Logistic regression is another statistical model where we try to classify where an object belongs to. In logistic regression, it may belong to two possible classes, therefore it's a binary classification. The logistic regression model has a dependend variable, an independent variables, coefficients, a threshold, a cost function which is used to show the difference between predicted and actual labels, and a logistic function. The logistic function applies a logistic function to the linear combination of the independent variables to get predictions. It's an S shaped curve and it maps the values between 0 and 1. The logistic function's formula is:
$f(x) = L / (1+e^{-k(x-x\_0)}$

Where:
L is the curve's maximum value
K is the logistic growth rate
x_0 is the midpoint of the logistic function
x is the real number

Practical example
We can use logistic regression to estimate the number of bacteria that grows over some time. Thus we have two data points, population and time. We can then use this to perform a prediction over some time.

```
import numpy as np

def logistic_function(x, L, k, x0):
    return L / (1 + np.exp(-k * (x - x0)))

time_points = np.array([1, 2, 3, 4, 5])
population_data = np.array([10, 20, 30, 40, 50])

# params
L = 100
k = 0.6
x0 = 3

for t in time_points:
    prediction = logistic_function(t, L, k, x0)
    print(f'Population prediction at time {t} hours: {prediction} bacteria')
```

```
Population prediction at time 1 hours: 23.147521650098238 bacteria
Population prediction at time 2 hours: 35.43436937742045 bacteria
Population prediction at time 3 hours: 50.0 bacteria
Population prediction at time 4 hours: 64.56563062257955 bacteria
Population prediction at time 5 hours: 76.85247834990176 bacteria
```