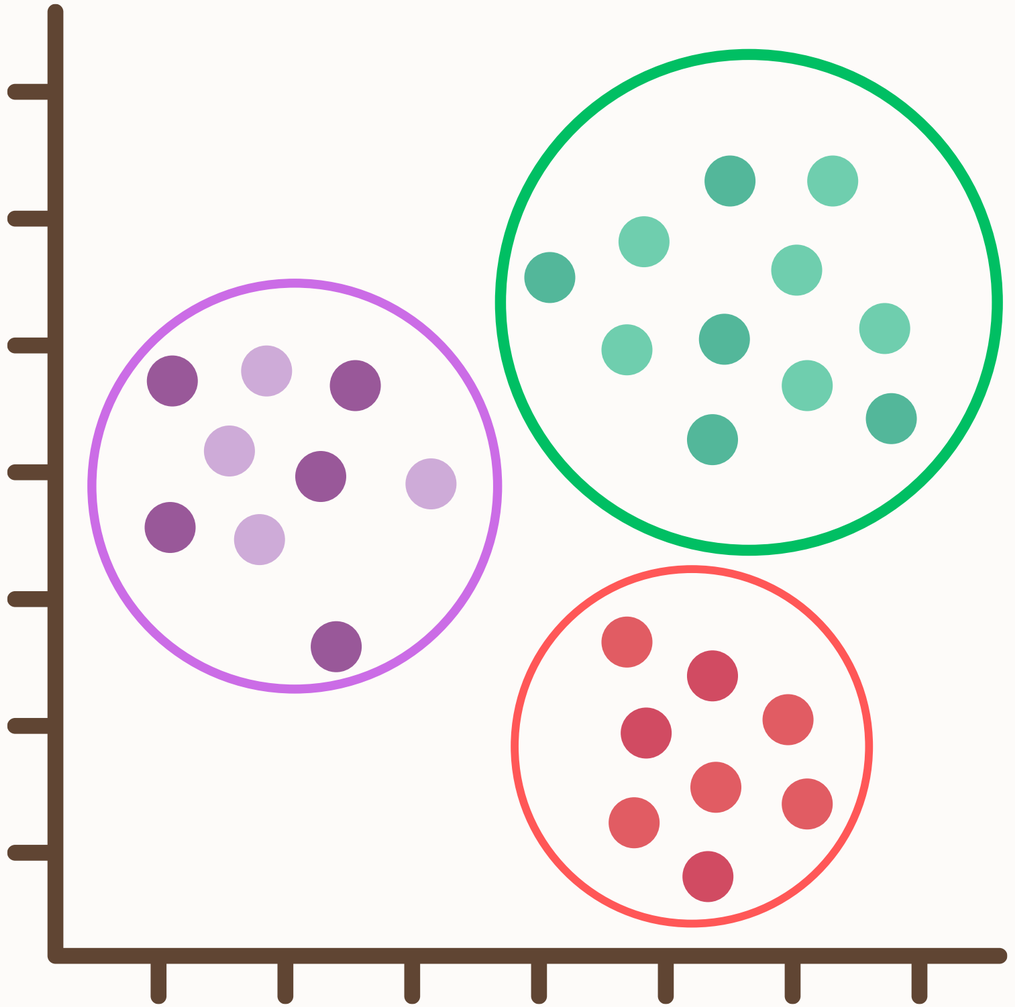
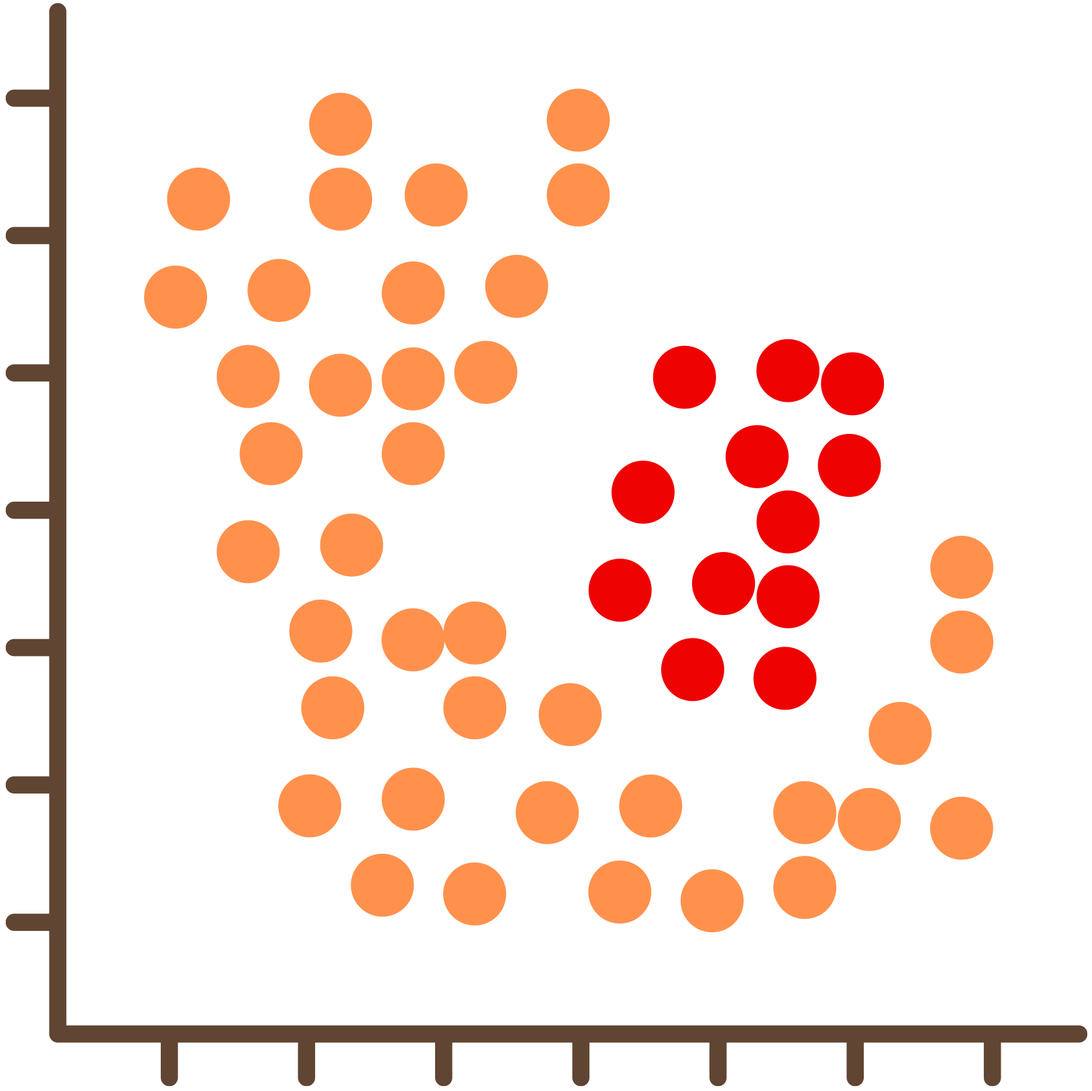
K-means can help us here

codefinity



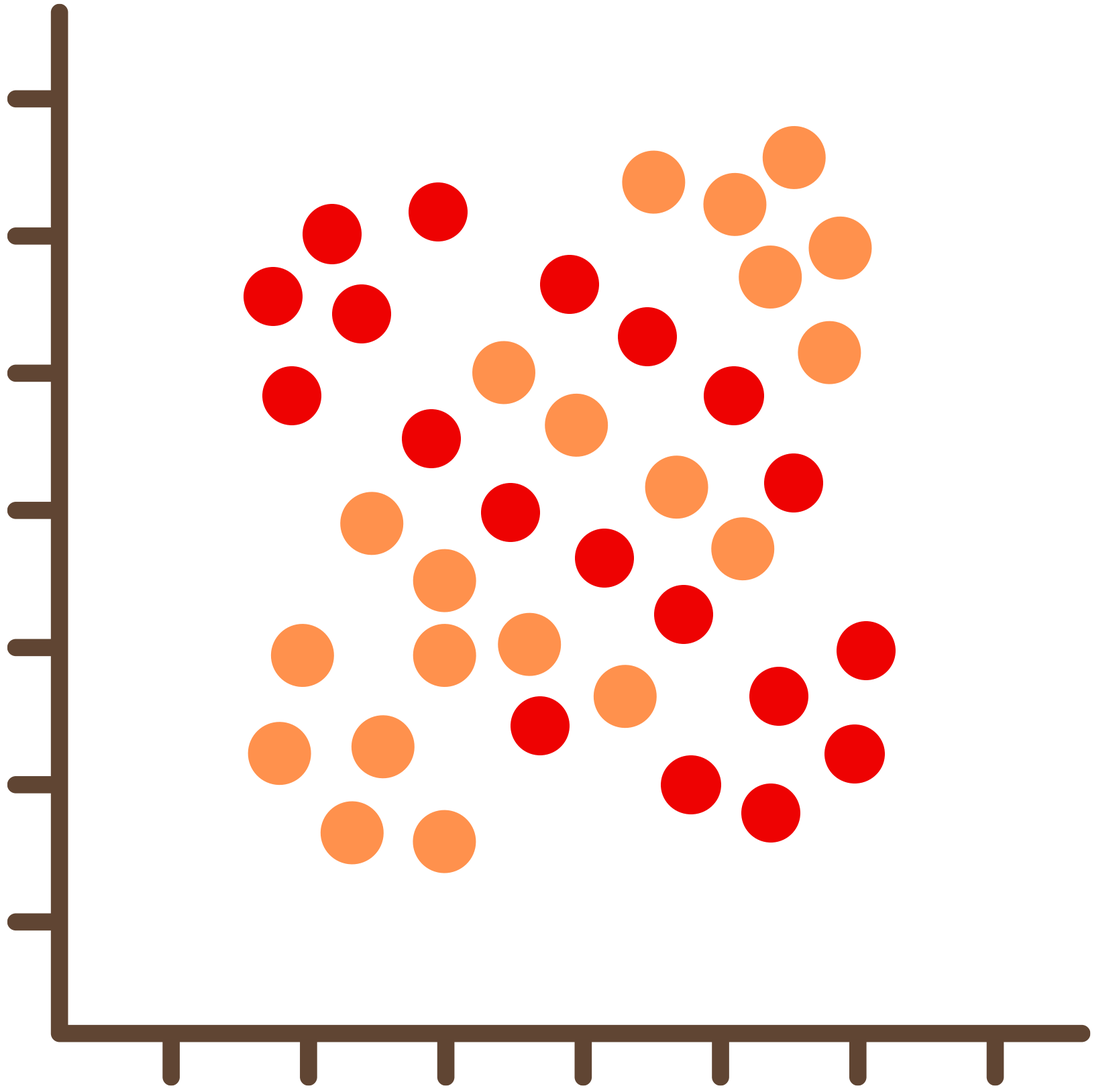
DBSCAN can help us here



codefinity

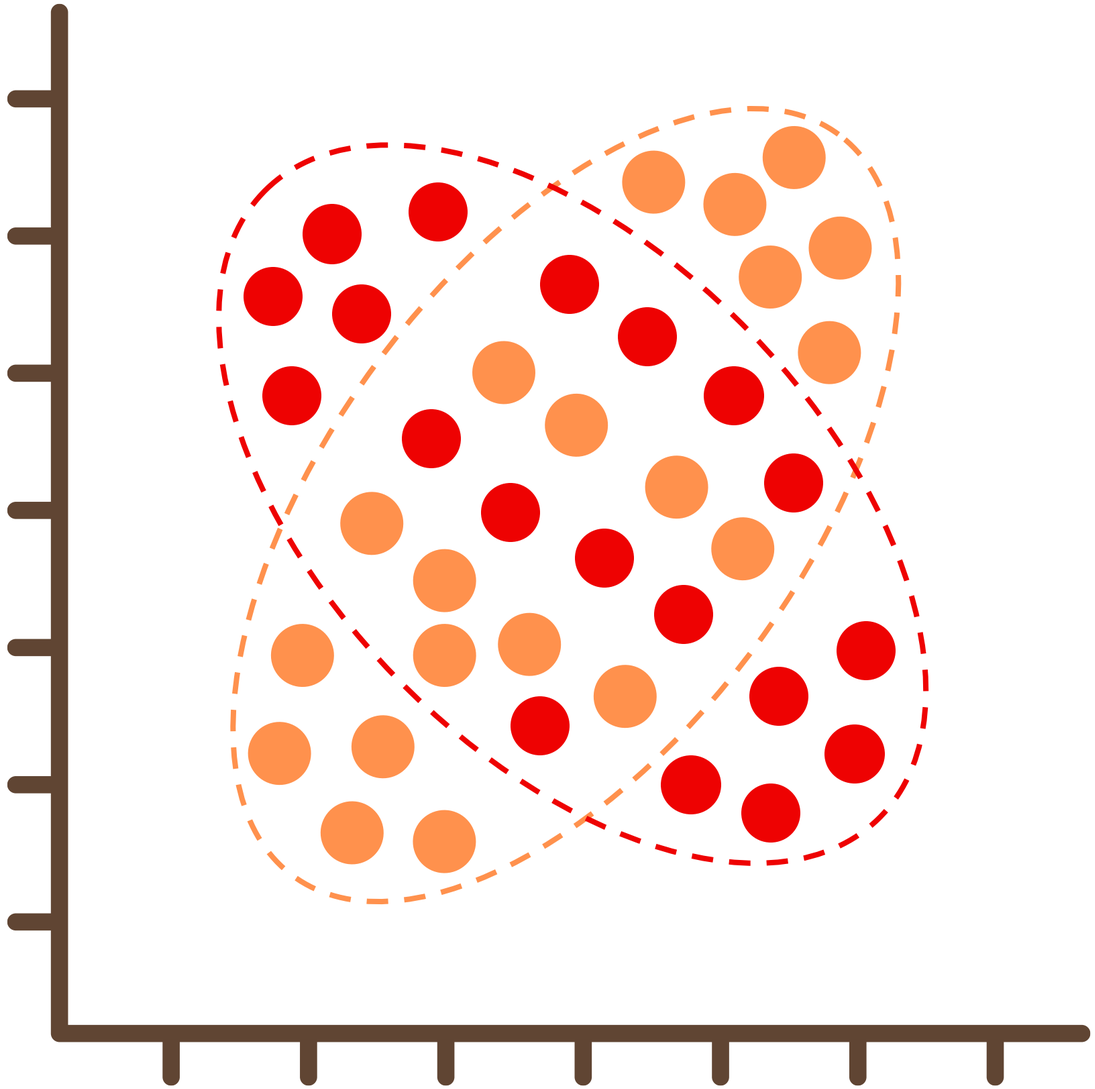
How about this type of Data?

codefinity



Gaussian Mixture Model

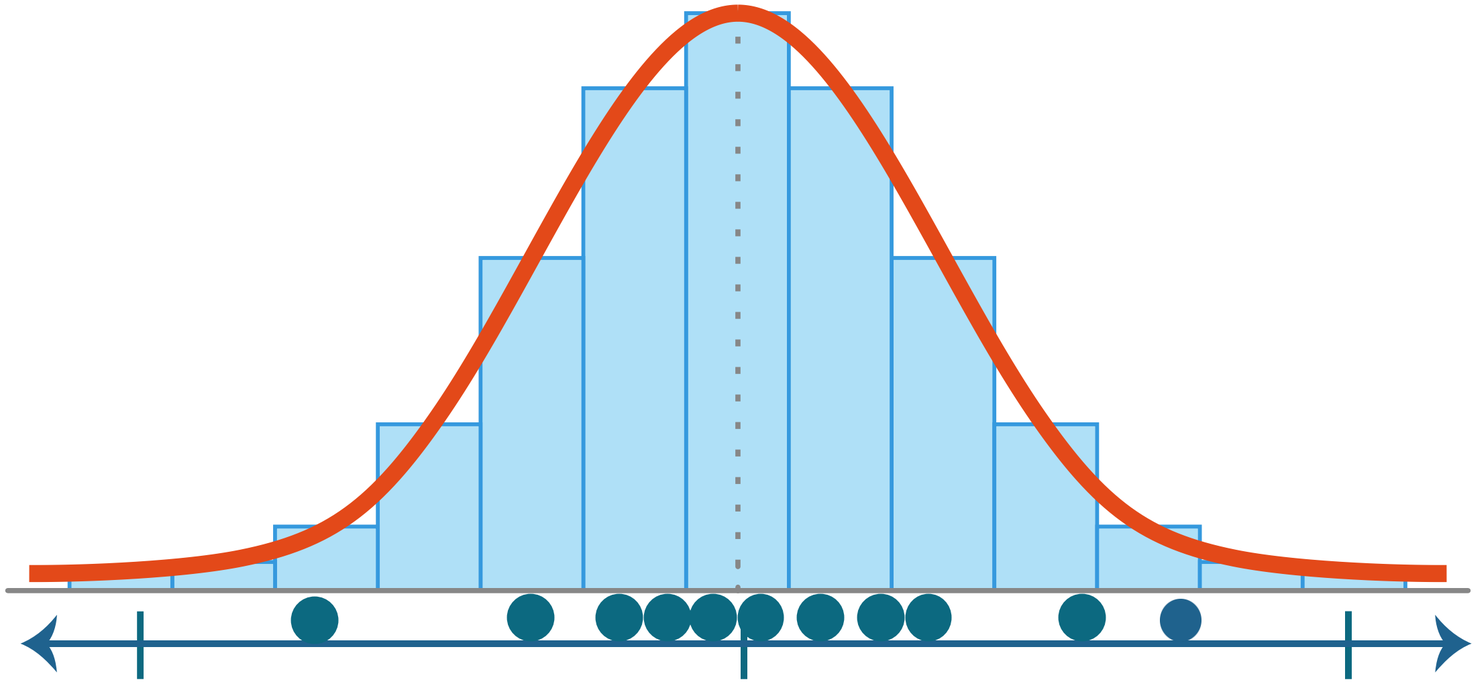
codefinity

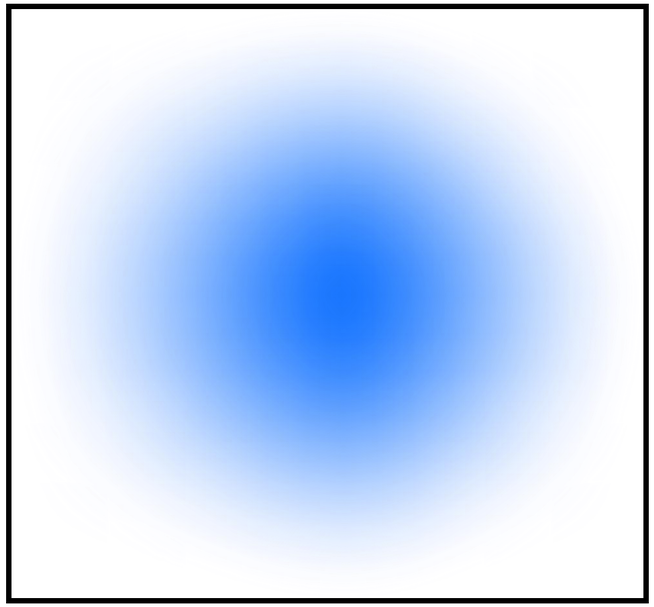


What is Gaussian?

codefinity

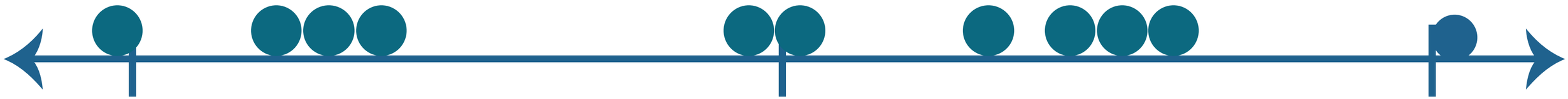
Gaussian Distribution vs Normal Distribution





How Gaussian Mixture Model Works

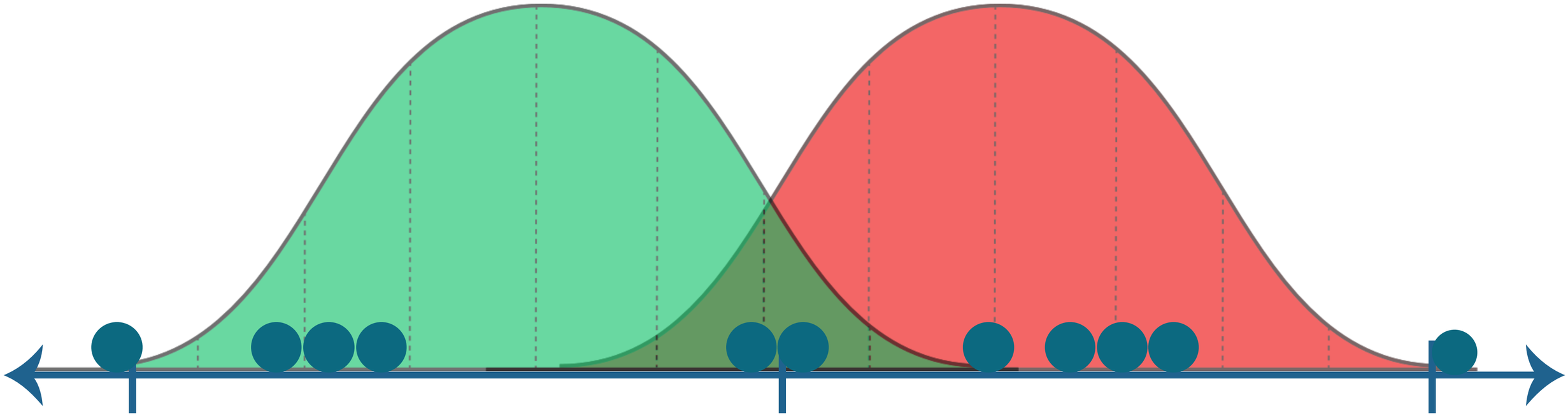
codefinity



How Gaussian Mixture Model Works

codefinity

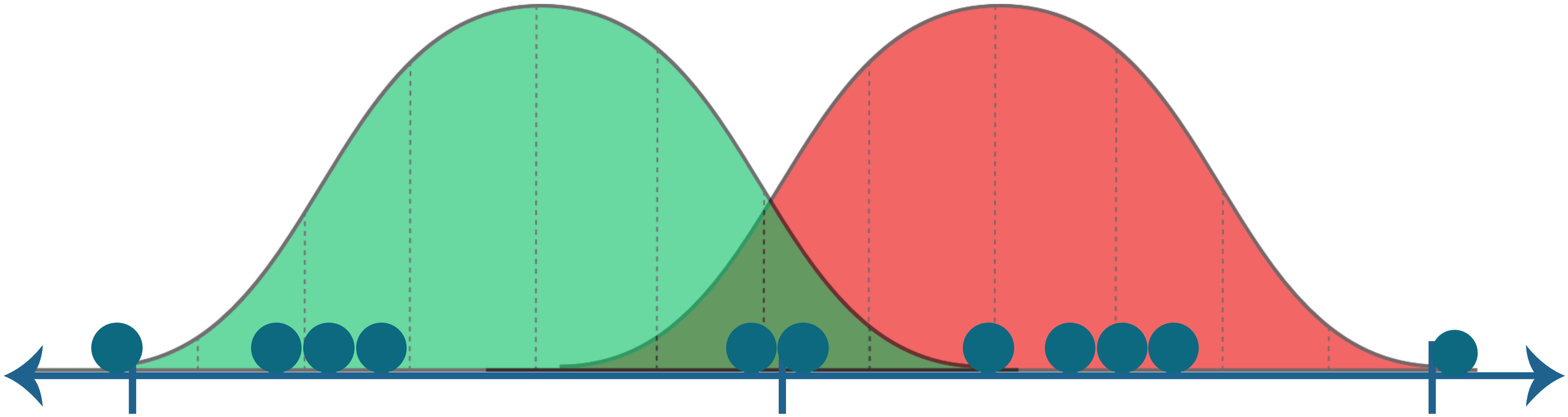
Step 1: Pick a random number of Gaussians



How Gaussian Mixture Model Works

codefinity

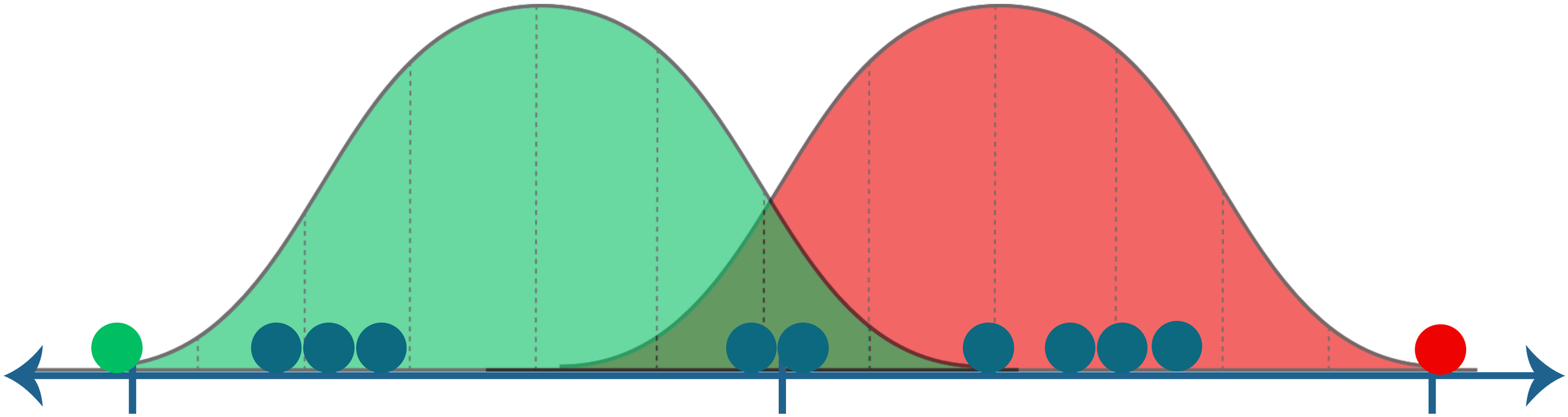
Step 2: Calculate the probability of each point’s belongingness



How Gaussian Mixture Model Works

codefinity

Step 2: Calculate the probability of each point’s belongingness

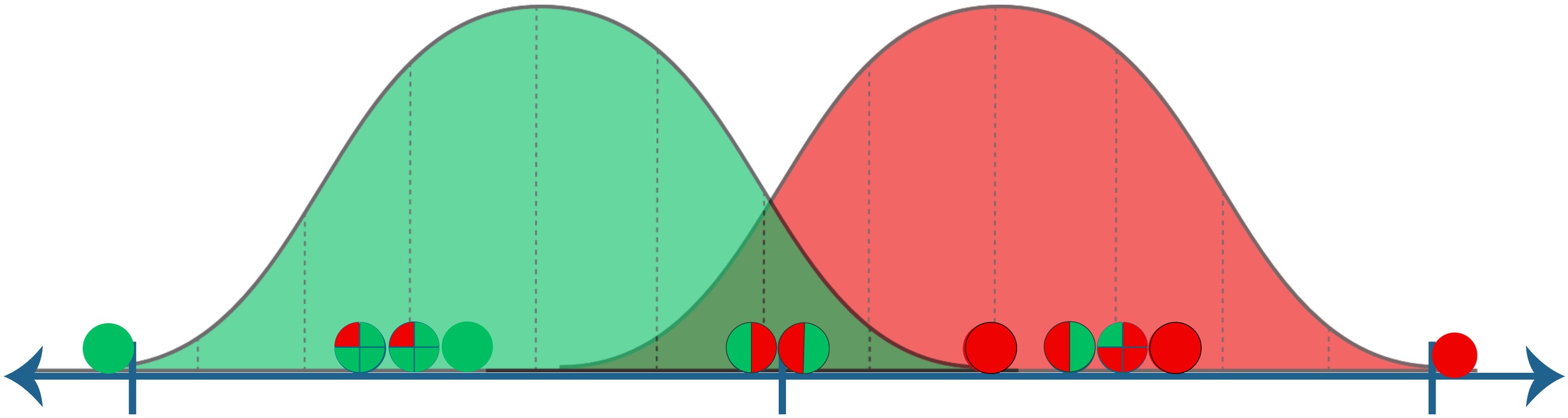


How Gaussian Mixture Model Works

codefinity

Step 2: Calculate the probability of each point’s belongingness

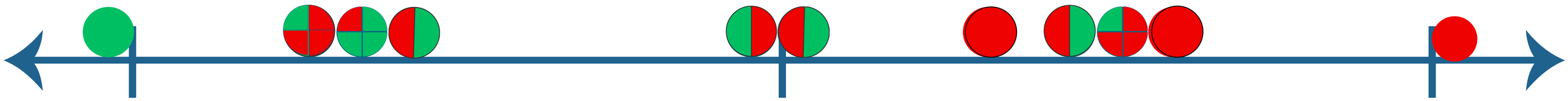
This process is called
“Responsibility”



How Gaussian Mixture Model Works

codefinity

Step 3: Shifting the Gaussians according to the points



How Gaussian Mixture Model Works

codefinity

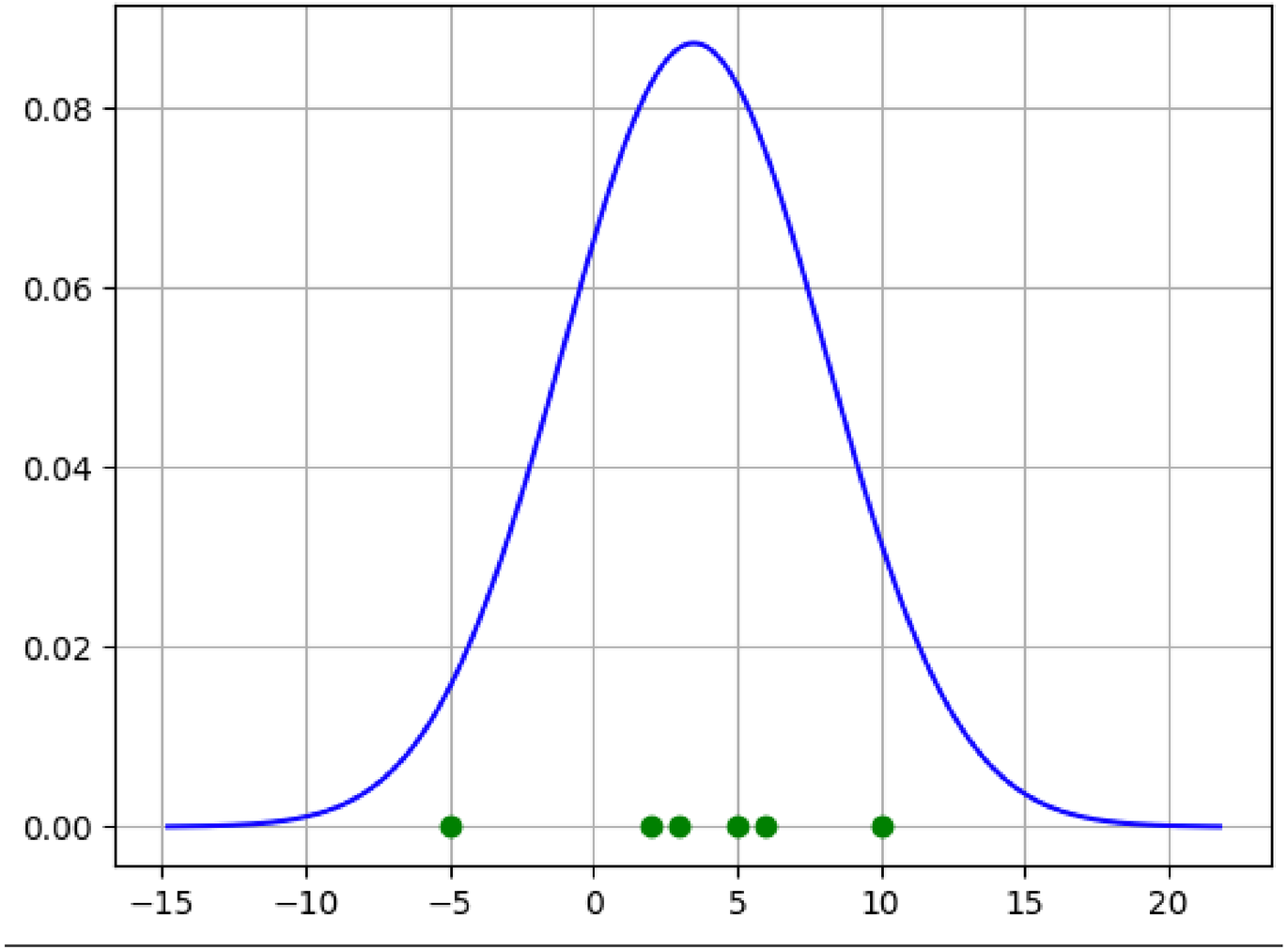
Step 3: Shifting the Gaussians according to the points

```python
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import norm

# Given data
data = [2, 5, 3, 6, 10, -5]
# Calculate mean and standard deviation
mean = np.mean(data)
std = np.std(data)
# Generate x values
x = np.linspace(mean - 4*std, mean + 4*std, 1000)
# Calculate the normal distribution values
y = norm.pdf(x, mean, std)

# Plot the normal distribution
plt.plot(x, y, label=f"Normal Distribution (mean={mean:.2f}, std={std:.2f})", color='blue')
# Plot the data points as green balls on the x-axis
plt.scatter(data, np.zeros_like(data), color='green', label='Data Points', zorder=5)
plt.grid(True)

# Display the plot
plt.show()
```

How Gaussian Mixture Model Works

codefinity

Step 3: Shifting the Gaussians according to the points

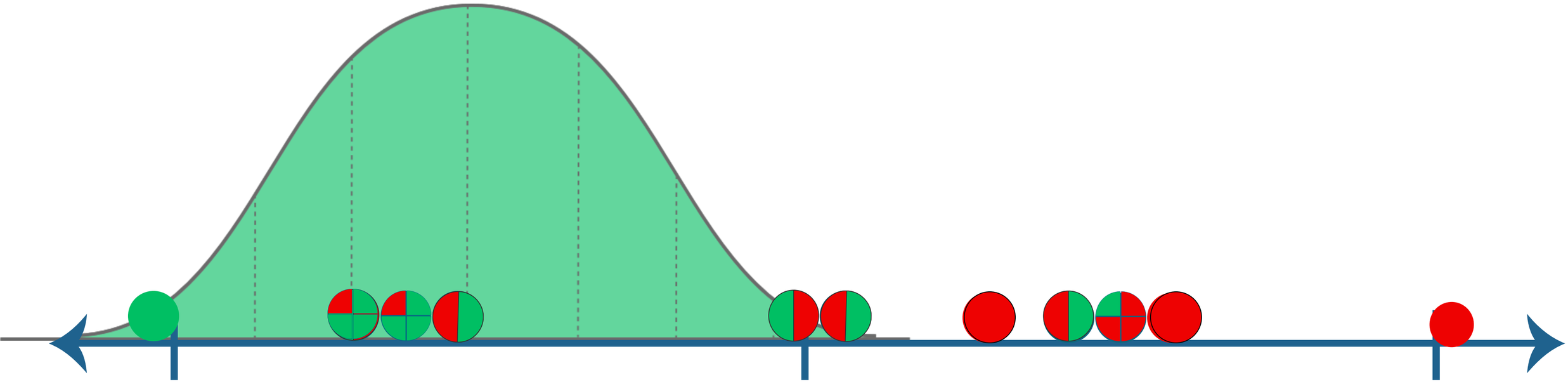



```python
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import norm

# Given data
data = [2, 5, 3, 6, 10, -5]
# Calculate mean and standard deviation
mean = np.mean(data)
std = np.std(data)
# Generate x values
x = np.linspace(mean - 4*std, mean + 4*std, 1000)
# Calculate the normal distribution values
y = norm.pdf(x, mean, std)

# Plot the normal distribution
plt.plot(x, y, label=f"Normal Distribution (mean={mean:.2f}, std={std:.2f})", color='blue')
# Plot the data points as green balls on the x-axis
plt.scatter(data, np.zeros_like(data), color='green', label='Data Points', zorder=5)
plt.grid(True)

# Display the plot
plt.show()
```

How Gaussian Mixture Model Works

codefinity

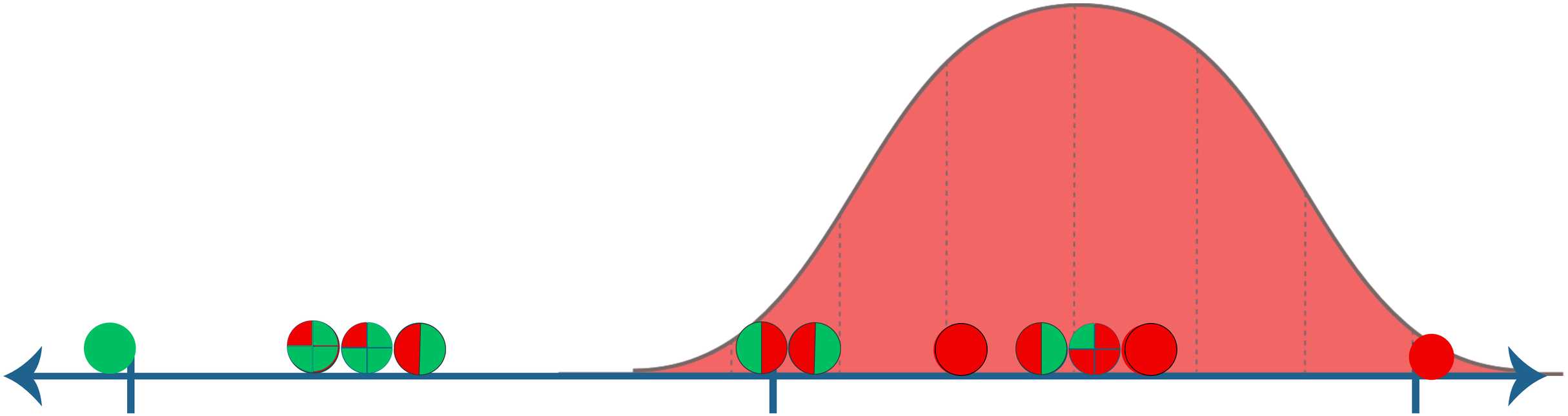
Step 3: Shifting the Gaussians according to the points



How Gaussian Mixture Model Works

codefinity

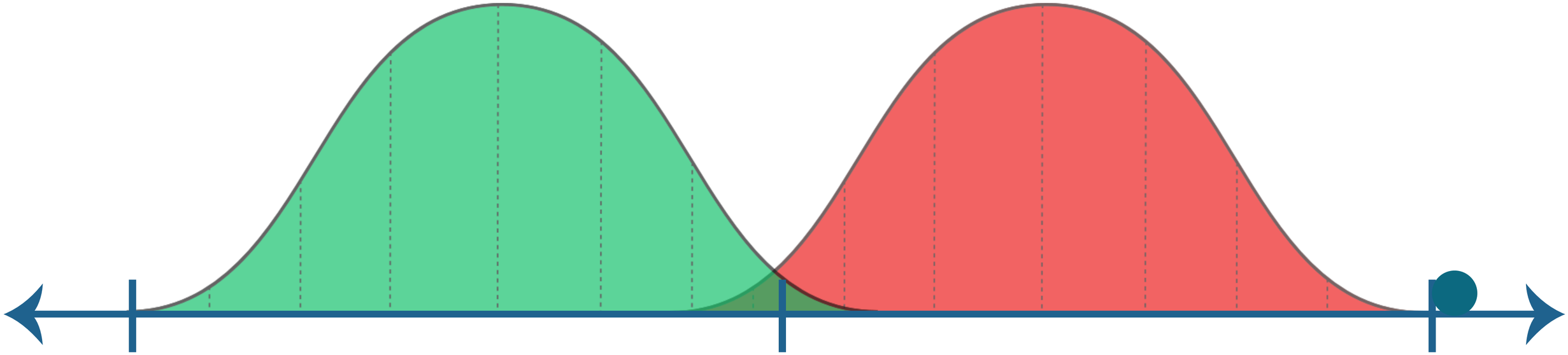
Step 3: Shifting the Gaussians according to the points



How Gaussian Mixture Model Works

codefinity

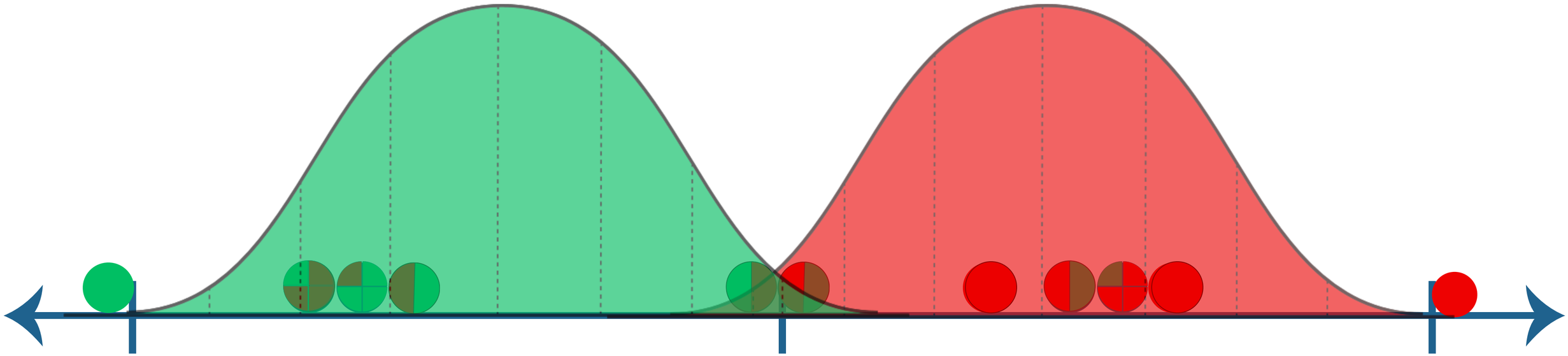
Step 4: Keep Repeating Step 2 & 3



How Gaussian Mixture Model Works

codefinity

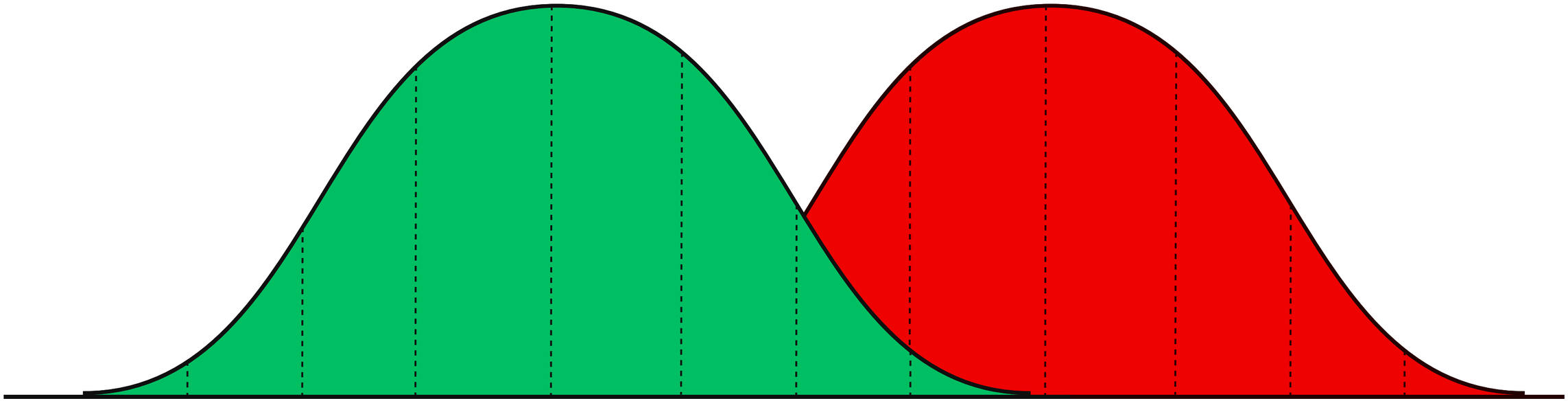
Step 4: Keep Repeating Step 2 & 3



How Gaussian Mixture Model Works

codefinity

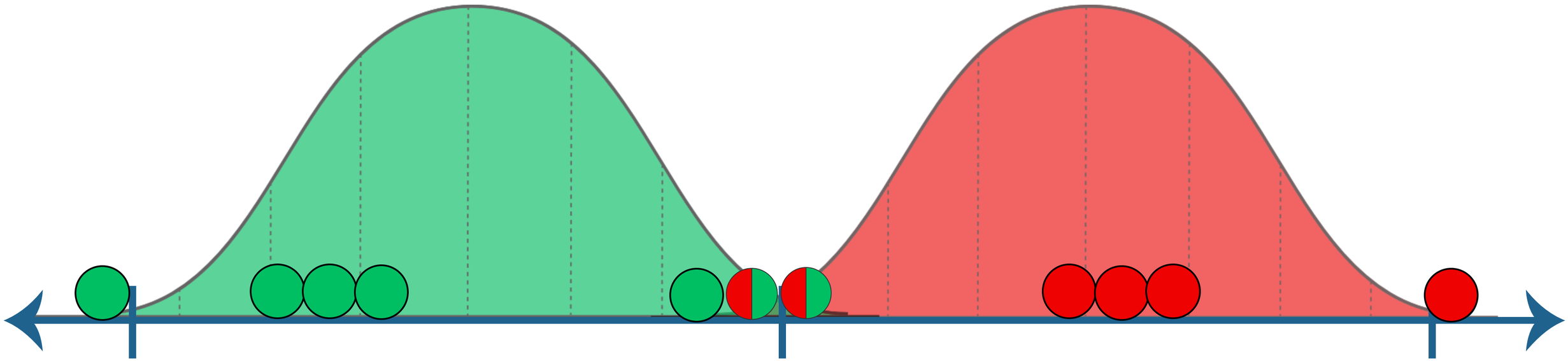
Step 4: Keep Repeating Step 2 & 3



How Gaussian Mixture Model Works

codefinity

Step 4: Keep Repeating Step 2 & 3

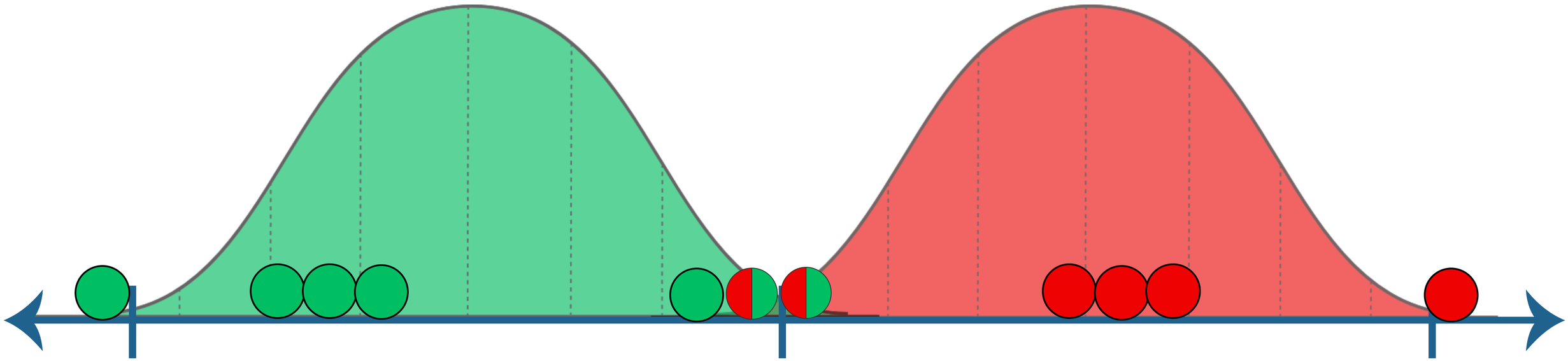


How Gaussian Mixture Model Works

codefinity

When do we stop?

When will we reach the convergence ?



Gaussian Mixture Model

codefinity

Similarity with K-means

Difference with K-means