* [Cross validation](http://facebook.github.io/prophet/docs/diagnostics.html" \l "cross-validation)
* [Parallelizing cross validation](http://facebook.github.io/prophet/docs/diagnostics.html#parallelizing-cross-validation)
* [Hyperparameter tuning](http://facebook.github.io/prophet/docs/diagnostics.html#hyperparameter-tuning)
* [Handling Shocks](http://facebook.github.io/prophet/docs/handling_shocks.html)
  + [Treating COVID-19 lockdowns as a one-off holidays](http://facebook.github.io/prophet/docs/handling_shocks.html#treating-covid-19-lockdowns-as-a-one-off-holidays)
  + [Sense checking the trend](http://facebook.github.io/prophet/docs/handling_shocks.html#sense-checking-the-trend)
  + [Changes in seasonality between pre- and post-COVID](http://facebook.github.io/prophet/docs/handling_shocks.html#changes-in-seasonality-between-pre--and-post-covid)
* [Additional Topics](http://facebook.github.io/prophet/docs/additional_topics.html)
  + [Saving models](http://facebook.github.io/prophet/docs/additional_topics.html#saving-models)
  + [Flat trend and custom trends](http://facebook.github.io/prophet/docs/additional_topics.html#flat-trend-and-custom-trends)
  + [Updating fitted models](http://facebook.github.io/prophet/docs/additional_topics.html#updating-fitted-models)
  + [External references](http://facebook.github.io/prophet/docs/additional_topics.html#external-references)
* [Getting Help and Contributing](http://facebook.github.io/prophet/docs/contributing.html)

Saturating Forecasts

# Forecasting Growth

By default, Prophet uses a linear model for its forecast. 默认情况下，Prophet使用线性模型(linear)对增长进行预测。

When forecasting growth, there is usually some maximum achievable point: total market size, total population size, etc.在预测增长时，通常是预测一些最大可实现点:总市场规模，总人口规模等。

This is called the carrying capacity, and the forecast should saturate at this point. 这种可实现点被称为承载力，预测结果在这一点上应该达到了最大限度（饱和）。

Prophet allows you to make forecasts using a [logistic growth](https://en.wikipedia.org/wiki/Logistic_function) trend model, with a specified carrying capacity. 以上这种情况下需要使用logistic growth趋势模型

We illustrate this with the log number of page visits to the [R (programming language)](https://en.wikipedia.org/wiki/R_%28programming_language%29) page on Wikipedia:

例子使用维基百科上“R语言”页面的访问量日志作为数据进行实验

1. # Python
2. df = pd.read\_csv('https://raw.githubusercontent.com/facebook/prophet/main/examples/example\_wp\_log\_R.csv')

We must specify the carrying capacity in a column **cap**. Here we will assume a particular value, but this would usually be set using data or expertise about the market size.

我们必须在列**cap**中指定承载能力。这里我们将假设一个特定的值，但这通常是根据关于市场规模的数据或专业知识来设置的。

1 # Python

2 df['cap'] = 8.5

**The important things to note are that cap must be specified for every row in the dataframe, and that it does not have to be constant. 需要注意的是，cap必须为了dataframe中的每一行指定，而且它不一定是数值型常量。**。

If the market size is growing, then cap can be an increasing sequence.

We then fit the model as before, except pass in an additional argument to specify logistic growth:

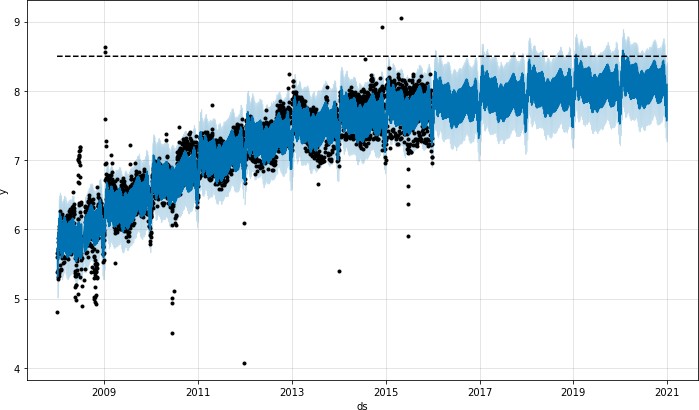
1. # Python
2. m = Prophet(growth='logistic') 3 m.fit(df)

We make a dataframe for future predictions as before, except we must also specify the capacity in the future. Here we keep capacity constant at the same value as in the history, and forecast 5 years into the future:

和之前一样，我们为未来的预测制作一个数据框，除了我们还必须指定未来的容量。在这里，我们将容量保持在历史值不变，并预测未来5年的情况:

1. # Python
2. future = m.make\_future\_dataframe(periods=1826) 3 future['cap'] = 8.5

4 fcst = m.predict(future) 5 fig = m.plot(fcst)



The logistic function has an implicit minimum of 0, and will saturate at 0 the same way that it saturates at the capacity. logistic函数有一个隐式的最小值0，并且在0处达到饱和，就像在容量处达到饱和一样。

It is possible to also specify a different saturating minimum. 也可以指定不同的饱和最小值。

# Saturating Minimum

The logistic growth model can also handle a saturating minimum, logistic model也可以处理一个最小饱和。(上例中处理的最大饱和)

which is specified with a column floor in the same way as the cap column specifies the maximum:

与之前指定最大饱和时使用 cap 相同，而指定最小饱和是使用 floor

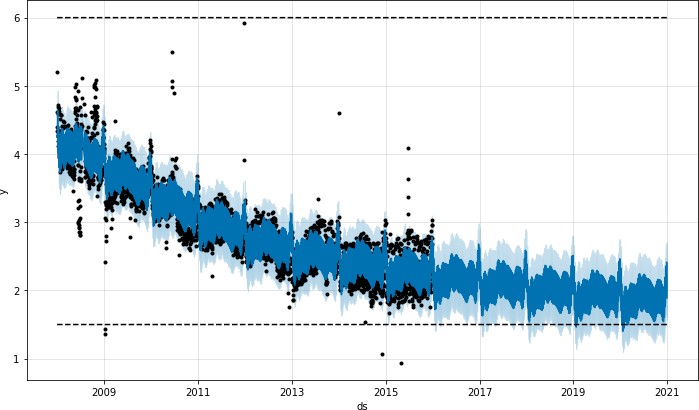
1 # Python

2 df['y'] = 10 - df['y']

3 df['cap'] = 6

4 df['floor'] = 1.5

1. future['cap'] = 6
2. future['floor'] = 1.5
3. m = Prophet(growth='logistic')
4. m.fit(df)
5. fcst = m.predict(future)
6. fig = m.plot(fcst)



To use a logistic growth trend with a saturating minimum, a maximum capacity must also be specified. [Edit on GitHub](https://github.com/facebook/prophet/blob/main/docs/_docs/saturating_forecasts.md)

**Facebook Open Source**

[Open Source Project](https://code.facebook.com/projects/)s [GitHu](https://github.com/facebook/)b [Twitte](https://twitter.com/fbOpenSource)r [Privacy](https://opensource.facebook.com/legal/privacy/) [Terms](https://opensource.facebook.com/legal/terms/) [Contribute to this project on GitHub](https://github.com/facebook/prophet)