SimpleBayesClassifier

February 21, 2026

1 Simple Bayes Classifier - Manual Calculation

1.1 Import Libraries

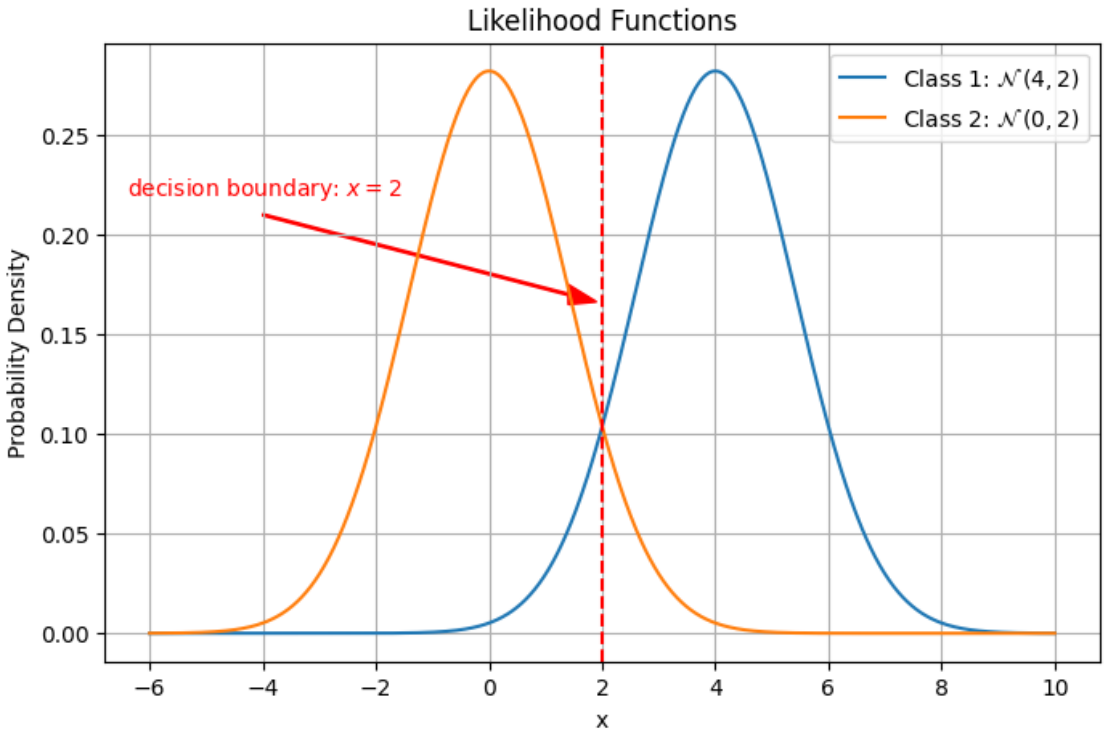
[1]:
```
import random as rnd
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np

from scipy import stats
```

1.2 Problems

1.2.1 Problem T2

[2]:
```
# Define x range
x = np.linspace(-6, 10, 1000)

# Parameters
mu1, mu2 = 0, 4
sigma = np.sqrt(2)

# Gaussian PDFs
p_w1 = stats.norm.pdf(x, mu2, sigma)  # N(4, 2)
p_w2 = stats.norm.pdf(x, mu1, sigma)  # N(0, 2)

# Plot
plt.figure(figsize=(8, 5))
plt.plot(x, p_w1, label=r"Class 1: $\mathcal{N}\,(4, 2)$")
plt.plot(x, p_w2, label=r"Class 2: $\mathcal{N}\,(0, 2)$")

plt.xlabel("x")
plt.ylabel("Probability Density")
plt.title("Likelihood Functions")
plt.legend()
plt.grid(True)
```

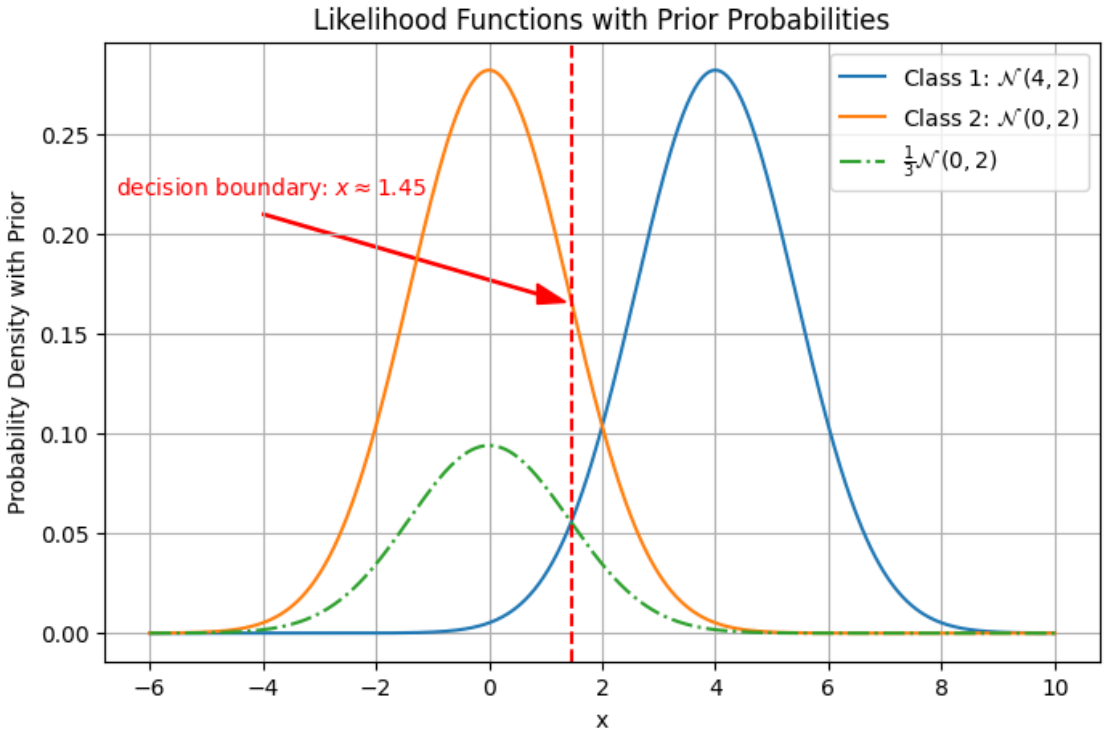
```
plt.axvline(x=2, color="red", linestyle="dashed")
plt.text(-6.4, 0.22, r"decision boundary: $x = 2$", color="red")
plt.arrow(-4, 0.21, 5.4, -0.04, head_width=0.01, head_length=0.5, fc="red",␣
 ↪ec="red")

plt.savefig("../images/LRT-T2.png", dpi=300)
plt.show()
```



1.2.2 Problem T3

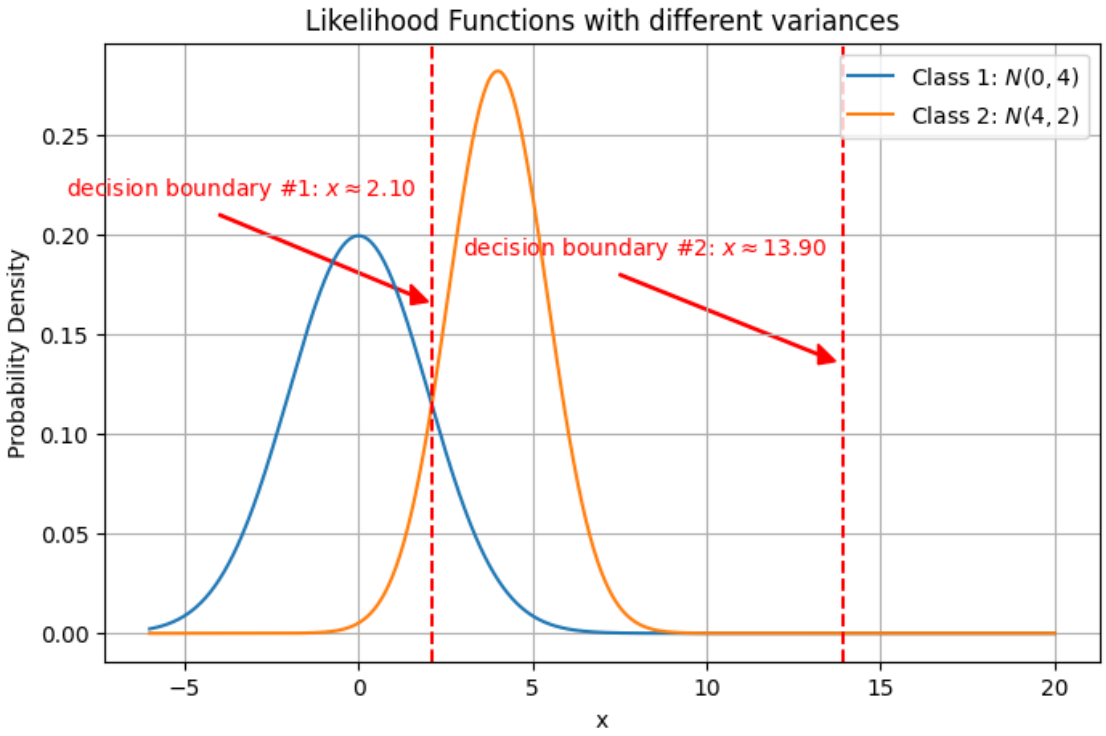
[3]:
```
# Define x range
x = np.linspace(-6, 10, 1000)

# Parameters
mu1, mu2 = 0, 4
sigma = np.sqrt(2)

# Gaussian PDFs
p_w1 = stats.norm.pdf(x, mu2, sigma)  # N(4, 2)
p_w2 = stats.norm.pdf(x, mu1, sigma)  # N(0, 2)
```

```
# Plot
plt.figure(figsize=(8, 5))
plt.plot(x, p_w1, label=r"Class 1: $\mathcal{N}\,(4, 2)$")
plt.plot(x, p_w2, label=r"Class 2: $\mathcal{N}\,(0, 2)$")
plt.plot(x, (1.0 / 3.0) * p_w2, label=r"$\frac{1}{3} \mathcal{N}\,(0, 2)$",␣
 ↪linestyle="dashdot")

plt.xlabel("x")
plt.ylabel("Probability Density with Prior")
plt.title("Likelihood Functions with Prior Probabilities")
plt.legend()
plt.grid(True)

boundary = 2 - (np.log(3) / 2.0)
plt.axvline(x=boundary, color="red", linestyle="dashed")
plt.text(-6.6, 0.22, r"decision boundary: $x \approx 1.45$", color="red")
plt.arrow(-4, 0.21, boundary + 3.4, -0.04, head_width=0.01, head_length=0.5,␣
 ↪fc="red", ec="red")

plt.savefig("../images/LRT-T3.png", dpi=300)
plt.show()
```



1.2.3 Problem OT3

[4]:
```
# Define x range
x = np.linspace(-6, 20, 1000)

# Parameters
mu1, mu2 = 4, 0
sigma1, sigma2 = np.sqrt(2), 2

# Gaussian PDFs
p_w1 = stats.norm.pdf(x, mu2, sigma2)  # N(0, 4)
p_w2 = stats.norm.pdf(x, mu1, sigma1)  # N(4, 2)

# Plot
plt.figure(figsize=(8, 5))
plt.plot(x, p_w1, label=r"Class 1: $N(0, 4)$")
plt.plot(x, p_w2, label=r"Class 2: $N(4, 2)$")

plt.xlabel("x")
plt.ylabel("Probability Density")
plt.title("Likelihood Functions with different variances")
plt.legend(loc="upper right")
plt.grid(True)

boundary1, boundary2 = 2.10, 13.90

plt.axvline(x=boundary1, color="red", linestyle="dashed")
plt.text(-8.4, 0.22, r"decision boundary #1: $x \approx 2.10$", color="red")
plt.arrow(-4, 0.21, boundary1 + 3.4, -0.04, head_width=0.01, head_length=0.5,␣
 ↪fc="red", ec="red")

plt.axvline(x=boundary2, color="red", linestyle="dashed")
plt.text(3, 0.19, r"decision boundary #2: $x \approx 13.90$", color="red")
plt.arrow(7.5, 0.18, boundary2 - 8.2, -0.04, head_width=0.01, head_length=0.5,␣
 ↪fc="red", ec="red")

plt.savefig("../images/LRT-OT3.png", dpi=300)
plt.show()
```


Likelihood Functions with different variances
Class 1: $N(0, 4)$
Class 2: $N(4, 2)$
decision boundary #1: $x \approx 2.10$
decision boundary #2: $x \approx 13.90$
Probability Density
0.25
0.20
0.15
0.10
0.05
0.00
−5
0
5
10
15
20
x
