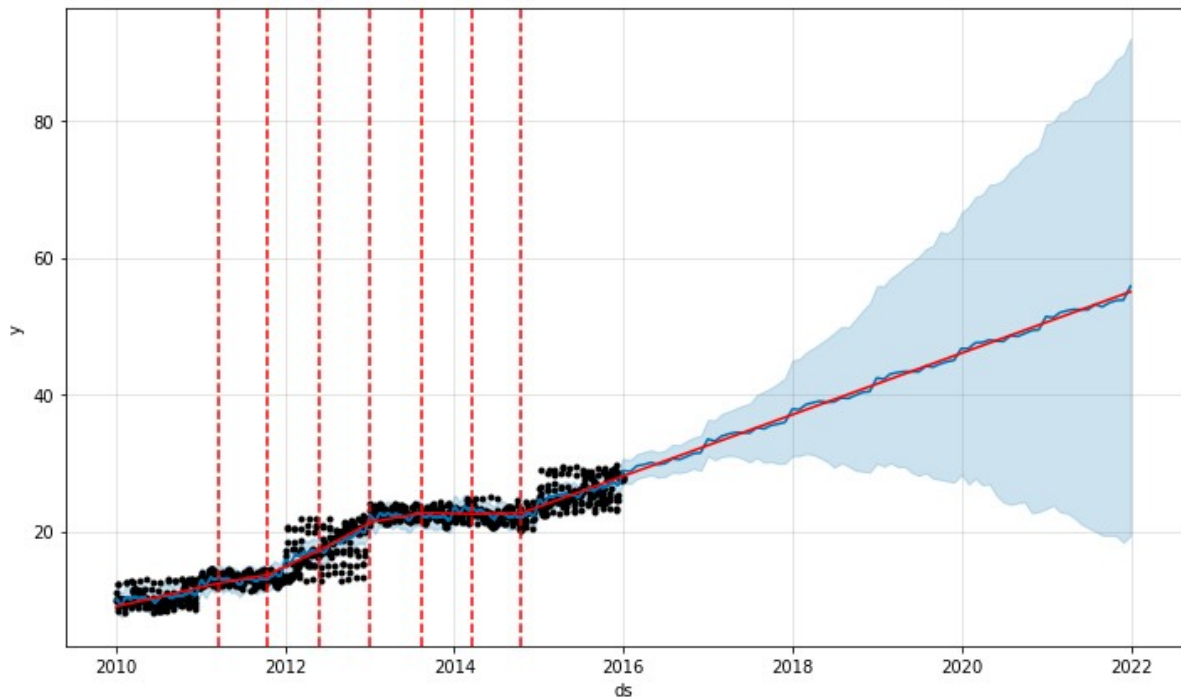


INFO:fbprophet:Disabling daily seasonality. Run prophet with daily_seasonality=True to override this.

Changepoint Prior Scale: 0.05

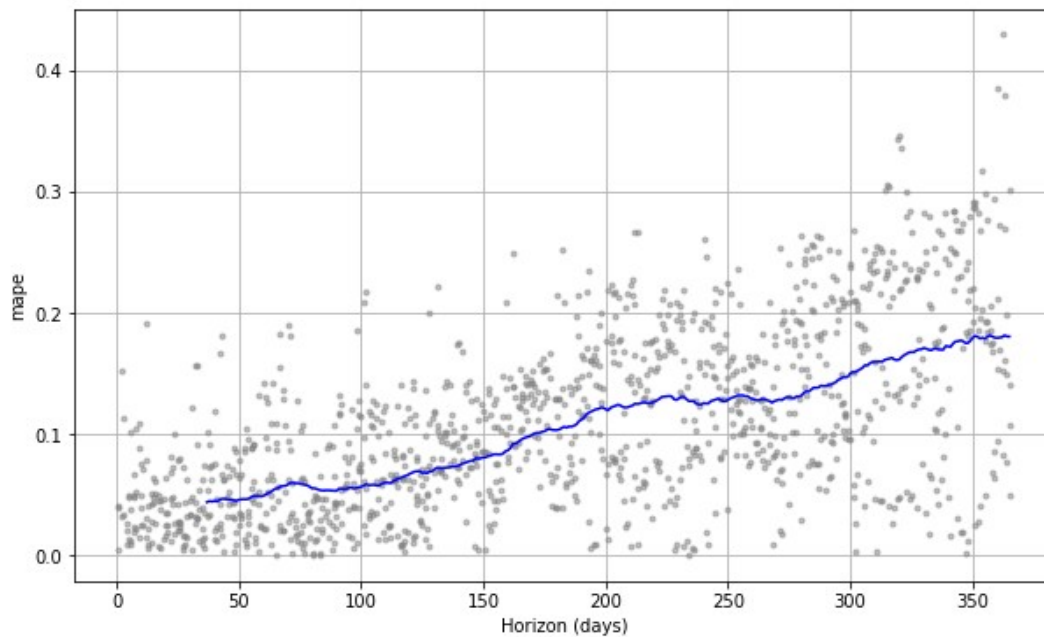
Number Changepoints: 8

WARNING:fbprophet:Optimization terminated abnormally. Falling back to Newton.



INFO:fbprophet:Making 5 forecasts with cutoffs between 2013-01-10 00:00:00 and 2014-12-31 00:00:00

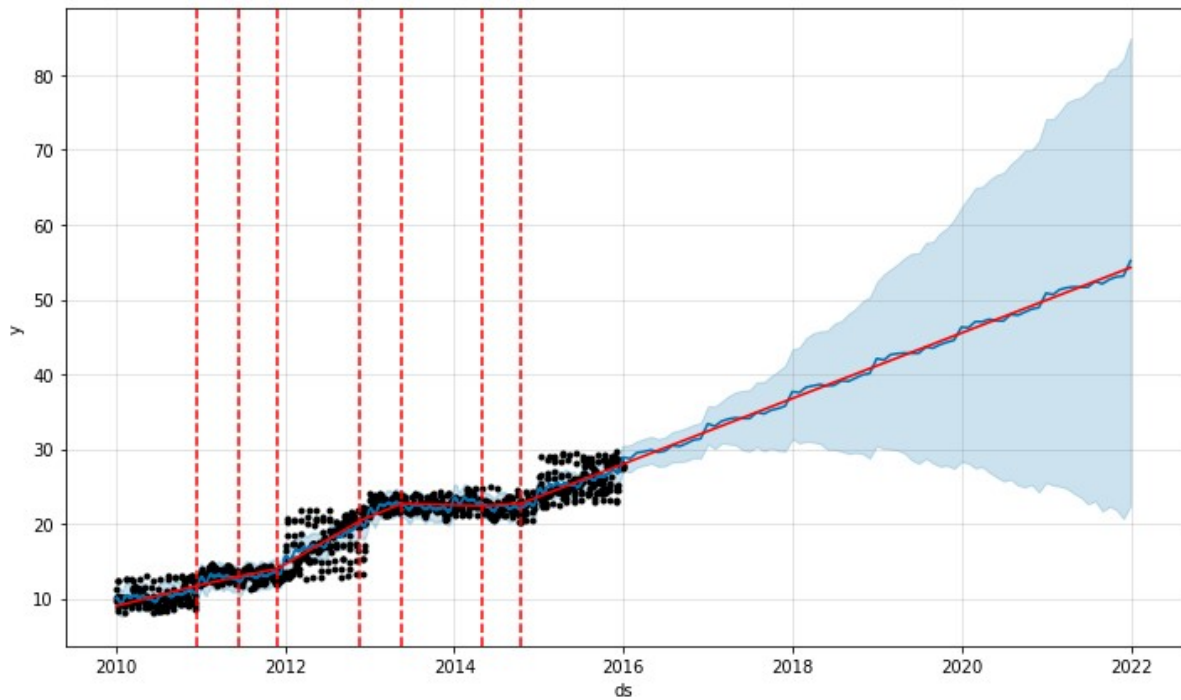
WARNING:fbprophet:Optimization terminated abnormally. Falling back to Newton.



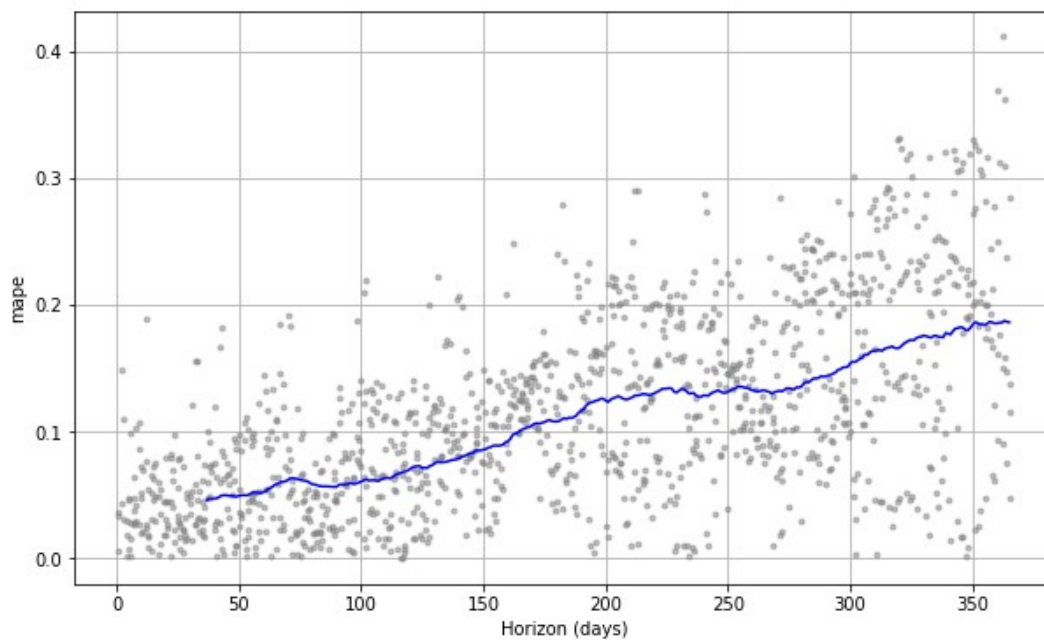
INFO:fbprophet:Disabling daily seasonality. Run prophet with daily_seasonality=True to override this.

Changepoint Prior Scale: 0.05

Number Changepoints: 10



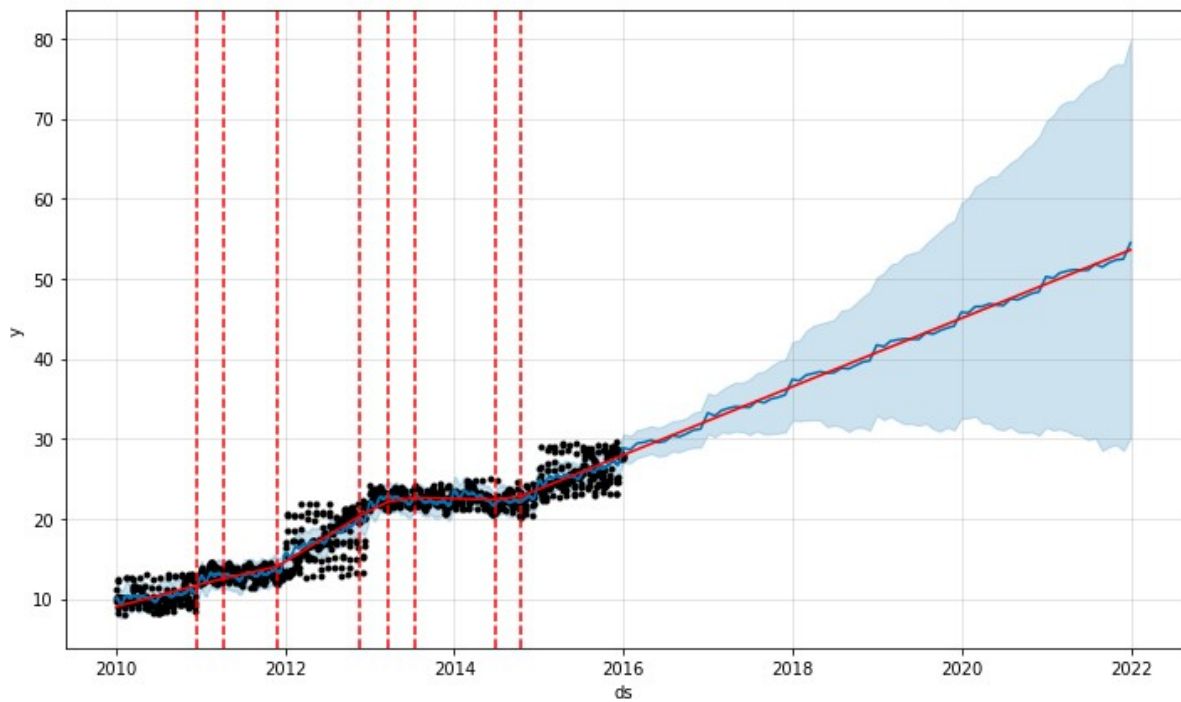
INFO:fbprophet:Making 5 forecasts with cutoffs between 2013-01-10 00:00:00 and 2014-12-31 00:00:00



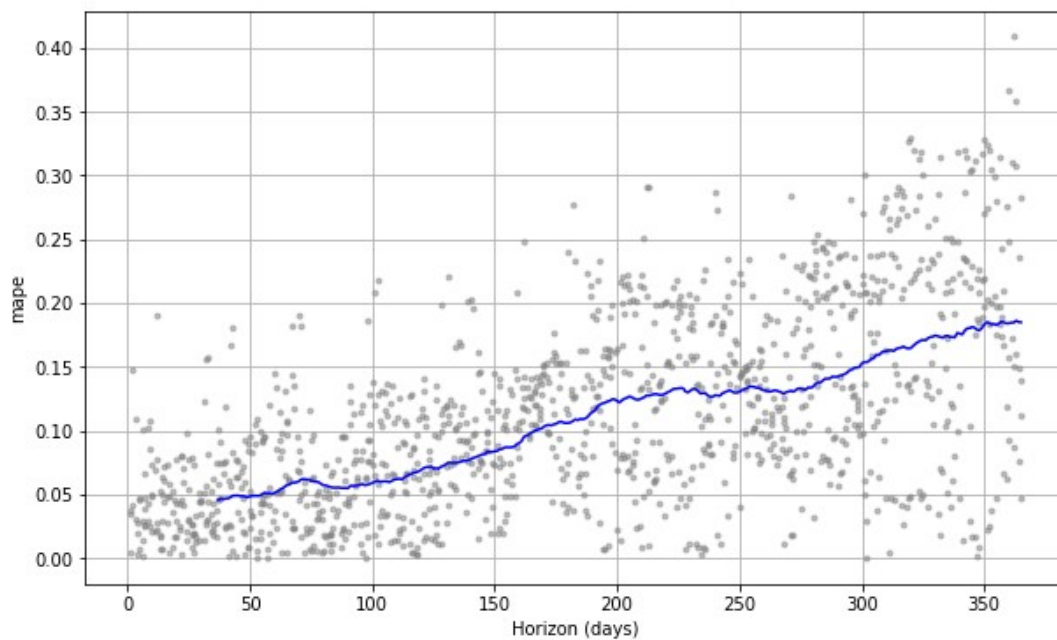
INFO:fbprophet:Disabling daily seasonality. Run prophet with daily_seasonality=True to override this.

Changepoint Prior Scale: 0.05

Number Changepoints: 15

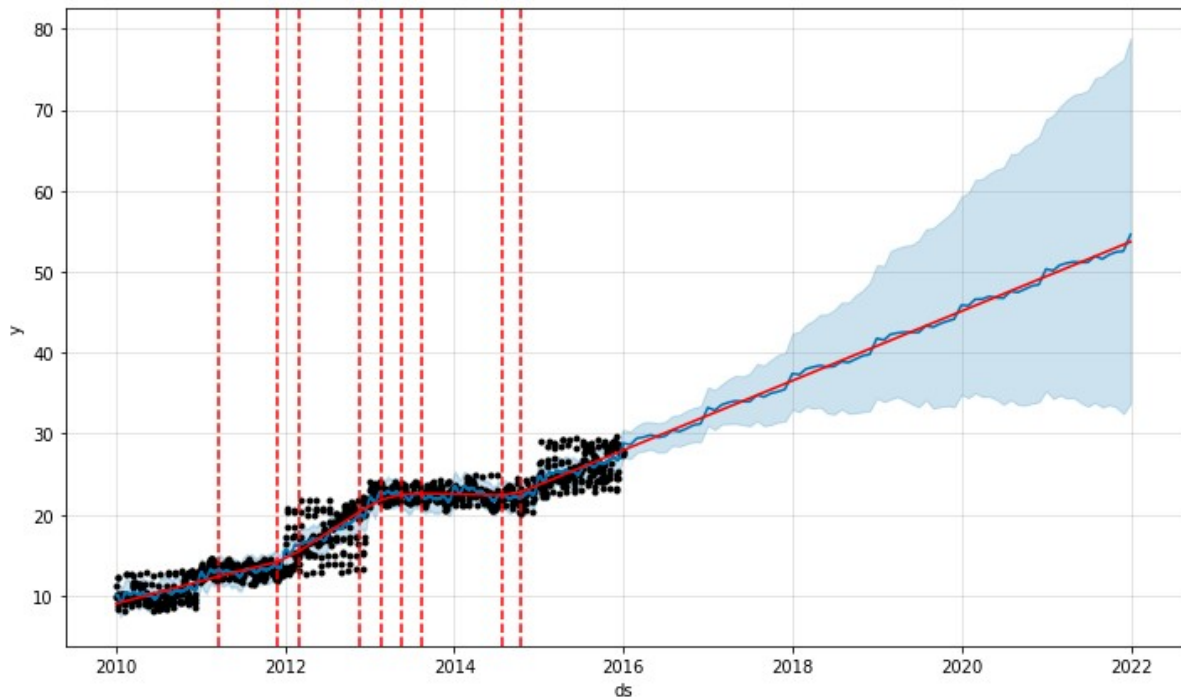


INFO:fbprophet:Making 5 forecasts with cutoffs between 2013-01-10 00:00:00 and 2014-12-31 00:00:00

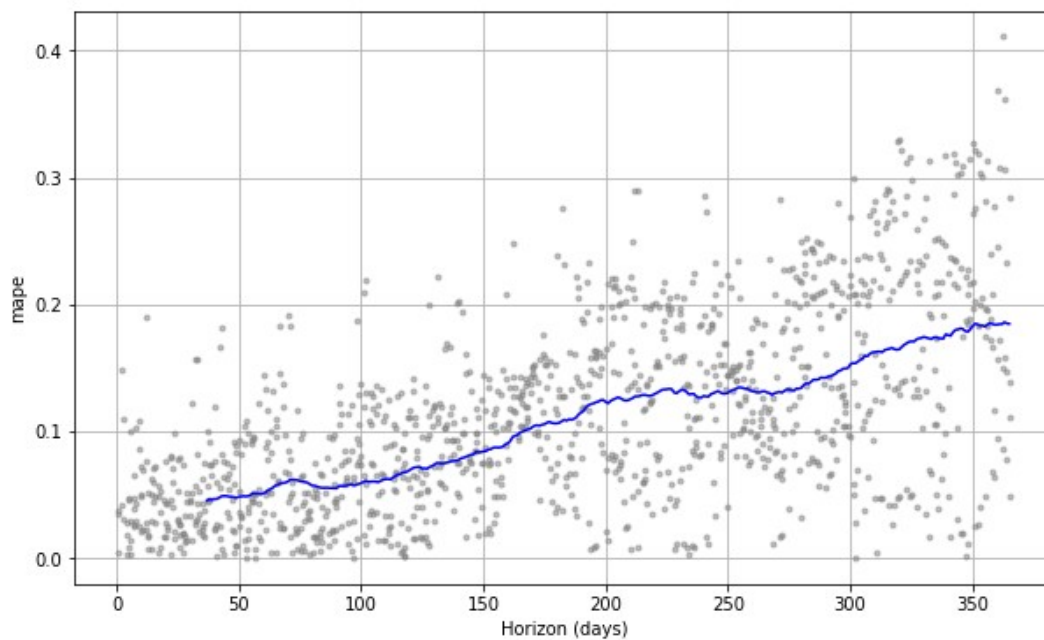


INFO:fbprophet:Disabling daily seasonality. Run prophet with daily_seasonality=True to override this.

Changepoint Prior Scale: 0.05
Number Changepoints: 20

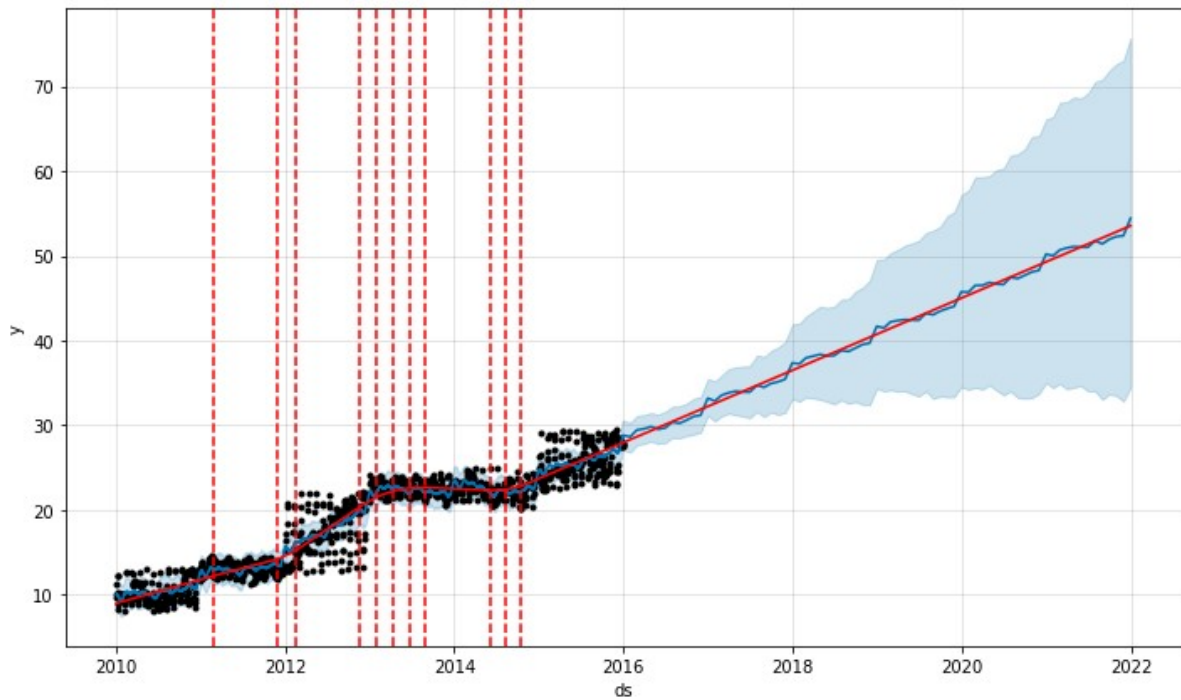


INFO:fbprophet:Making 5 forecasts with cutoffs between 2013-01-10 00:00:00 and 2014-12-31 00:00:00

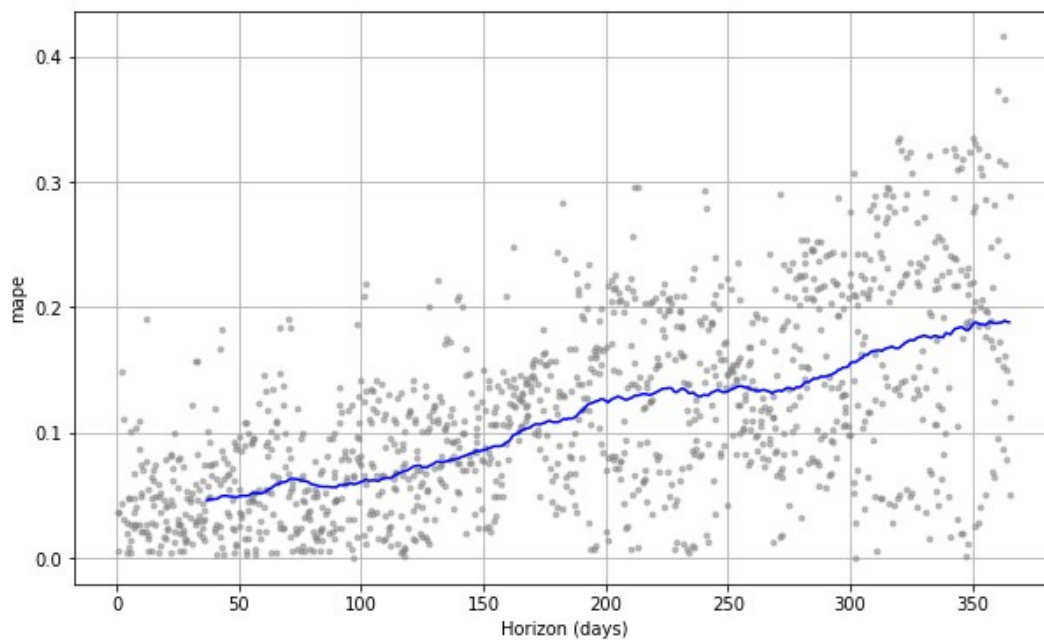


INFO:fbprophet:Disabling daily seasonality. Run prophet with daily_seasonality=True to override this.

Changepoint Prior Scale: 0.05
Number Changepoints: 25

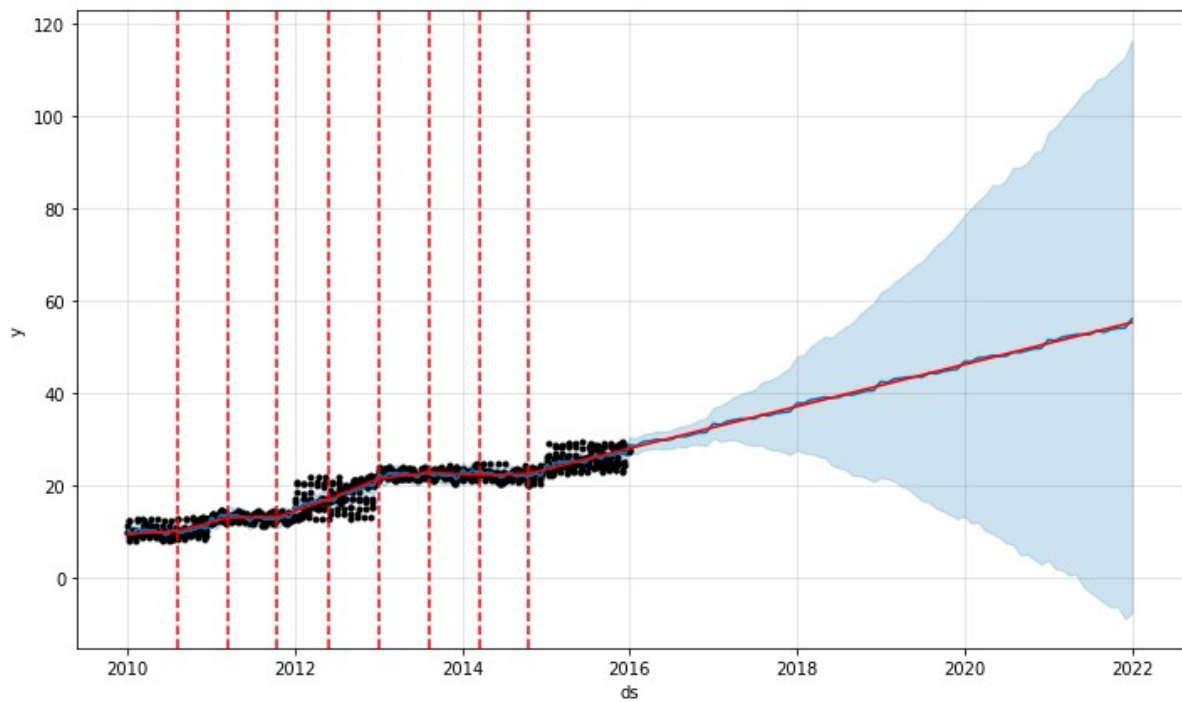


INFO:fbprophet:Making 5 forecasts with cutoffs between 2013-01-10 00:00:00 and 2014-12-31 00:00:00

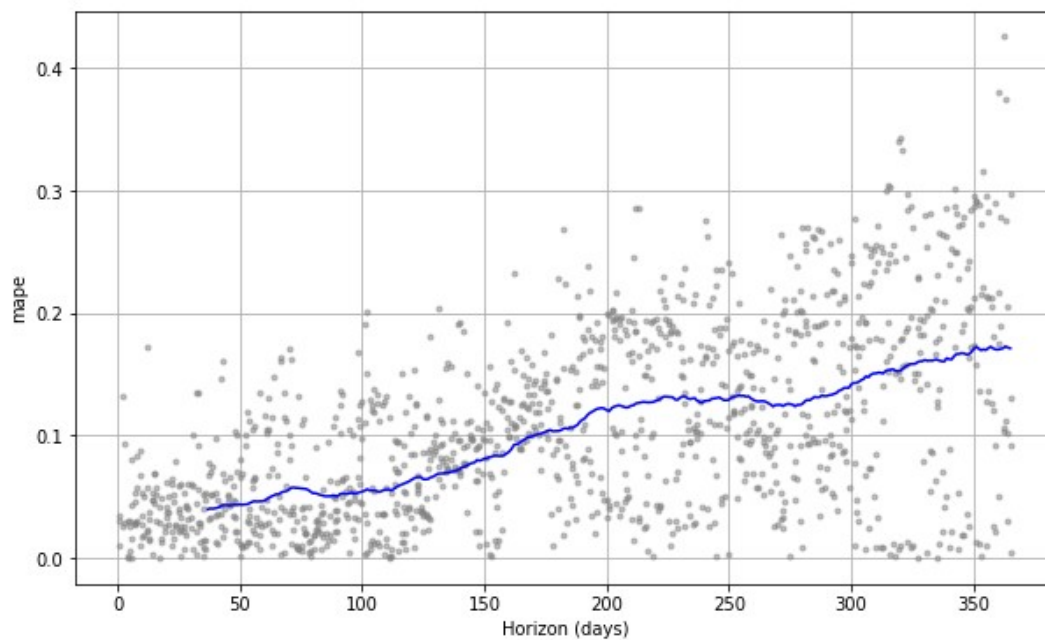


INFO:fbprophet:Disabling daily seasonality. Run prophet with daily_seasonality=True to override this.

Changepoint Prior Scale: 0.5
Number Changepoints: 8

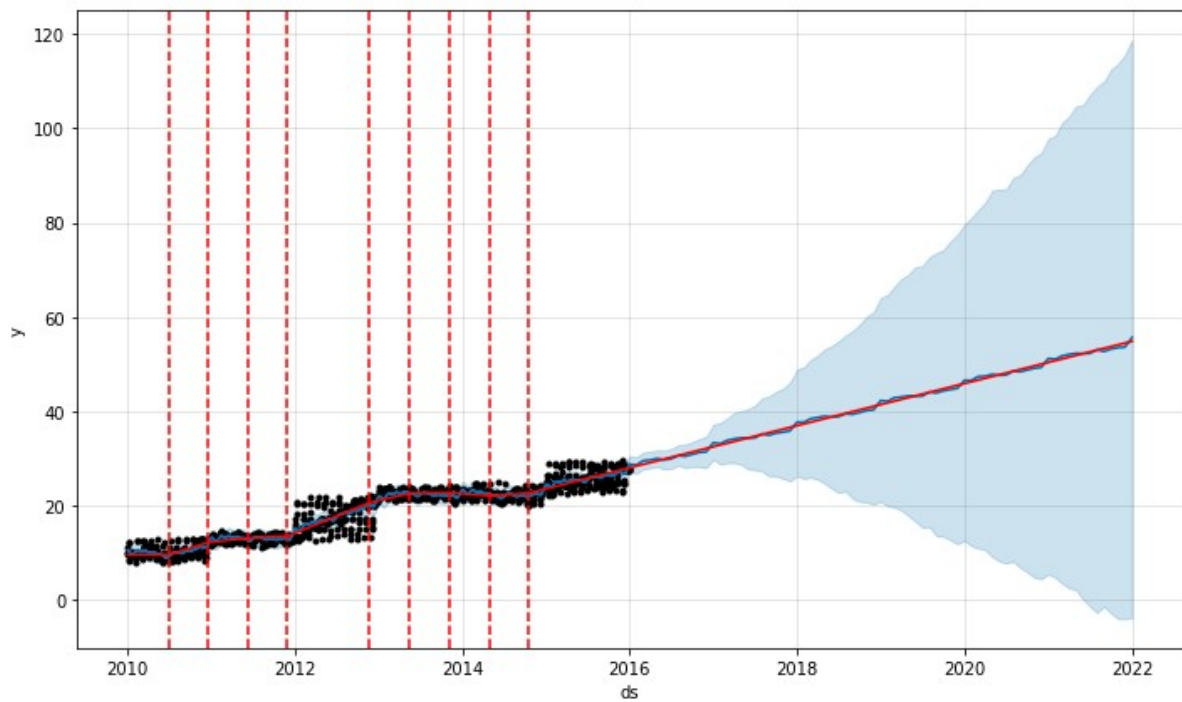


INFO:fbprophet:Making 5 forecasts with cutoffs between 2013-01-10 00:00:00 and 2014-12-31 00:00:00

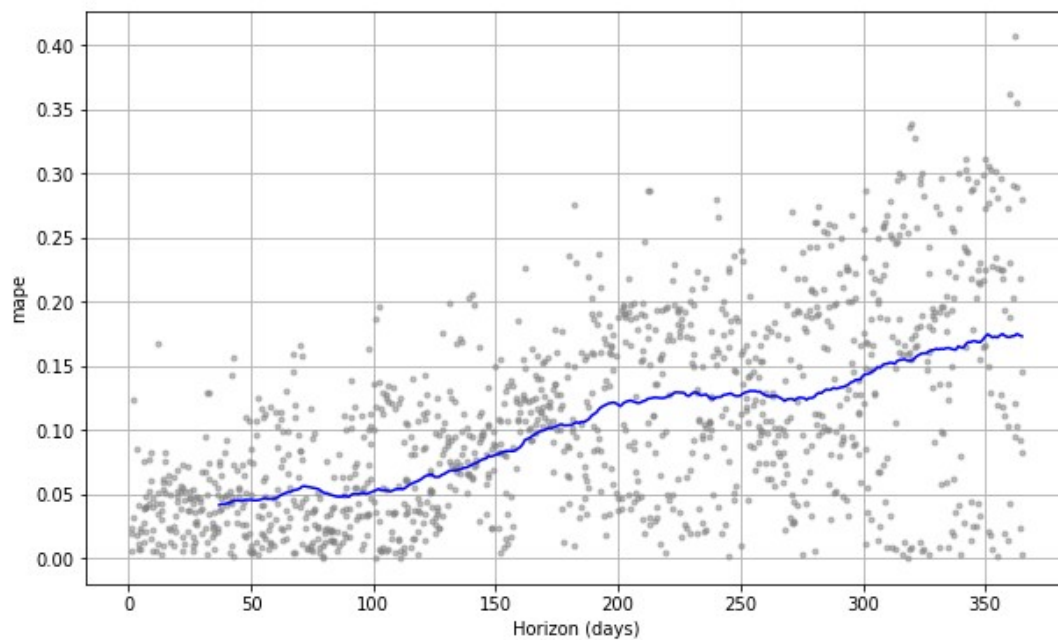


INFO:fbprophet:Disabling daily seasonality. Run prophet with daily_seasonality=True to override this.

Changepoint Prior Scale: 0.5
Number Changepoints: 10

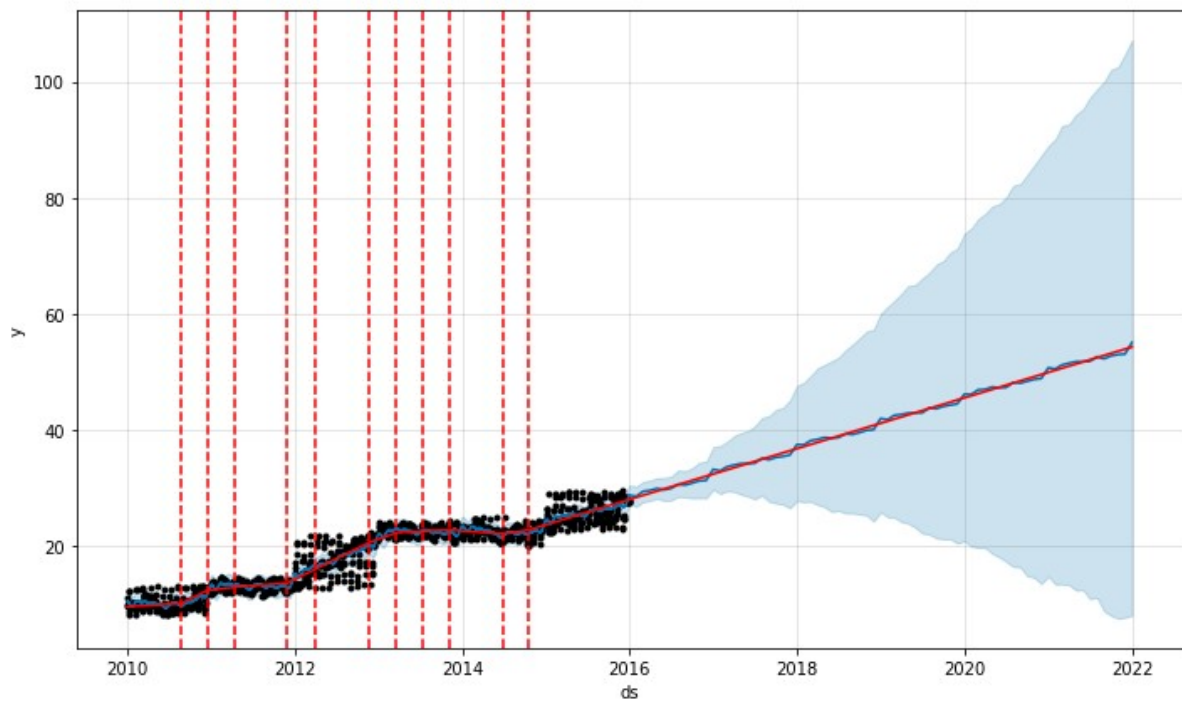


INFO:fbprophet:Making 5 forecasts with cutoffs between 2013-01-10 00:00:00 and 2014-12-31 00:00:00

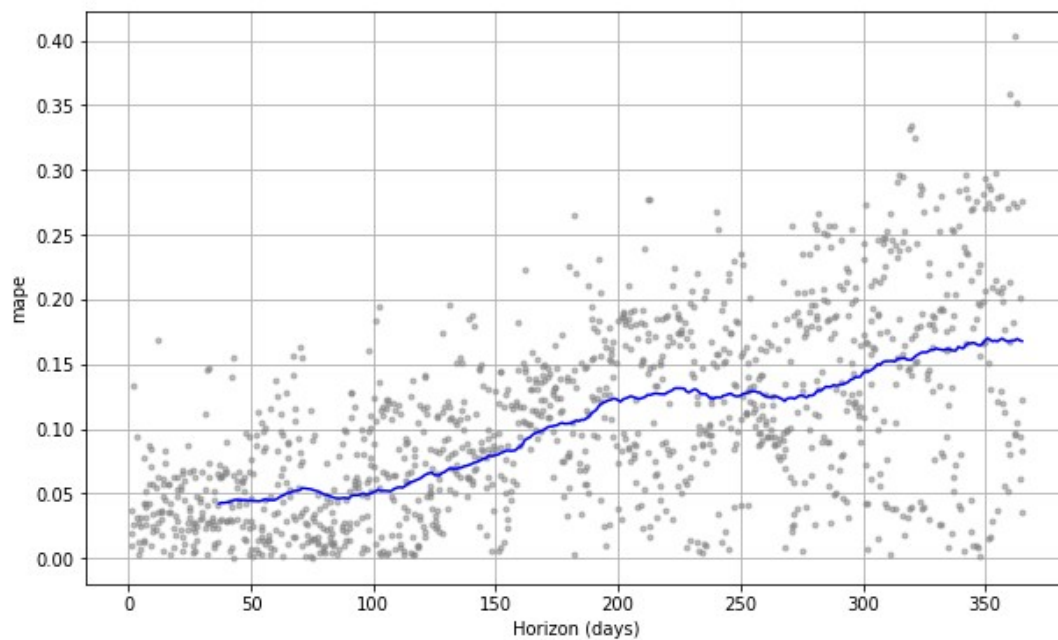


INFO:fbprophet:Disabling daily seasonality. Run prophet with daily_seasonality=True to override this.

Changepoint Prior Scale: 0.5
Number Changepoints: 15

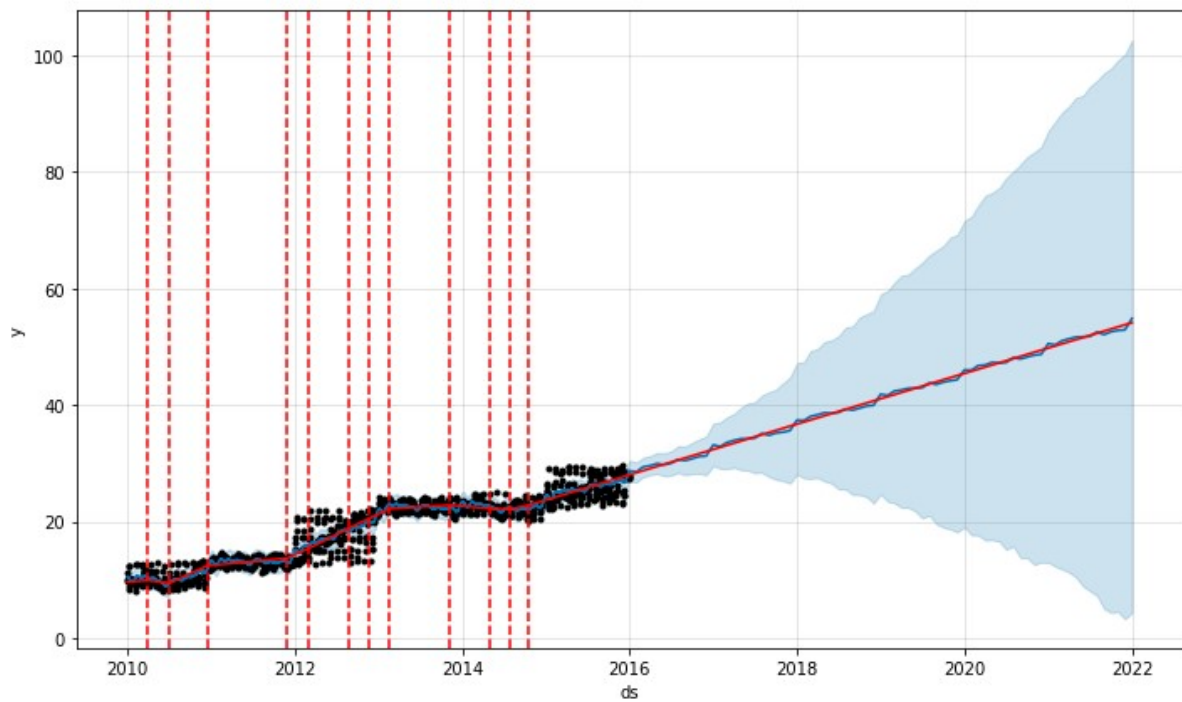


INFO:fbprophet:Making 5 forecasts with cutoffs between 2013-01-10 00:00:00 and 2014-12-31 00:00:00

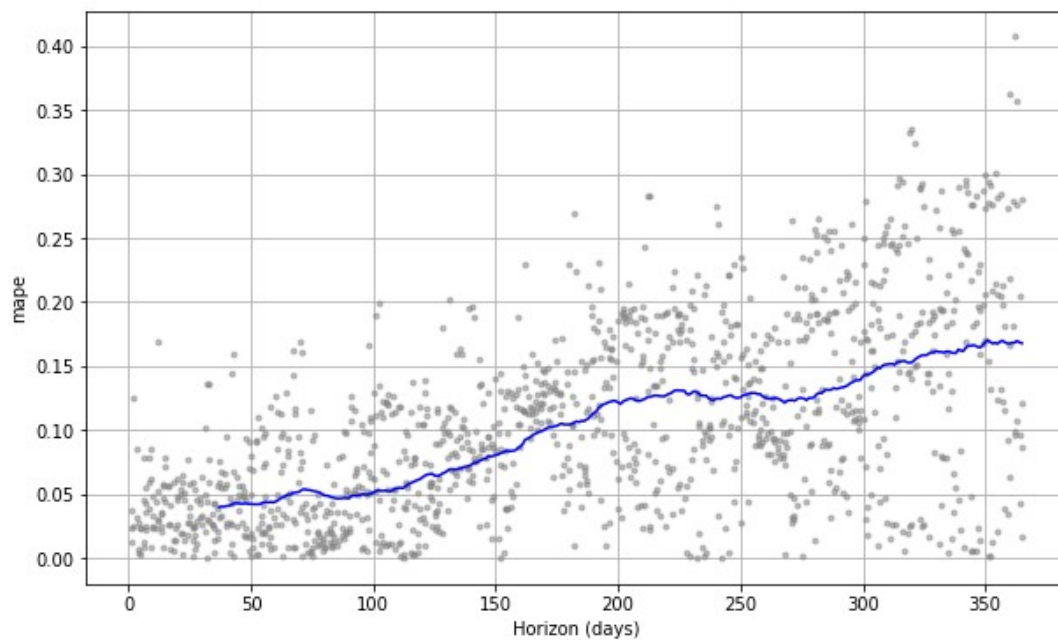


INFO:fbprophet:Disabling daily seasonality. Run prophet with daily_seasonality=True to override this.

Changepoint Prior Scale: 0.5
Number Changepoints: 20

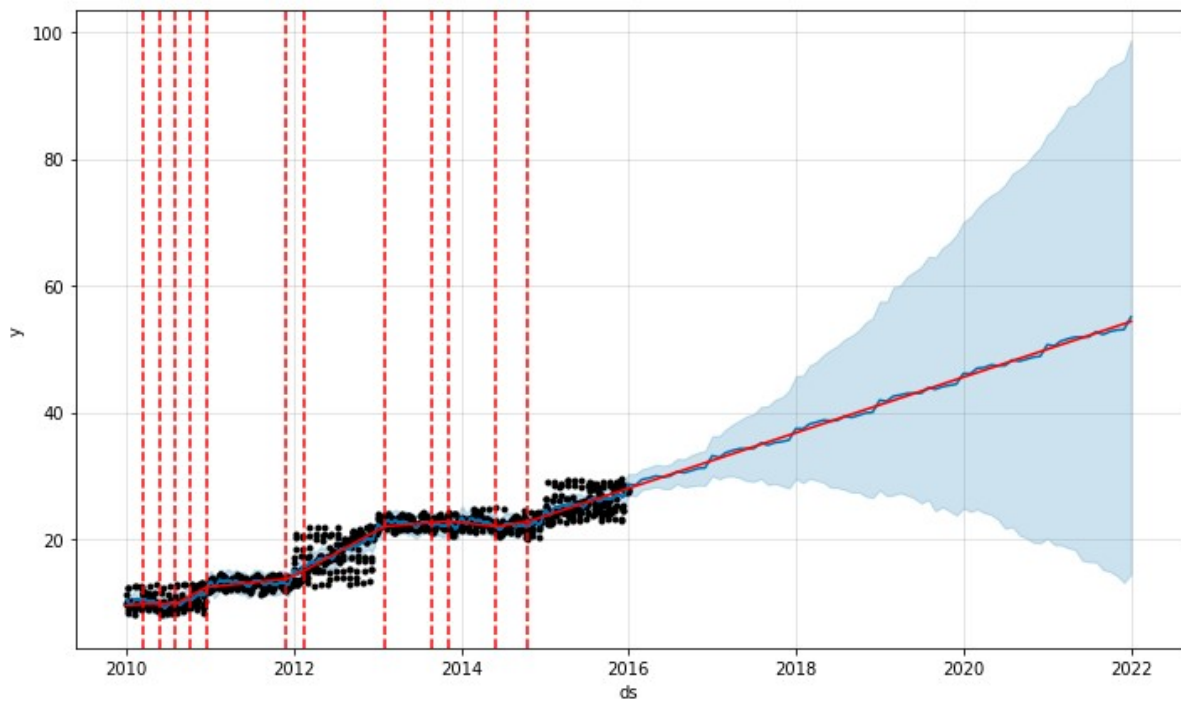


INFO:fbprophet:Making 5 forecasts with cutoffs between 2013-01-10 00:00:00 and 2014-12-31 00:00:00

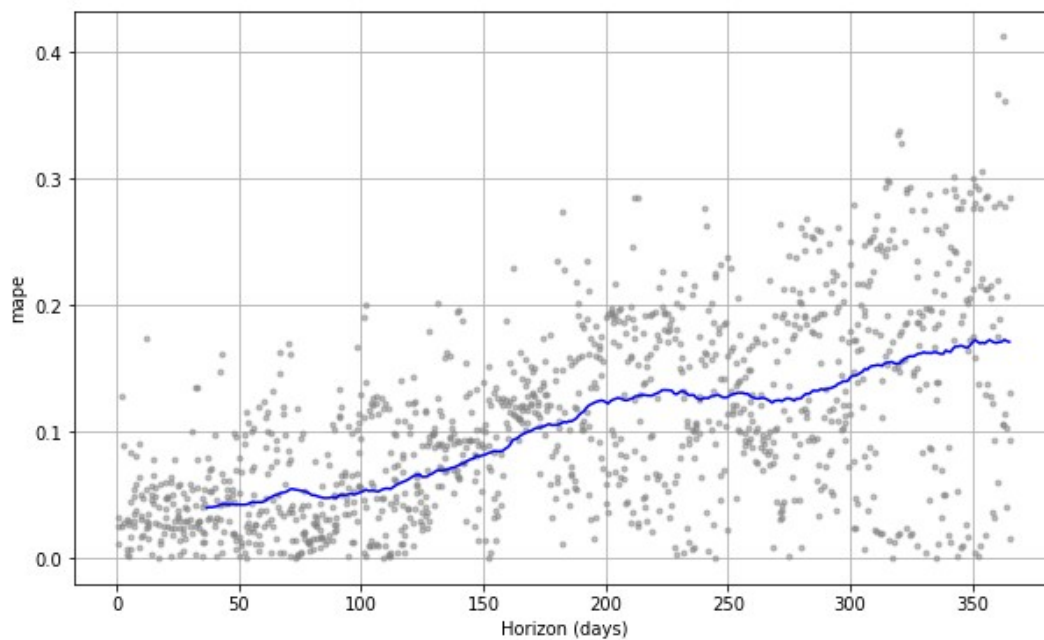


INFO:fbprophet:Disabling daily seasonality. Run prophet with daily_seasonality=True to override this.

Changepoint Prior Scale: 0.5
Number Changepoints: 25

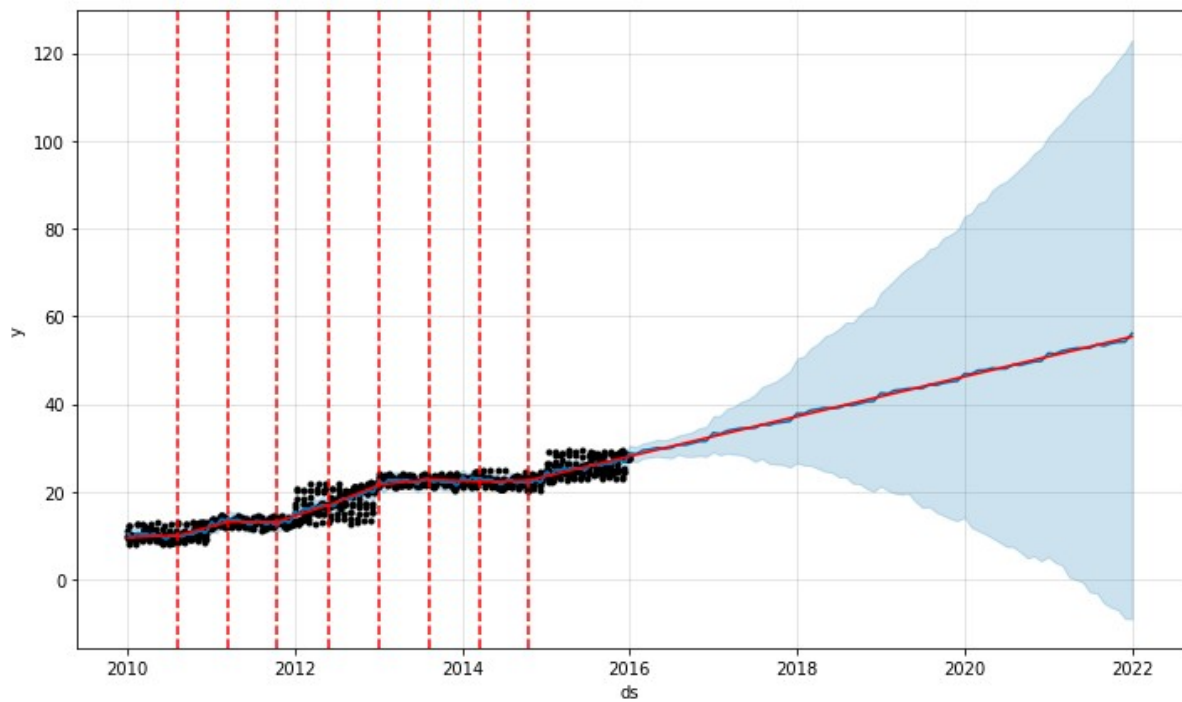


INFO:fbprophet:Making 5 forecasts with cutoffs between 2013-01-10 00:00:00 and 2014-12-31 00:00:00

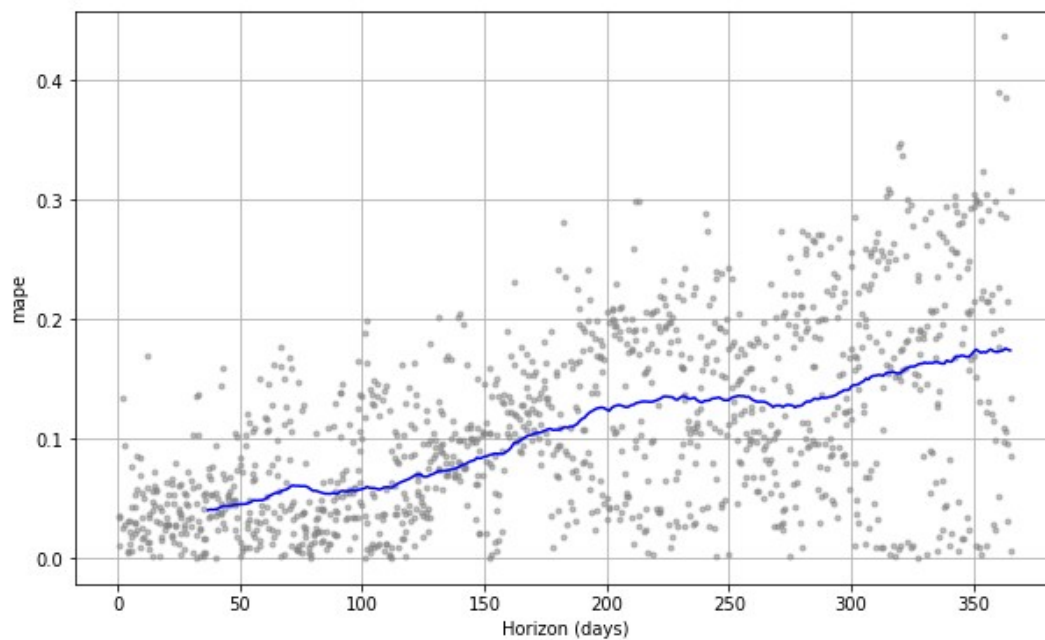


INFO:fbprophet:Disabling daily seasonality. Run prophet with daily_seasonality=True to override this.

Changepoint Prior Scale: 2
Number Changepoints: 8

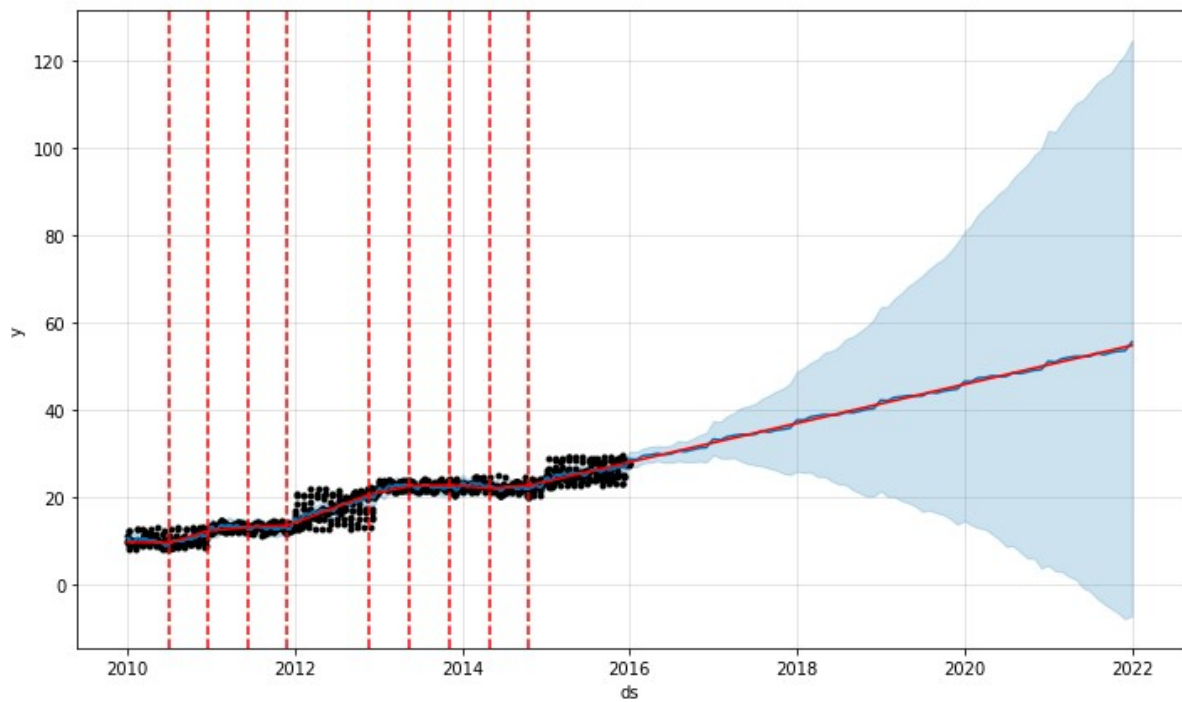


INFO:fbprophet:Making 5 forecasts with cutoffs between 2013-01-10 00:00:00 and 2014-12-31 00:00:00

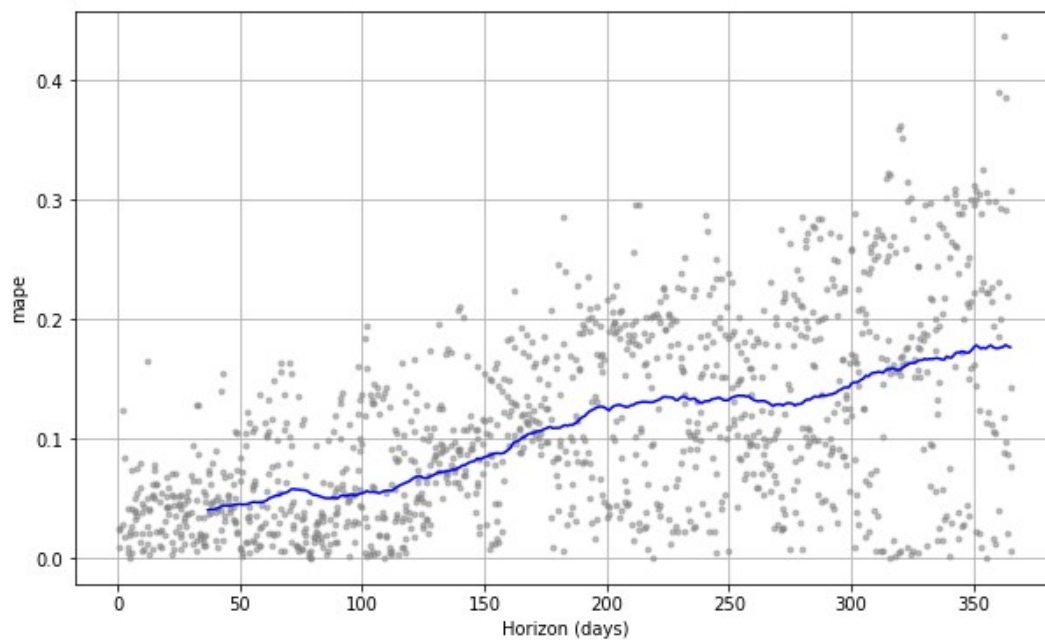


INFO:fbprophet:Disabling daily seasonality. Run prophet with daily_seasonality=True to override this.

Changepoint Prior Scale: 2
Number Changepoints: 10

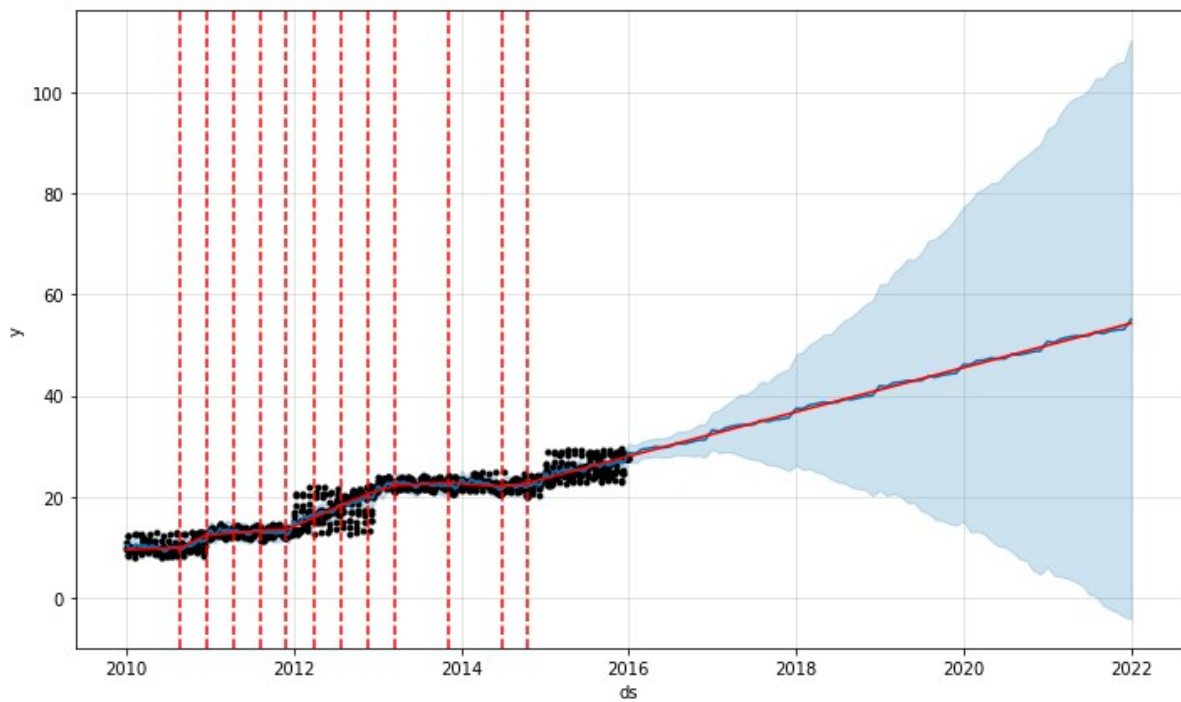


INFO:fbprophet:Making 5 forecasts with cutoffs between 2013-01-10 00:00:00 and 2014-12-31 00:00:00

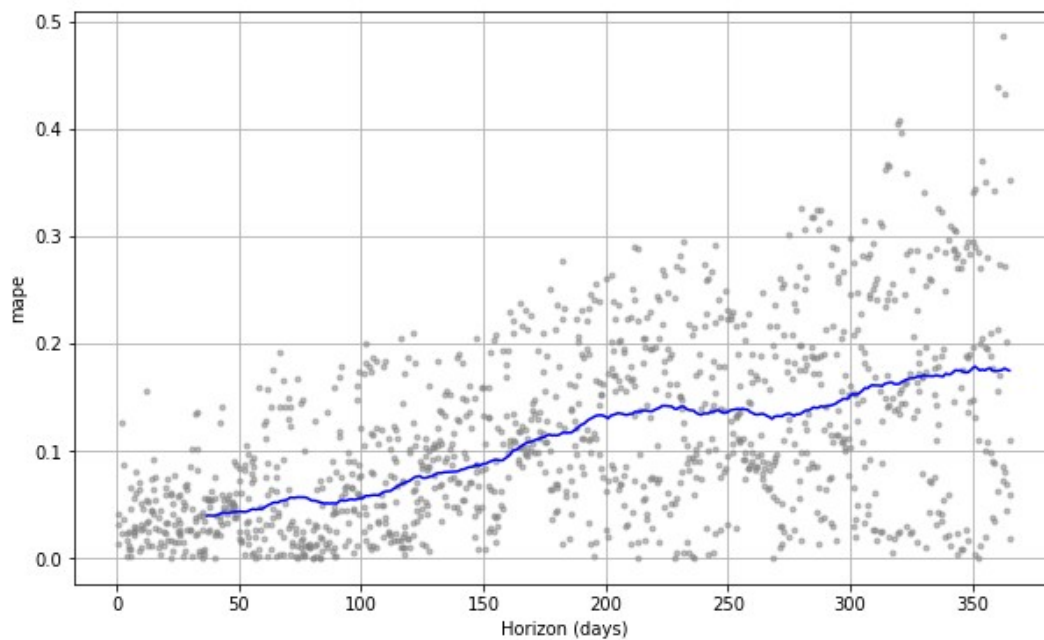


INFO:fbprophet:Disabling daily seasonality. Run prophet with daily_seasonality=True to override this.

Changepoint Prior Scale: 2
Number Changepoints: 15

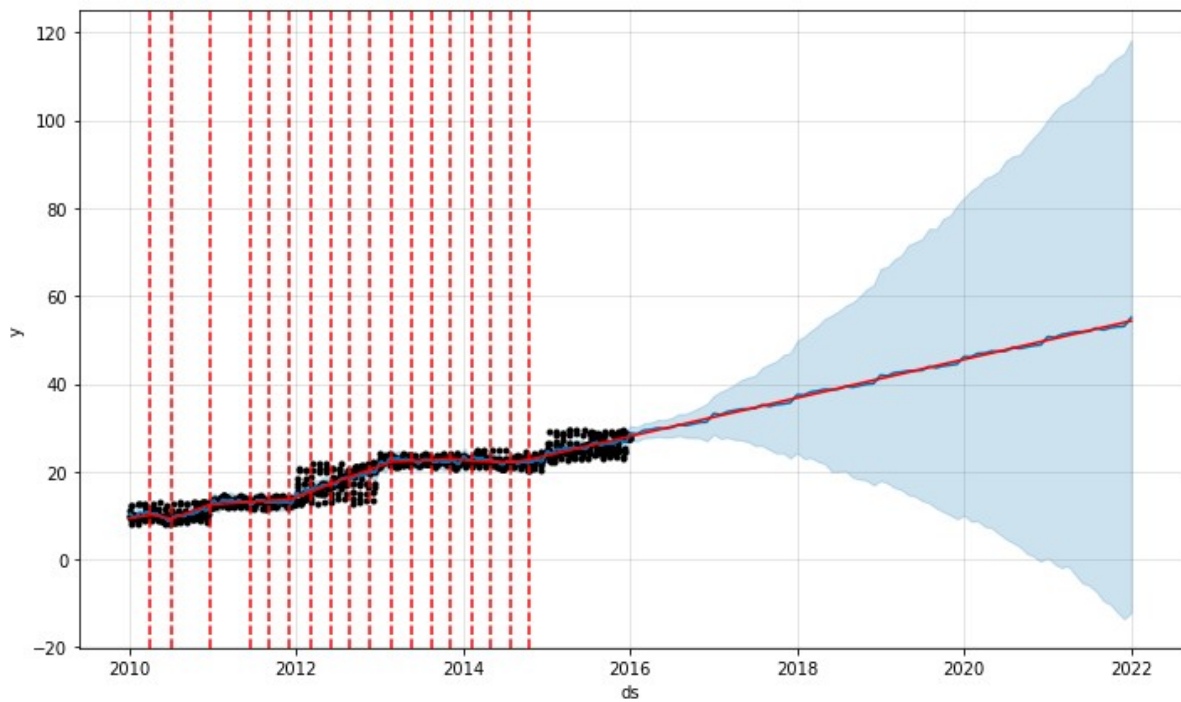


INFO:fbprophet:Making 5 forecasts with cutoffs between 2013-01-10 00:00:00 and 2014-12-31 00:00:00

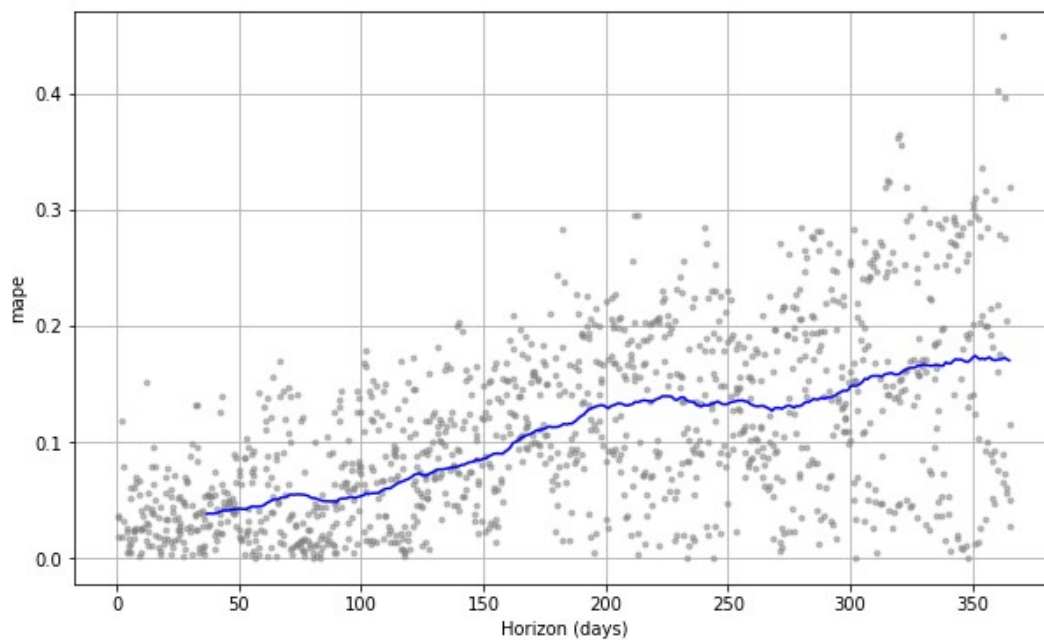


INFO:fbprophet:Disabling daily seasonality. Run prophet with daily_seasonality=True to override this.

Changepoint Prior Scale: 2
Number Changepoints: 20

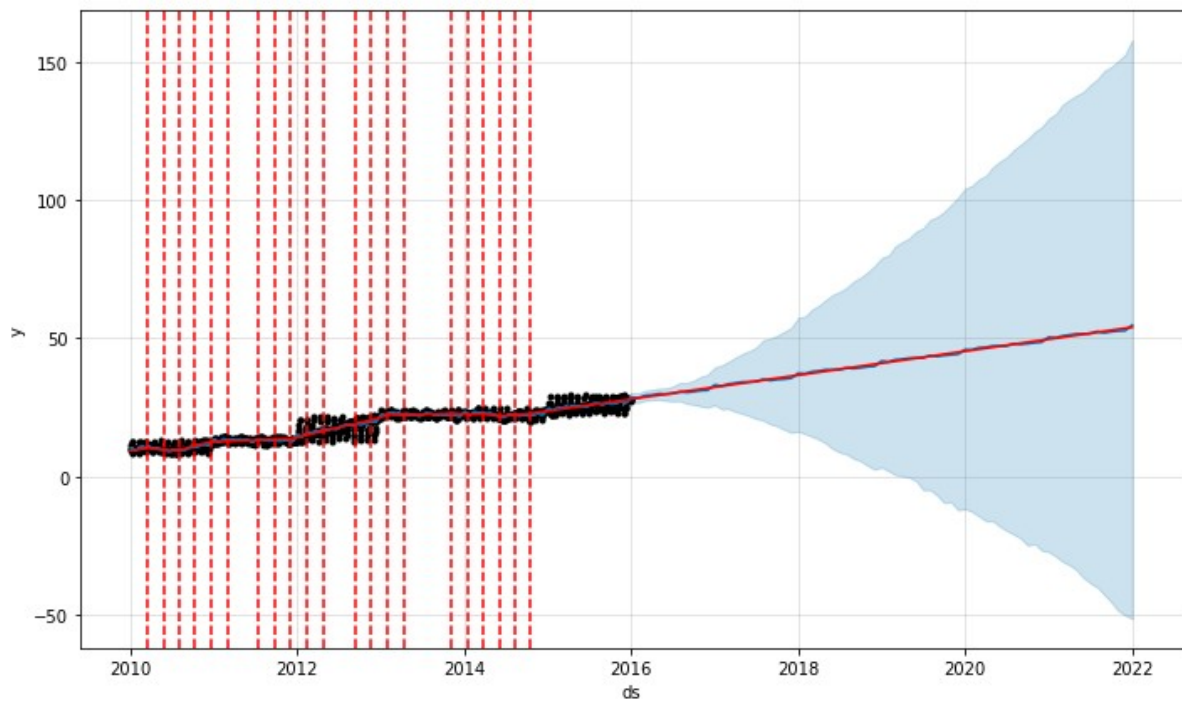


INFO:fbprophet:Making 5 forecasts with cutoffs between 2013-01-10 00:00:00 and 2014-12-31 00:00:00

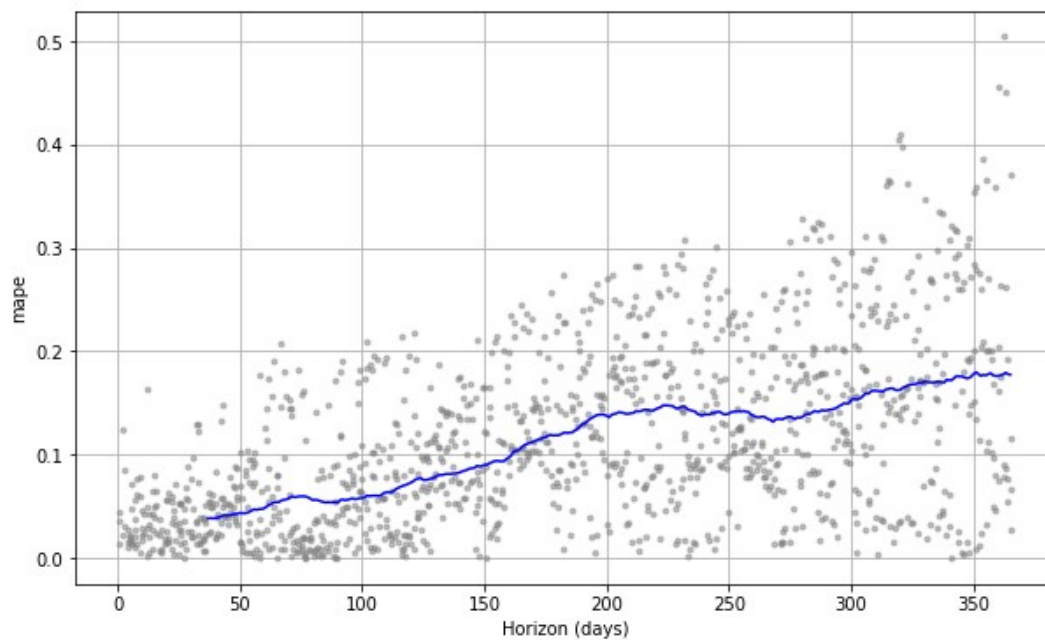


INFO:fbprophet:Disabling daily seasonality. Run prophet with daily_seasonality=True to override this.

Changepoint Prior Scale: 2
Number Changepoints: 25



INFO:fbprophet:Making 5 forecasts with cutoffs between 2013-01-10 00:00:00 and 2014-12-31 00:00:00



```
In [32]: #printing out the new rmse value for the predicted dataset
```

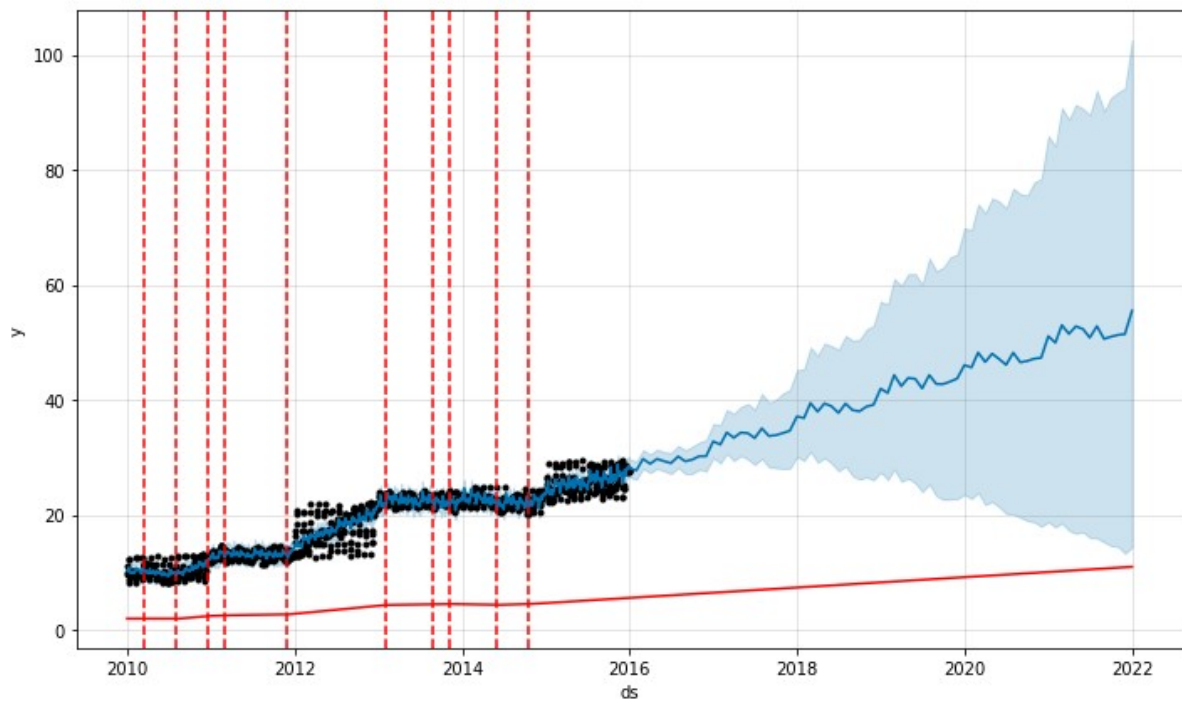
In [33]: rmse

```
Out[33]: [(3.068702694617269, {'changepoint_prior_scale': 0.005, 'n_changepoint': 8}),
(3.0507367877836433, {'changepoint_prior_scale': 0.005, 'n_changepoint': 10}),
(3.0331452777907466, {'changepoint_prior_scale': 0.005, 'n_changepoint': 15}),
(3.0481520647392895, {'changepoint_prior_scale': 0.005, 'n_changepoint': 20}),
(3.0482246272291236, {'changepoint_prior_scale': 0.005, 'n_changepoint': 25}),
(2.9225991076969926, {'changepoint_prior_scale': 0.05, 'n_changepoint': 8}),
(3.01099104082029, {'changepoint_prior_scale': 0.05, 'n_changepoint': 10}),
(2.9860570536328055, {'changepoint_prior_scale': 0.05, 'n_changepoint': 15}),
(2.9868138888541123, {'changepoint_prior_scale': 0.05, 'n_changepoint': 20}),
(3.037569300085768, {'changepoint_prior_scale': 0.05, 'n_changepoint': 25}),
(2.8676603571696635, {'changepoint_prior_scale': 0.5, 'n_changepoint': 8}),
(2.847959491822904, {'changepoint_prior_scale': 0.5, 'n_changepoint': 10}),
(2.797869214487519, {'changepoint_prior_scale': 0.5, 'n_changepoint': 15}),
(2.809767299123482, {'changepoint_prior_scale': 0.5, 'n_changepoint': 20}),
(2.8466691378368303, {'changepoint_prior_scale': 0.5, 'n_changepoint': 25}),
(2.9421267421194326, {'changepoint_prior_scale': 2, 'n_changepoint': 8}),
(2.9498260130804845, {'changepoint_prior_scale': 2, 'n_changepoint': 10}),
(3.0469397535470586, {'changepoint_prior_scale': 2, 'n_changepoint': 15}),
(2.9135648658888105, {'changepoint_prior_scale': 2, 'n_changepoint': 20}),
(3.1005066618726094, {'changepoint_prior_scale': 2, 'n_changepoint': 25})]
```

```
In [34]: #dataframe of annual US Public Holidays over training and forecasting periods
#creating a dataset of holidays so as to reduce the anomalies
ny = pd.DataFrame({'holiday': 'New Year's Day', 'ds' : pd.to_datetime(['2016-01-01',
    '2017-01-01'])})
mlk = pd.DataFrame({'holiday': 'Birthday of Martin Luther King, Jr.', 'ds' : pd.to_datetime(['2016-01-18', '2017-01-16'])})
wash = pd.DataFrame({'holiday': 'Washington's Birthday', 'ds' : pd.to_datetime(['2016-02-15', '2017-02-20'])})
mem = pd.DataFrame({'holiday': 'Memorial Day', 'ds' : pd.to_datetime(['2016-05-30', '2017-05-29'])})
ind = pd.DataFrame({'holiday': 'Independence Day', 'ds' : pd.to_datetime(['2015-07-04', '2016-07-04', '2017-07-04'])})
lab = pd.DataFrame({'holiday': 'Labor Day', 'ds' : pd.to_datetime(['2015-09-07', '2016-09-05', '2017-09-04'])})
col = pd.DataFrame({'holiday': 'Columbus Day', 'ds' : pd.to_datetime(['2015-10-12', '2016-10-10', '2017-10-09'])})
vet = pd.DataFrame({'holiday': 'Veteran's Day', 'ds' : pd.to_datetime(['2015-11-11', '2016-11-11', '2017-11-11'])})
thanks = pd.DataFrame({'holiday': 'Thanksgiving Day', 'ds' : pd.to_datetime(['2015-11-26', '2016-11-24'])})
christ = pd.DataFrame({'holiday': 'Christmas', 'ds' : pd.to_datetime(['2015-12-25', '2016-12-25'])})

holidays = pd.concat([ny, mlk, wash, mem, ind, lab, col, vet, thanks, christ])
```

```
In [35]: #defining the growth, seasonalities, changepoint prior scales, holidays
prophet = Prophet(growth='linear',
                  yearly_seasonality=True,
                  weekly_seasonality=True,
                  daily_seasonality=True,
                  holidays=holidays,
                  seasonality_mode='multiplicative',
                  seasonality_prior_scale=10,
                  holidays_prior_scale=10,
                  changepoint_prior_scale=.05,
                  mcmc_samples=0
                  ).add_seasonality(name='quarterly',
                                   period=365.25 / 4, fourier_order=15
                                   )
prophet.fit(df) # fits the model to the algorithm
future = prophet.make_future_dataframe(periods=12 * 6, freq='M') #defining the period of prediction
forecast = prophet.predict(future) #forecasting
fig = prophet.plot(forecast) #creating a figure of forecast
a = add_changepoints_to_plot(fig.gca(), prophet, forecast)
plt.show()
df_cv = cross_validation(prophet, initial='1095 days', period='180 days', horizon =
'365 days') #cross validating the data
df_p = performance_metrics(df_cv) #adding the performance metrics to the plot
fig = plot_cross_validation_metric(df_cv, metric='mape') #plotting the cross validation performance metric by mape
plt.show()
```



INFO:fbprophet:Making 5 forecasts with cutoffs between 2013-01-10 00:00:00 and 2014-12-31 00:00:00

