How can I calculate and plot a cumulative distribution function (CDF) in Python?

Authored by

stats writer

May 3, 2024

To calculate and plot a cumulative distribution function (CDF) in Python, you can use the built-in functions and libraries such as NumPy and Matplotlib. First, you need to import the necessary libraries and data. Then, you can use the cumulative sum function from NumPy to calculate the cumulative probability for each data point. Next, you can use Matplotlib to plot the CDF by passing the calculated cumulative probabilities and the corresponding data points. This will generate a step-like graph representing the cumulative distribution of the data. Finally, you can add labels and customize the plot to make it more visually appealing. This process can be used for creating a CDF for any dataset in Python.

Calculate & Plot a CDF in Python

You can use the following basic syntax to calculate the cumulative distribution function (CDF) in Python:

#sort data
x = np.sort(data)

#calculate CDF values
y = 1. * np.arange(len(data)) / (len(data) - 1)

#plot CDF
plt.plot(x, y)

The following examples show how to use this syntax in practice.

Example 1: CDF of Random Distribution

The following code shows how to calculate and plot a

cumulative distribution function (CDF) for a random sample of data in Python:

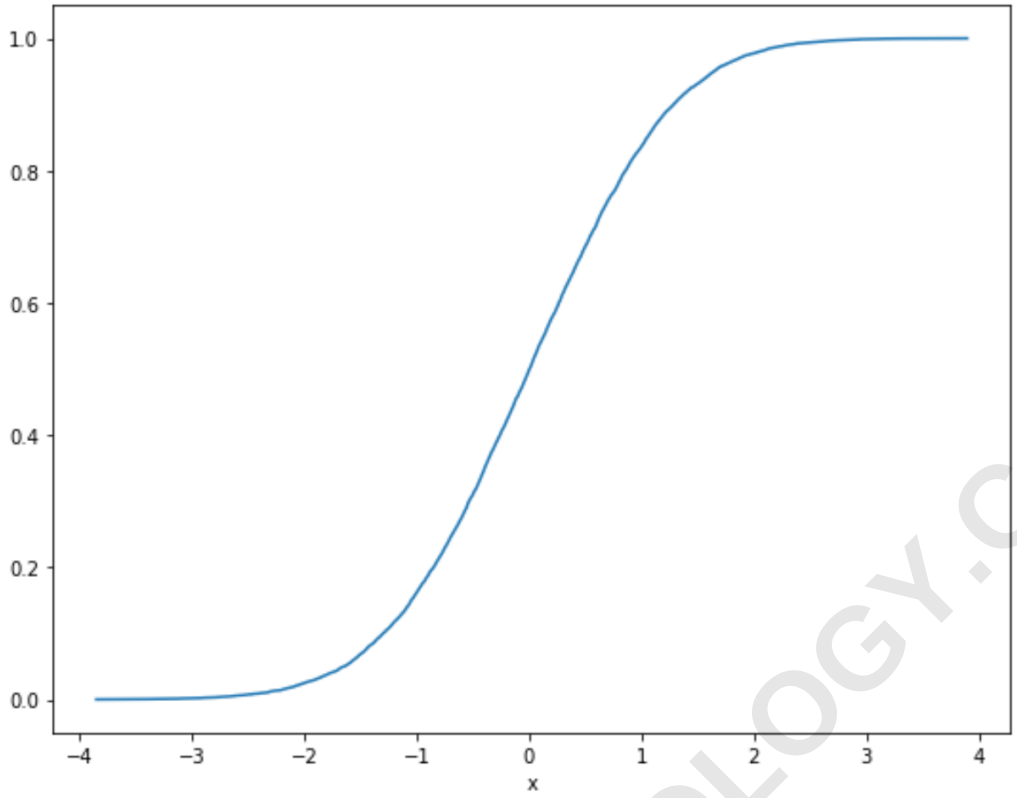
```
import numpy as np
import matplotlib.pyplotas plt

#define random sample of data
data = np.random.randn(10000)

#sort data
x = np.sort(data)

#calculate CDF values
y = 1. * np.arange(len(data)) / (len(data) - 1)

#plot CDF
plt.plot(x, y)
plt.xlabel('x')
```



The x-axis displays the raw data values and the y-axis displays the corresponding CDF values.

Example 2: CDF of Normal Distribution

If you'd like to plot the cumulative distribution function of a known distribution (such as the) then you can use the following functions from the library:

```
import numpy as np
import scipy
import matplotlib.pyplotas plt#generate data from normal distribution
```

```
data = np.random.randn(1000)

#sort data
x = np.sort(data)

#calculate CDF values
y = scipy.stats.norm.cdf(x)

#plot CDF
plt.plot(data_sorted, norm_cdf)

#plot CDF
plt.plot(x, y)
plt.xlabel('x')
```