

Setup for Notebooks

```
In [1]: from bokeh.io import output_notebook, show
output_notebook()
```

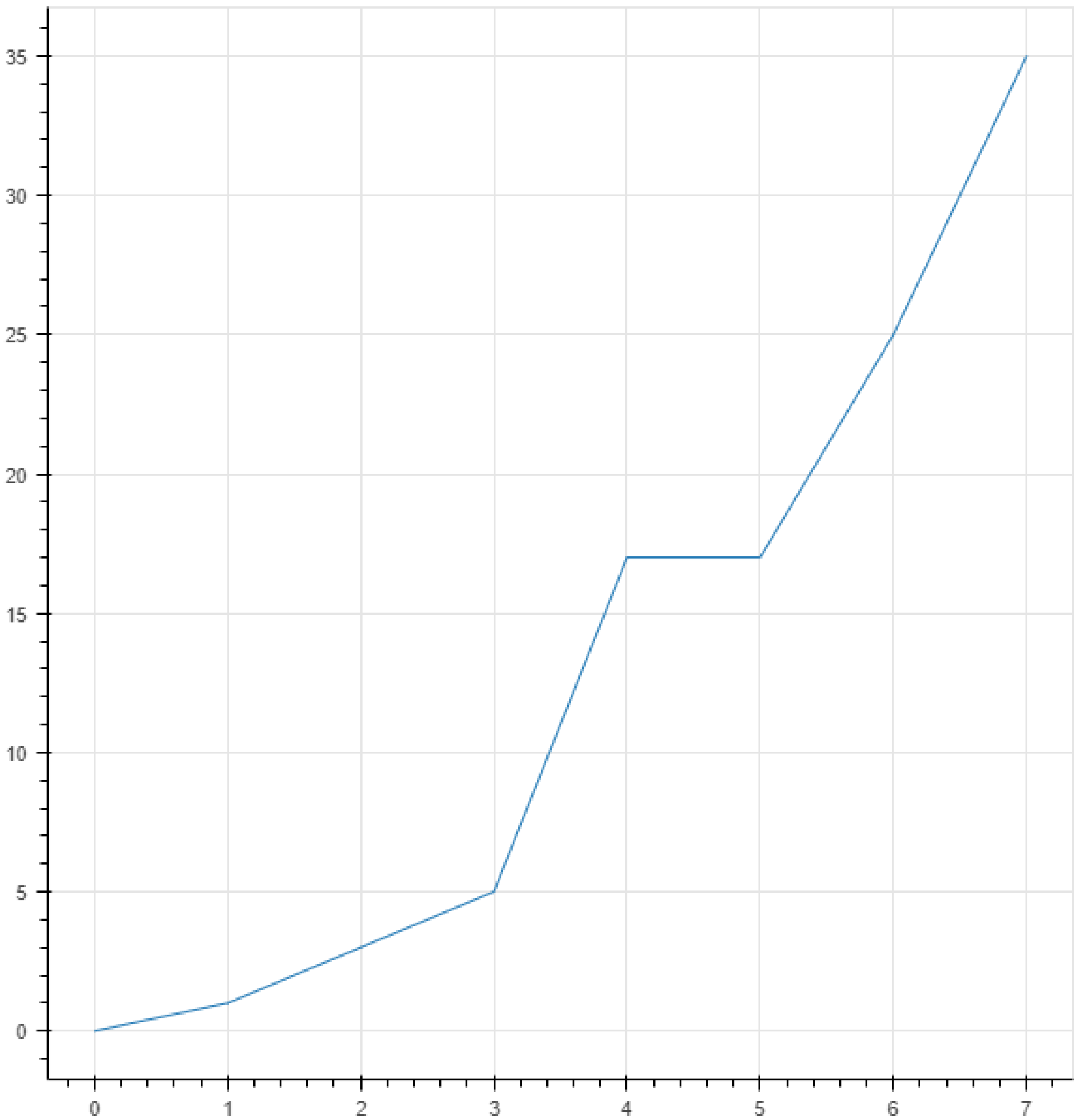
(<https://bokeh.pydata.org/>) successfully loaded.

```
In [2]: # We will just use that a lot in general
import numpy as np
```

My First Plot

```
In [3]: from bokeh.plotting import figure as bokeh_figure

figure = bokeh_figure()
ys = [0, 1, 3, 5, 17, 17, 25, 35]
xs = np.arange(len(ys))
figure.line(xs, ys)
show(figure)
```



(<https://bokeh.pydata.org/>)

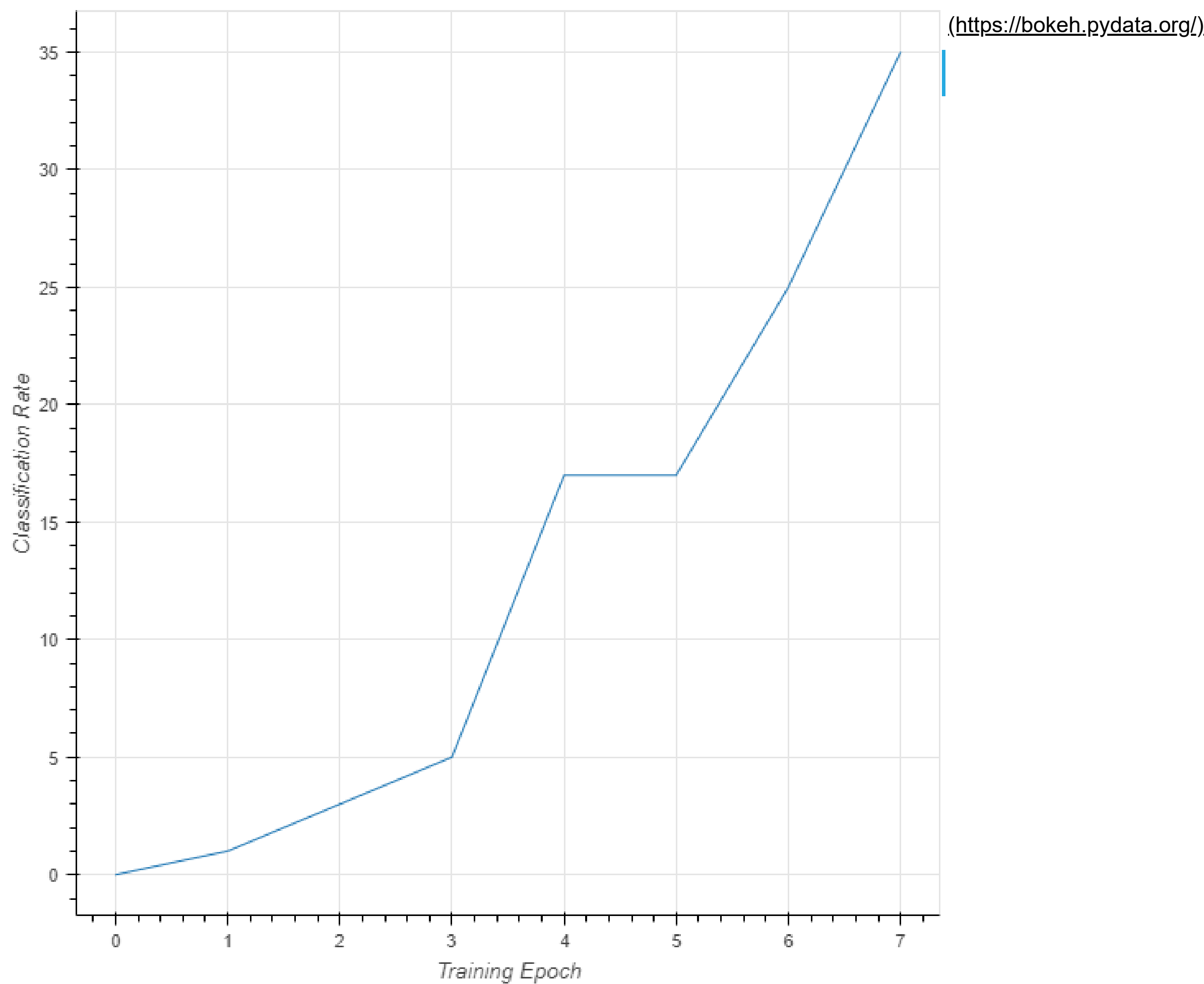
Clean Version

```
In [4]: # Bokeh is already clean \o/
```

Adding Labels

In [5]: `from bokeh.plotting import figure as bokeh_figure`

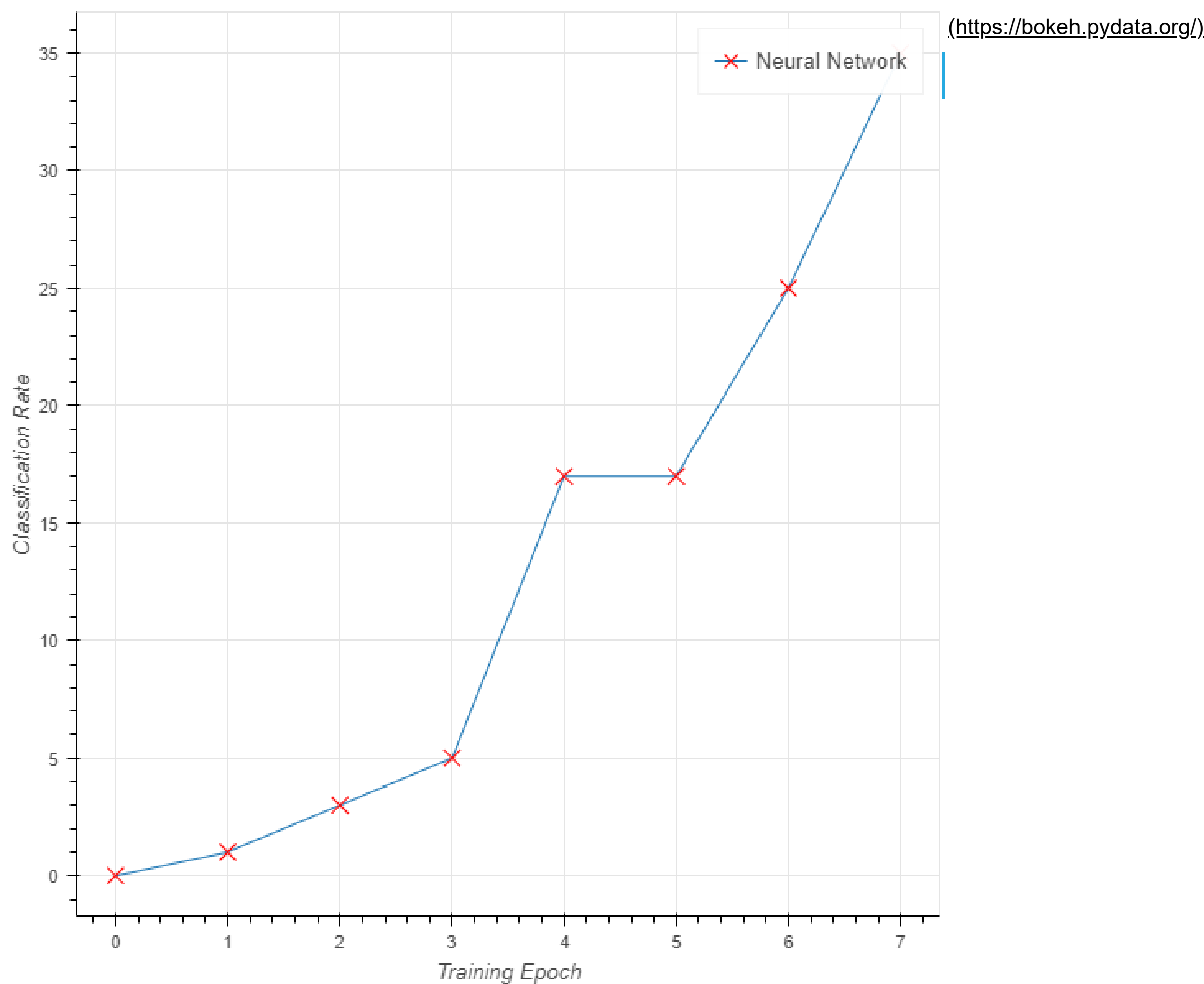
```
figure = bokeh_figure()
ys = [0, 1, 3, 5, 17, 17, 25, 35]
xs = np.arange(len(ys))
figure.line(xs, ys)
figure.xaxis.axis_label = "Training Epoch"
figure.yaxis.axis_label = "Classification Rate"
show(figure)
```



Adding a Legend

In [6]: `from bokeh.plotting import figure as bokeh_figure`
`from bokeh.models import Label, Arrow, NormalHead`

```
figure = bokeh_figure()
ys = [0, 1, 3, 5, 17, 17, 25, 35]
xs = np.arange(len(ys))
figure.line(xs, ys, legend="Neural Network")
figure.scatter(xs, ys, marker="x", size=10, color="red", legend="Neural Network")
figure.xaxis.axis_label = "Training Epoch"
figure.yaxis.axis_label = "Classification Rate"
show(figure)
```

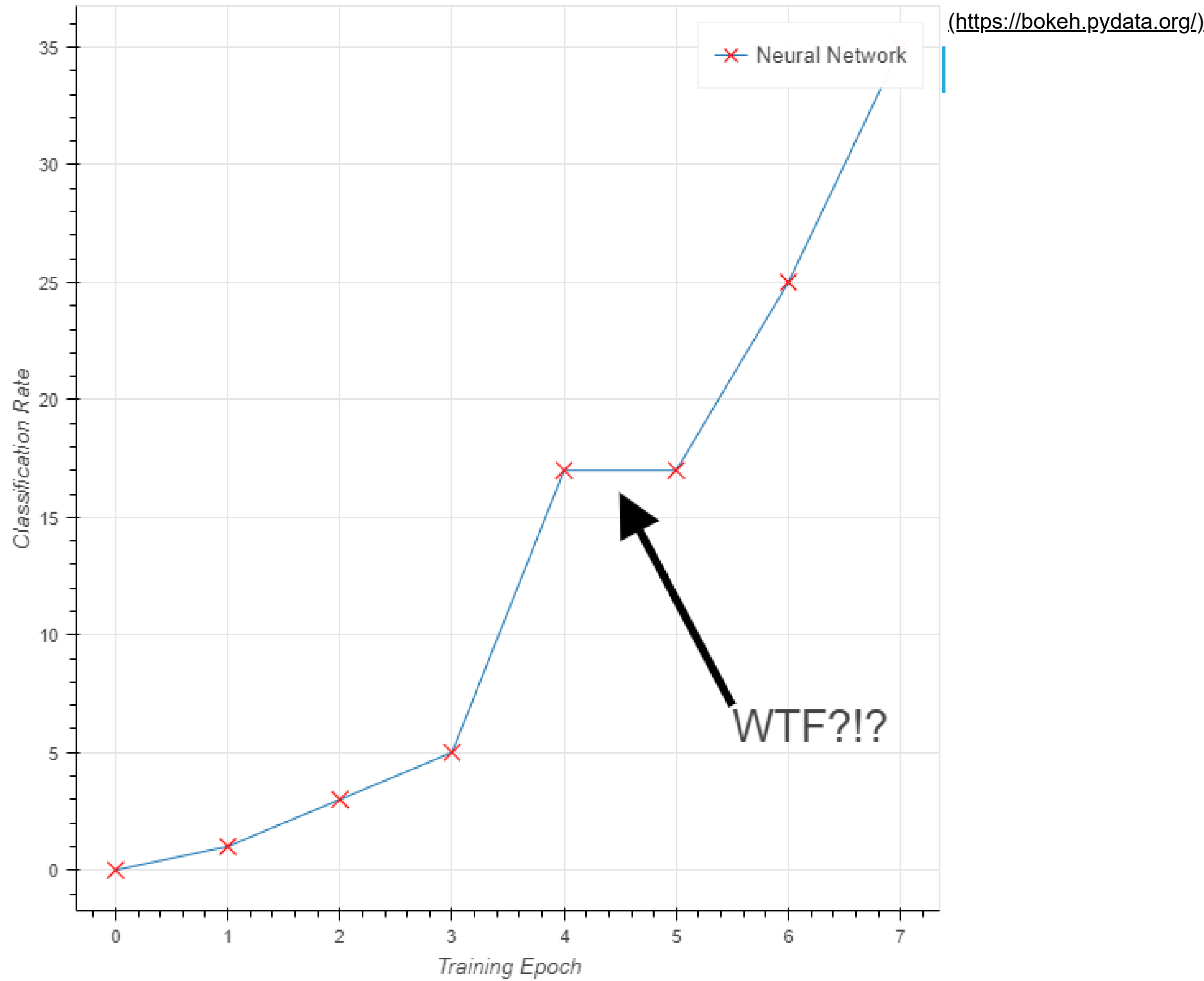


Adding Markers

In [7]:

```
from bokeh.plotting import figure as bokeh_figure
from bokeh.models import Label, Arrow, NormalHead

figure = bokeh_figure()
ys = [0, 1, 3, 5, 17, 17, 25, 35]
xs = np.arange(len(ys))
figure.line(xs, ys, legend="Neural Network")
figure.scatter(xs, ys, marker="x", size=10, color="red", legend="Neural Network")
figure.add_layout(Label(x=5.5, y=5, text="WTF?!?", text_font_size="20pt"))
figure.add_layout(Arrow(end=NormalHead(), x_start=5.5, y_start=7, x_end=4.5, y_end=16, line_width=5))
figure.xaxis.axis_label = "Training Epoch"
figure.yaxis.axis_label = "Classification Rate"
show(figure)
```



Multiple Lines

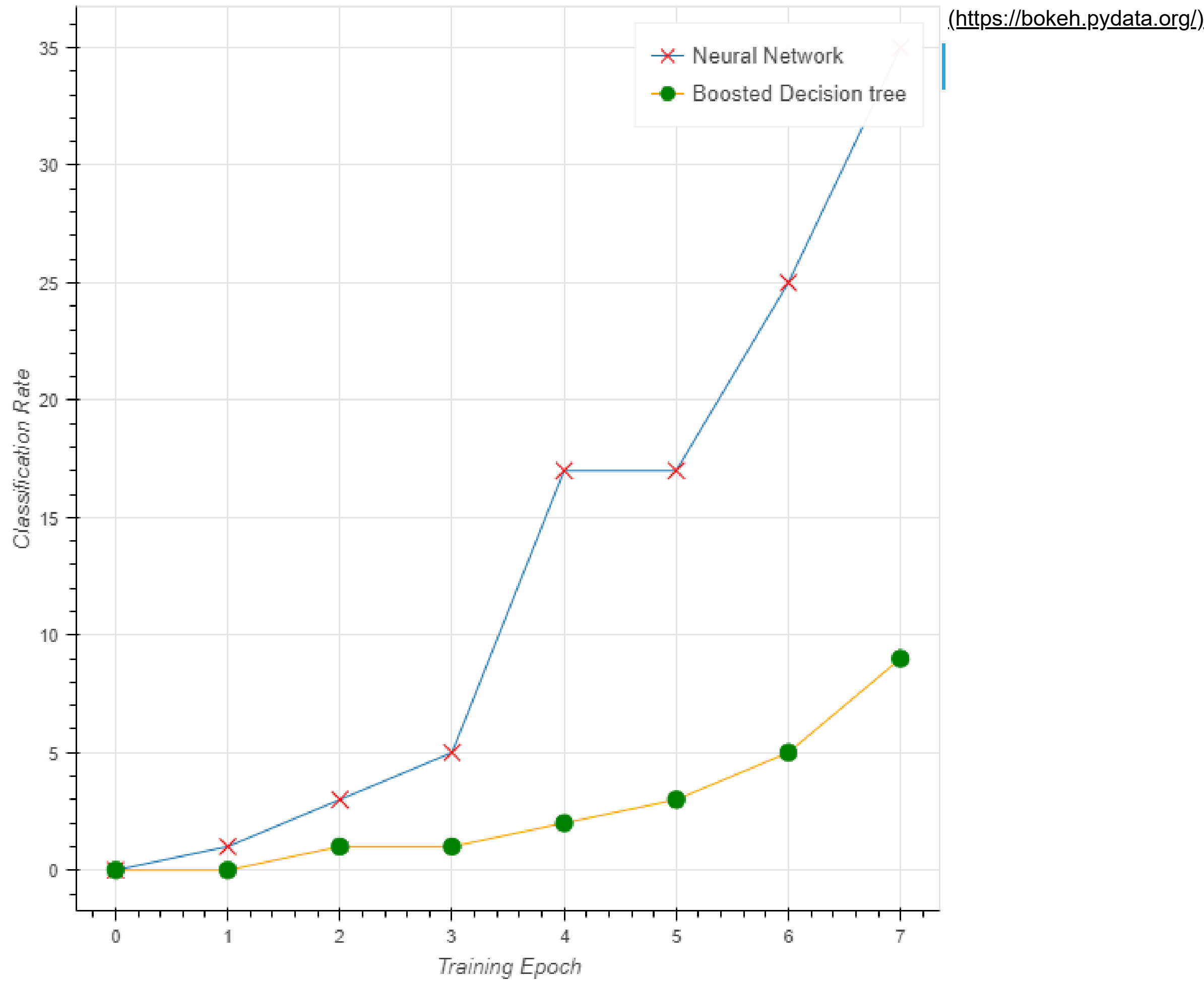
In [8]:

```
from bokeh.plotting import figure as bokeh_figure
from bokeh.models import Label, Arrow, NormalHead

figure = bokeh_figure()
ys = [0, 1, 3, 5, 17, 17, 25, 35]
xs = np.arange(len(ys))
figure.line(xs, ys, legend="Neural Network")
figure.scatter(xs, ys, marker="x", size=10, color="red", legend="Neural Network")

figure.line(xs, [0, 0, 1, 1, 2, 3, 5, 9], color="orange", legend="Boosted Decision tree")
figure.scatter(xs, [0, 0, 1, 1, 2, 3, 5, 9], marker="o", size=10, color="green", legend="Boosted Decision tree")

figure.xaxis.axis_label = "Training Epoch"
figure.yaxis.axis_label = "Classification Rate"
show(figure)
```



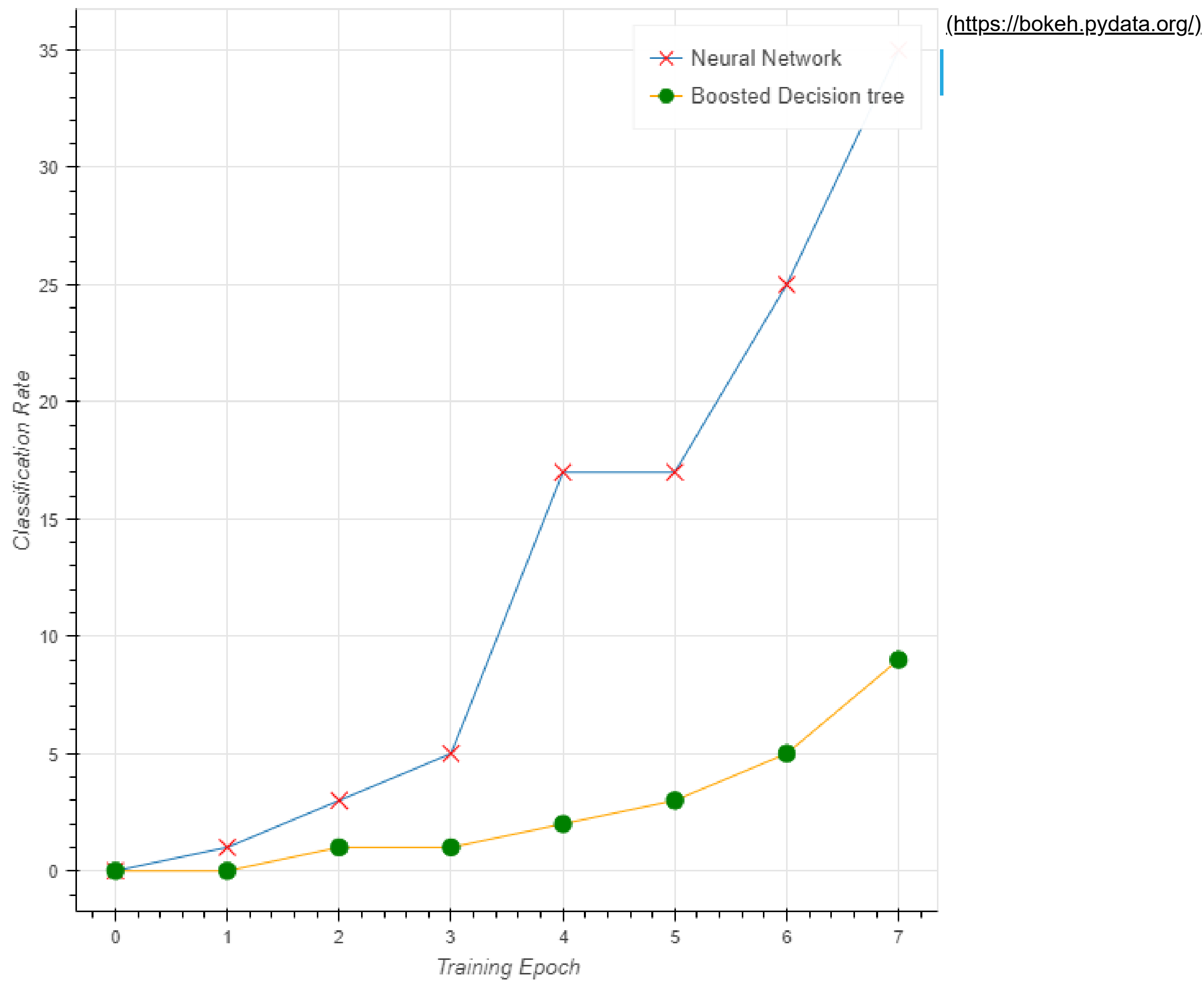
Saving Plots

```
In [9]: from bokeh.plotting import figure as bokeh_figure
from bokeh.models import Label, Arrow, NormalHead

figure = bokeh_figure()
ys = [0, 1, 3, 5, 17, 17, 25, 35]
xs = np.arange(len(ys))
figure.line(xs, ys, legend="Neural Network")
figure.scatter(xs, ys, marker="x", size=10, color="red", legend="Neural Network")

figure.line(xs, [0, 0, 1, 1, 2, 3, 5, 9], color="orange", legend="Boosted Decision tree")
figure.scatter(xs, [0, 0, 1, 1, 2, 3, 5, 9], marker="o", size=10, color="green", legend="Boosted Decision tree")

figure.xaxis.axis_label = "Training Epoch"
figure.yaxis.axis_label = "Classification Rate"
show(figure)
```



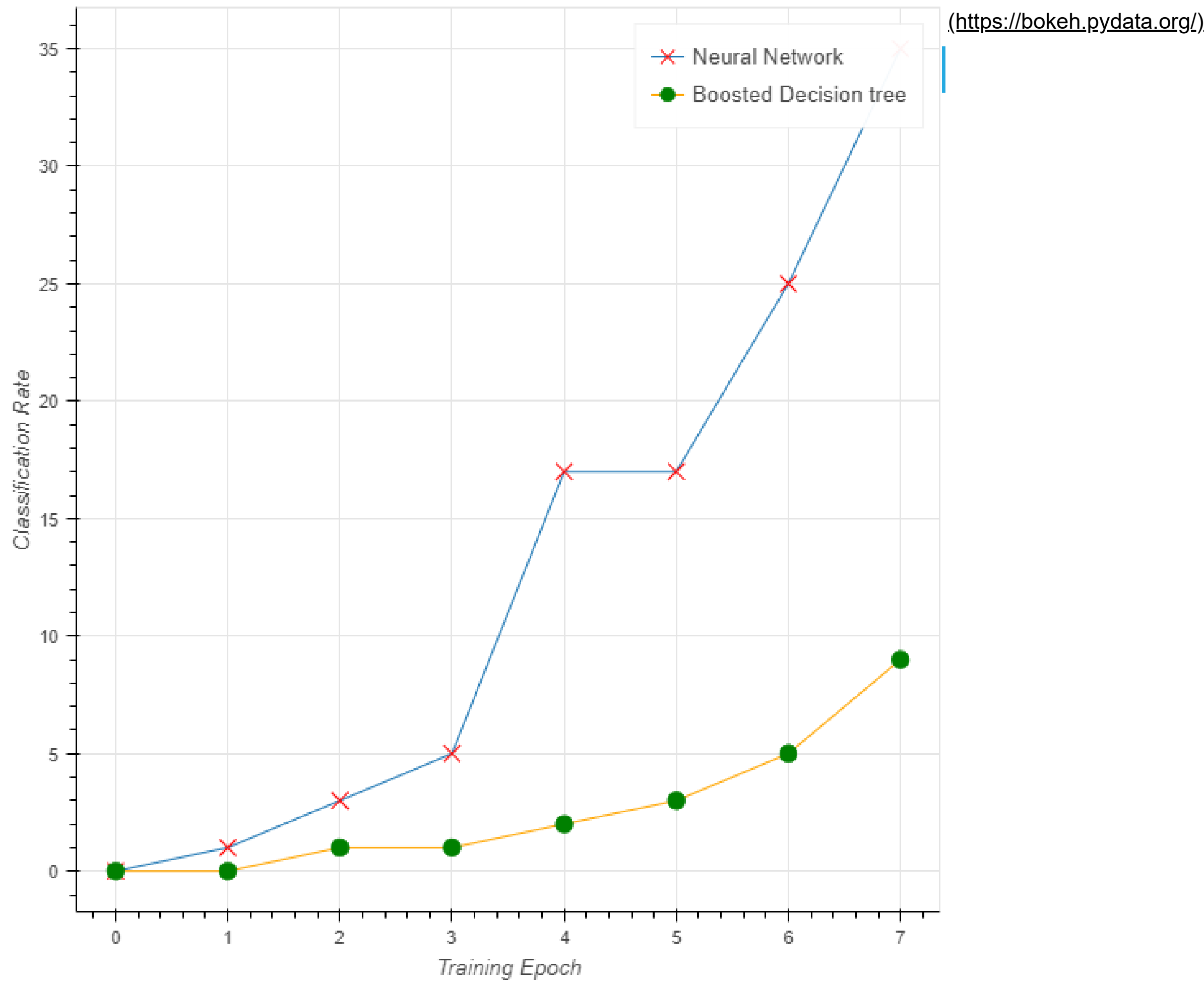
```
In [10]: from bokeh.plotting import output_file, save

from bokeh.plotting import figure as bokeh_figure
from bokeh.models import Label, Arrow, NormalHead

figure = bokeh_figure()
# figure.output_backend = "svg" # use if you want the save option to store a svg
output_file("rate_over_epochs.html")
ys = [0, 1, 3, 5, 17, 17, 25, 35]
xs = np.arange(len(ys))
figure.line(xs, ys, legend="Neural Network")
figure.scatter(xs, ys, marker="x", size=10, color="red", legend="Neural Network")
figure.line(xs, [0, 0, 1, 1, 2, 3, 5, 9], color="orange", legend="Boosted Decision tree")
figure.scatter(xs, [0, 0, 1, 1, 2, 3, 5, 9], marker="o", size=10, color="green", legend="Boosted Decision tree")

figure.xaxis.axis_label = "Training Epoch"
figure.yaxis.axis_label = "Classification Rate"
show(figure)

save(figure)
```



```
Out[10]: '/home/jovyan/rate_over_epochs.html'
```

Other Plot Types

They work slightly different in Bokeh so we omit that for now

Subplots

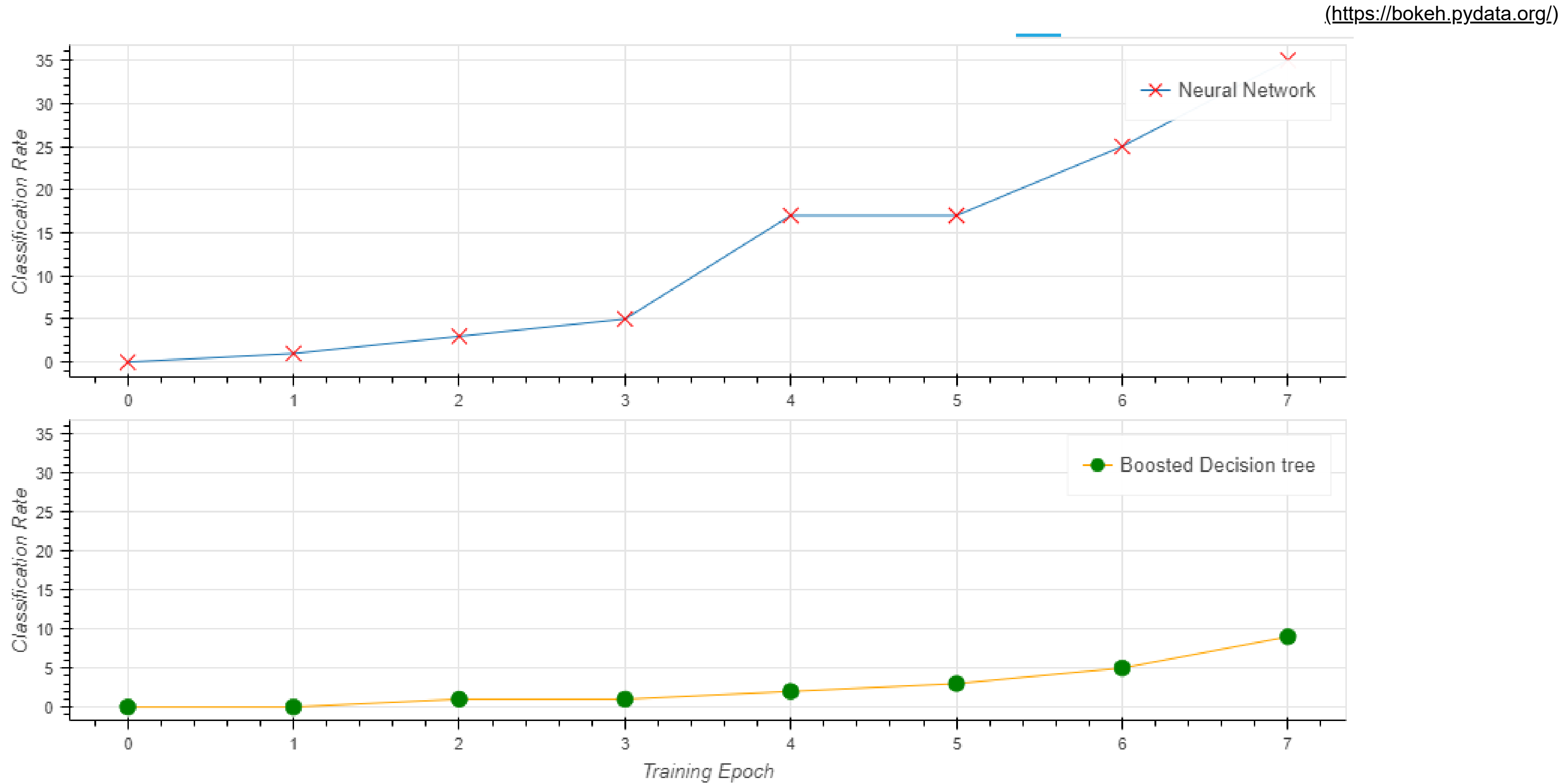
```
In [11]: from bokeh.layouts import gridplot

figure_1 = bokeh_figure(plot_width=900, plot_height=250)
ys = [0, 1, 3, 5, 17, 17, 25, 35]
xs = np.arange(len(ys))
figure_1.line(xs, ys, legend="Neural Network")
figure_1.scatter(xs, ys, marker="x", size=10, color="red", legend="Neural Network")
figure_1.yaxis.axis_label = "Classification Rate"

figure_2 = bokeh_figure(plot_width=900, plot_height=250, x_range=figure_1.x_range, y_range=figure_1.y_range)
figure_2.line(xs, [0, 0, 1, 1, 2, 3, 5, 9], color="orange", legend="Boosted Decision tree")
figure_2.scatter(xs, [0, 0, 1, 1, 2, 3, 5, 9], marker="o", size=10, color="green", legend="Boosted Decision tree")
figure_2.xaxis.axis_label = "Training Epoch"
figure_2.yaxis.axis_label = "Classification Rate"

# Syntax gridplot([[array of plots in first line],
#                  [array of plots in second line],
#                  ...])
grid = gridplot([[figure_1], [figure_2]])

show(grid)
```

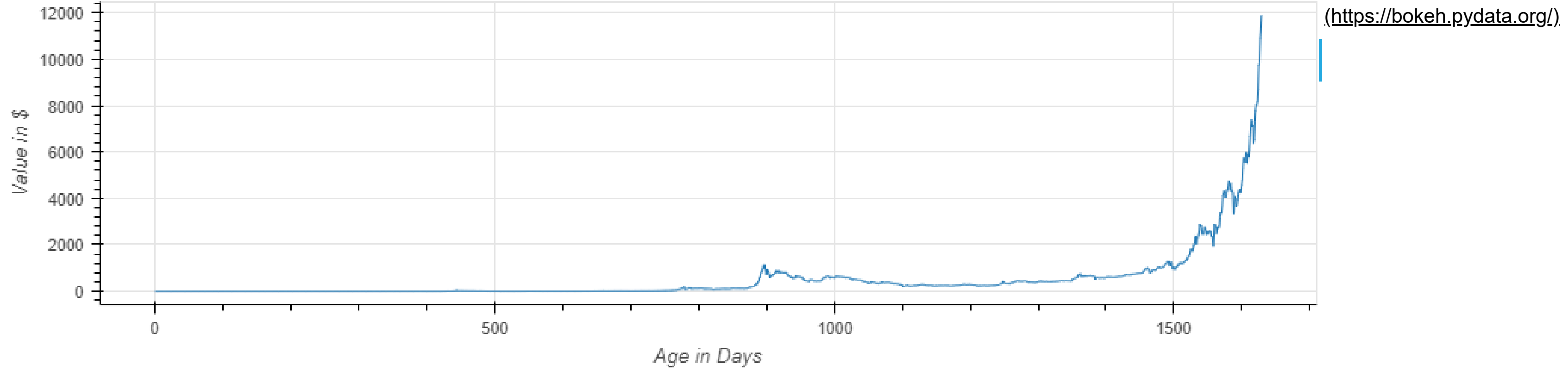


Scaling Normal vs Log

```
In [12]: import csv
with open('market-price.csv', "r") as csvfile:
    csv_data = list(csv.reader(csvfile, delimiter=","))
    timestamps = [x[0] for x in csv_data]
    values = [float(x[1]) for x in csv_data]
```

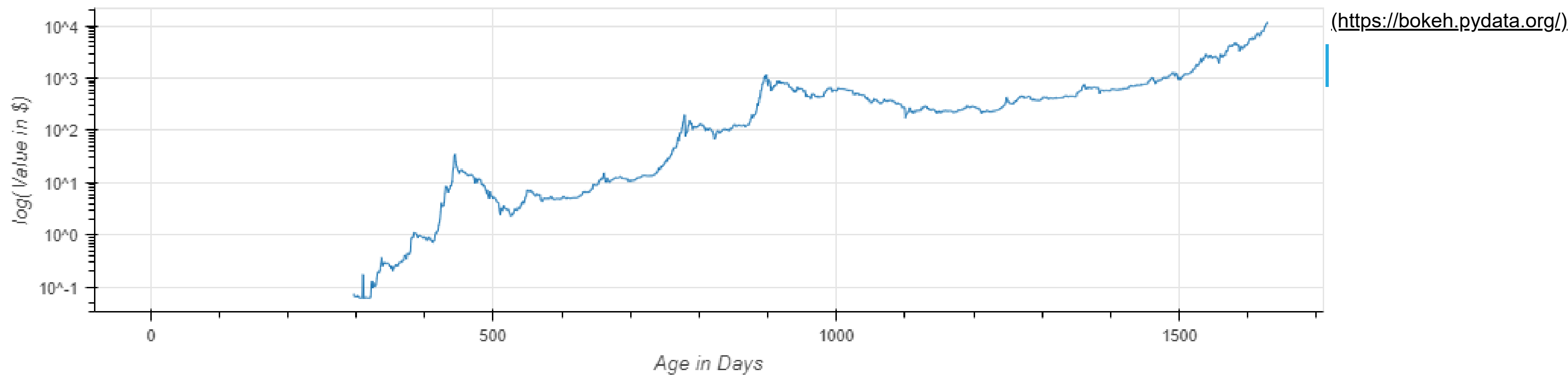
```
In [13]: figure = bokeh_figure(plot_width=900, plot_height=250)
ys = values
xs = np.arange(len(ys))
figure.line(xs, ys)

figure.xaxis.axis_label = "Age in Days"
figure.yaxis.axis_label = "Value in $"
show(figure)
```



```
In [14]: figure = bokeh_figure(plot_width=900, plot_height=250, y_axis_type="log")
ys = values
xs = np.arange(len(ys))
figure.line(xs, ys)

figure.xaxis.axis_label = "Age in Days"
figure.yaxis.axis_label = "log(Value in $)"
show(figure)
```



Hovertool

```
In [15]: from bokeh.models import HoverTool

ys = values
ts = timestamps
xs = np.arange(len(ys))

data = dict(x=xs,
            y=ys,
            ts=ts)
hover = HoverTool(tooltips=[
    ("Age in Days", "@x"),
    ("Date", "@ts"),
    ("Value in $", "@y{0,0.000}")
])

figure = bokeh_figure(plot_width=900, plot_height=250, tools=[hover])
figure.line("x", "y", source=data)

figure.xaxis.axis_label = "Age in Days"
figure.yaxis.axis_label = "Value in $"

show(figure)
```

