h3_code

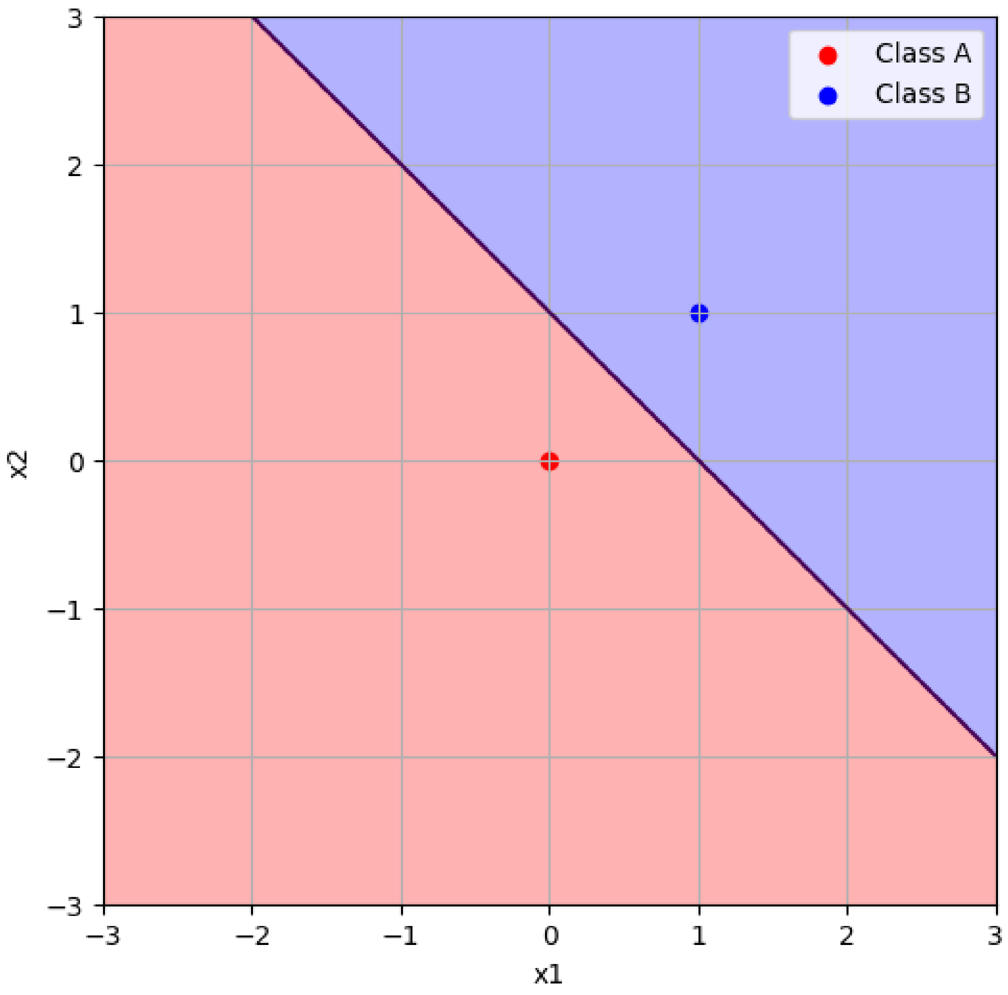
February 21, 2026

[2]:
```python
import numpy as np
import matplotlib.pyplot as plt
```

1 3

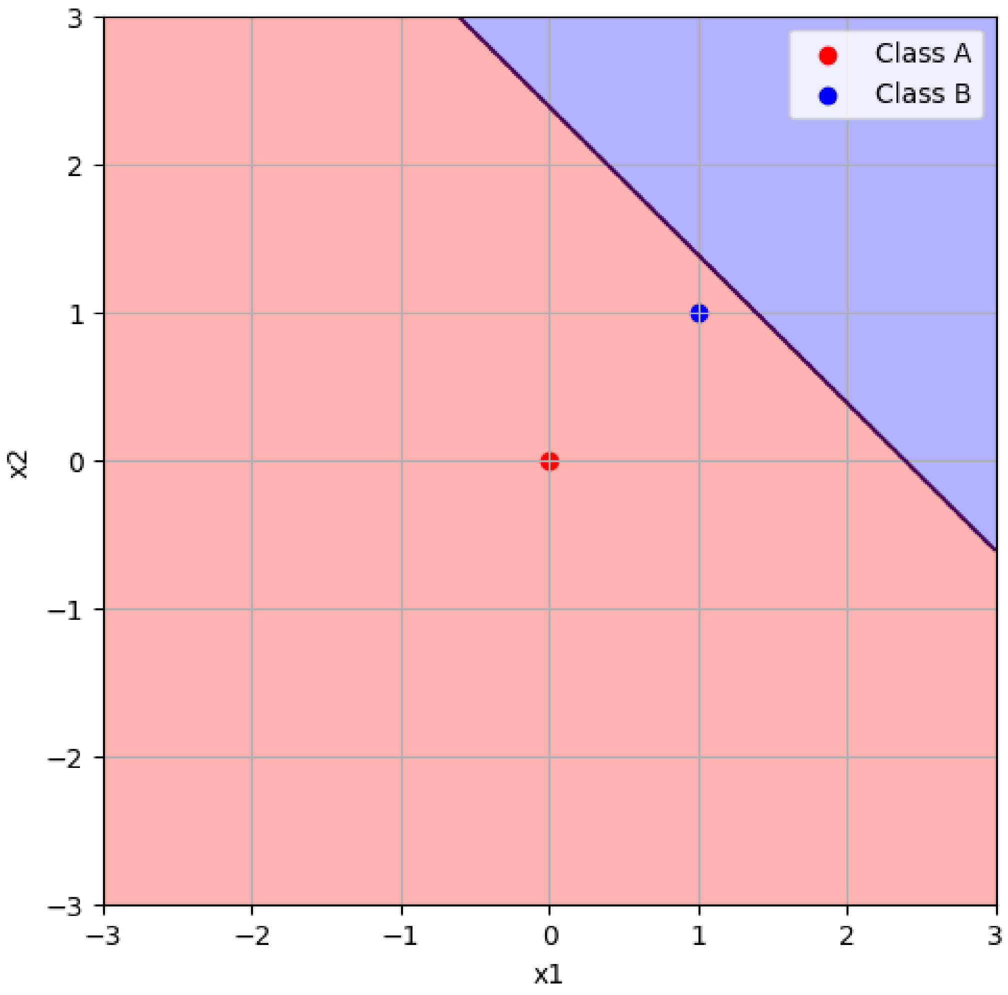
1.1 a

[9]:
```python
mu1 = np.array([0, 0])
mu2 = np.array([1, 1])

x1 = np.linspace(-3, 3, 100)
x2 = np.linspace(-3, 3, 100)
X1, X2 = np.meshgrid(x1, x2)

Z = X1 + X2 - 1

plt.figure(figsize=(6, 6))
plt.contourf(X1, X2, Z, levels=[-10, 0, 10], colors=["red", "blue"], alpha=0.3)
plt.contour(X1, X2, Z, levels=[0])
plt.scatter(mu1[0], mu1[1], color='red', label='Class A')
plt.scatter(mu2[0], mu2[1], color='blue', label='Class B')
plt.legend()
plt.xlabel('x1')
plt.ylabel('x2')
plt.grid(True)
plt.show()
```



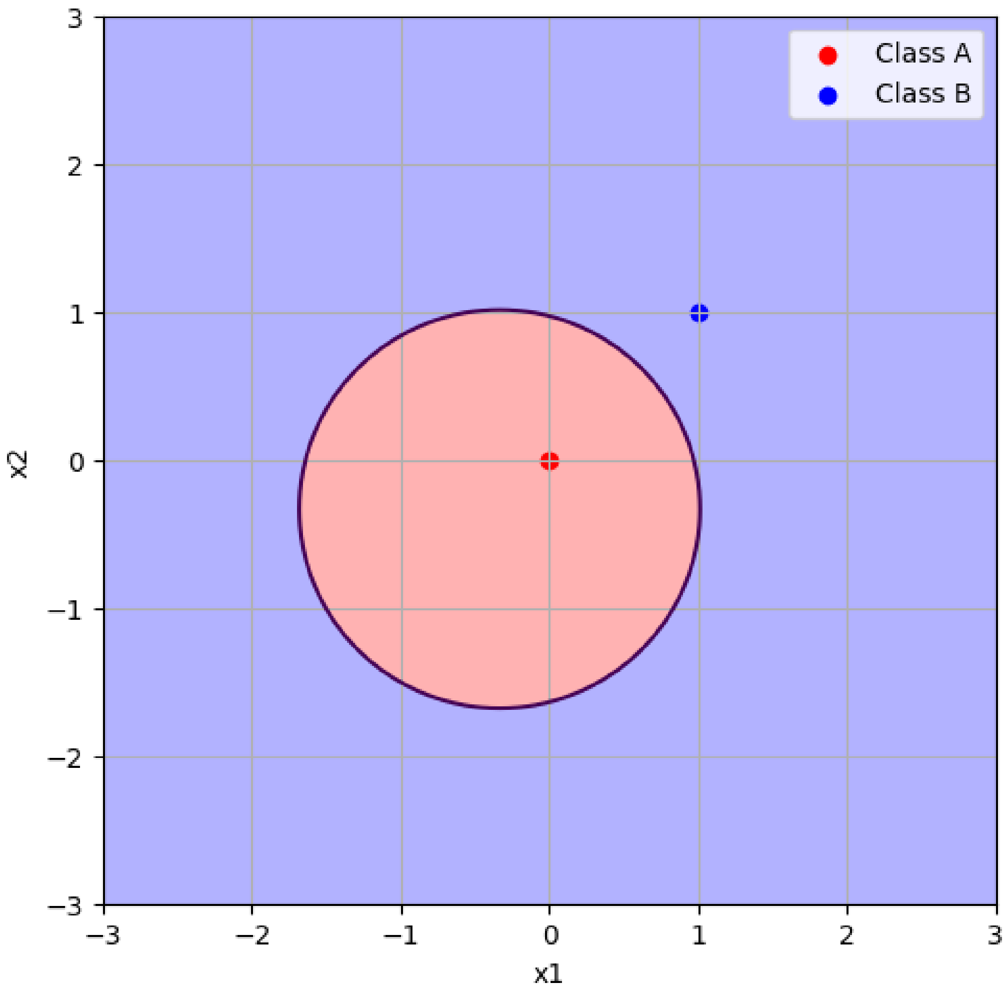
1.2 b

```
[10]: p = 0.8
      Z = (X1 + X2) - (1 + np.log(p / (1 - p)))

      plt.figure(figsize=(6, 6))
      plt.contourf(X1, X2, Z, levels=[-10, 0, 10], colors=["red", "blue"], alpha=0.3)
      plt.contour(X1, X2, Z, levels=[0])
      plt.scatter(mu1[0], mu1[1], color='red', label='Class A')
      plt.scatter(mu2[0], mu2[1], color='blue', label='Class B')
      plt.legend()
      plt.xlabel('x1')
      plt.ylabel('x2')
      plt.grid(True)
      plt.show()
```



1.3 c

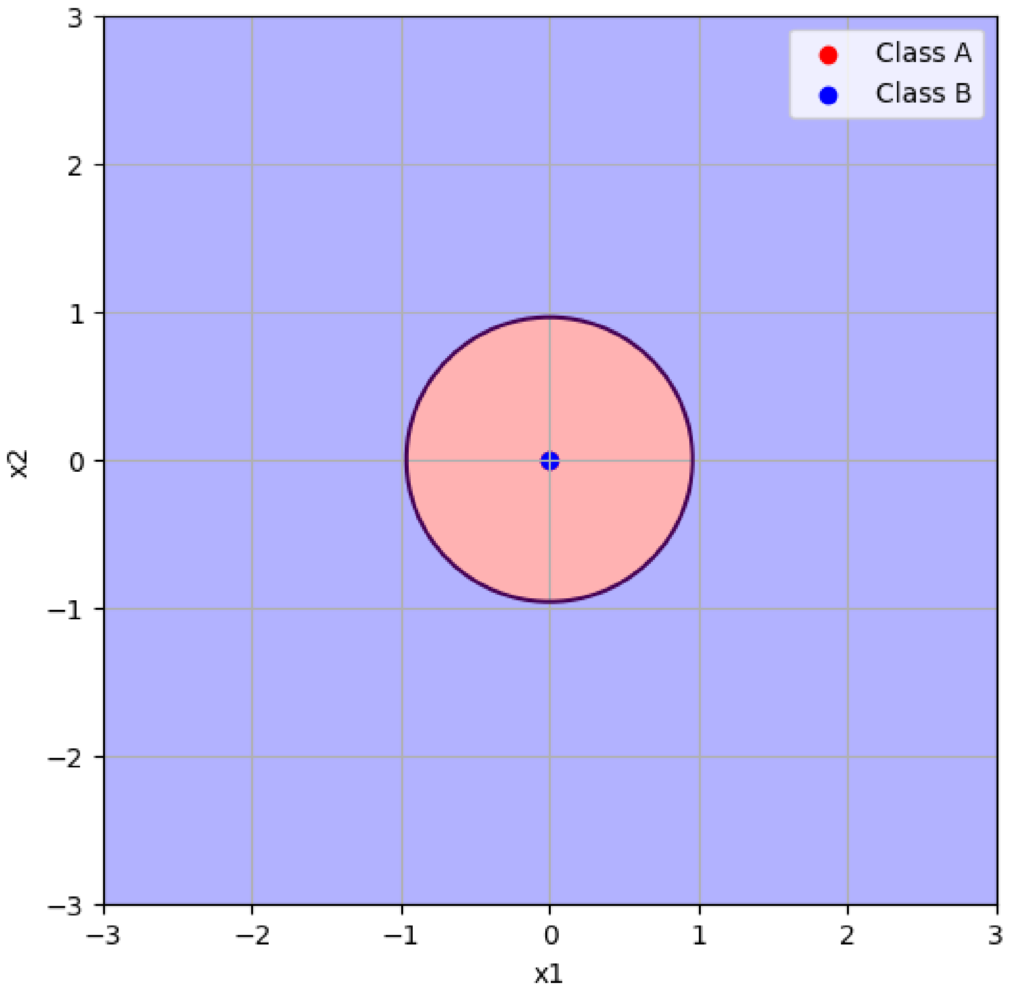
```
[11]: p = 0.5
      sigma2 = 0.25
      d = 2

      A_dist2 = (X1 - mu1[0])**2 + (X2 - mu1[1])**2
      B_dist2 = (X1 - mu2[0])**2 + (X2 - mu2[1])**2

      Z = -(d/2)*np.log(sigma2) - (1/(2*sigma2))*A_dist2 + 0.5*B_dist2

      plt.figure(figsize=(6, 6))
      plt.contourf(X1, X2, Z, levels=[-1e9, 0, 1e9], colors=["blue", "red"], alpha=0.
       ↪3)
      plt.contour(X1, X2, Z, levels=[0])
      plt.scatter(mu1[0], mu1[1], color='red', label='Class A')
      plt.scatter(mu2[0], mu2[1], color='blue', label='Class B')
      plt.legend()
      plt.xlabel('x1')
      plt.ylabel('x2')
      plt.grid(True)
      plt.show()
```



1.4 d

[13]:
```
dist2 = X1**2 + X2**2

Z = -(d/2)*np.log(sigma2) - (1/(2*sigma2))*dist2 + 0.5*dist2

plt.figure(figsize=(6, 6))
plt.contourf(X1, X2, Z, levels=[-1e9, 0, 1e9], colors=["blue", "red"], alpha=0.
  ↪3)
```

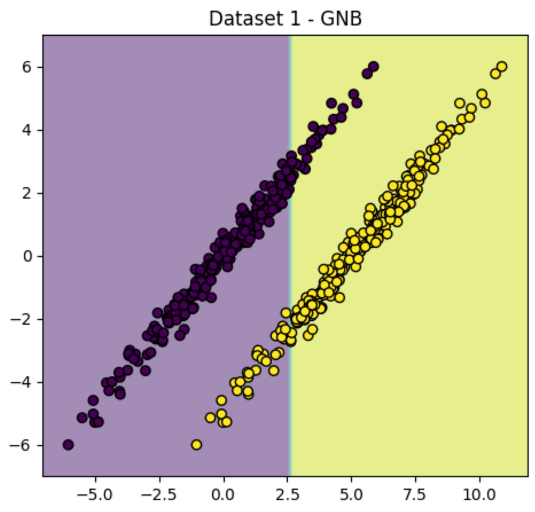
```
plt.contour(X1, X2, Z, levels=[0])
plt.scatter(mu1[0], mu1[1], color='red', label='Class A')
plt.scatter(mu1[0], mu1[1], color='blue', label='Class B')
plt.legend()
plt.xlabel('x1')
plt.ylabel('x2')
plt.grid(True)
plt.show()
```

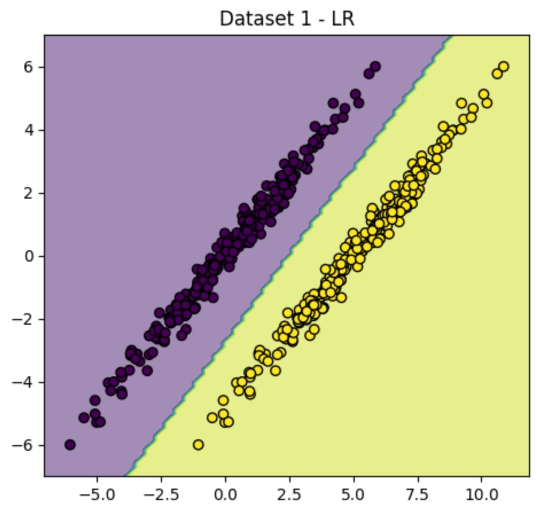


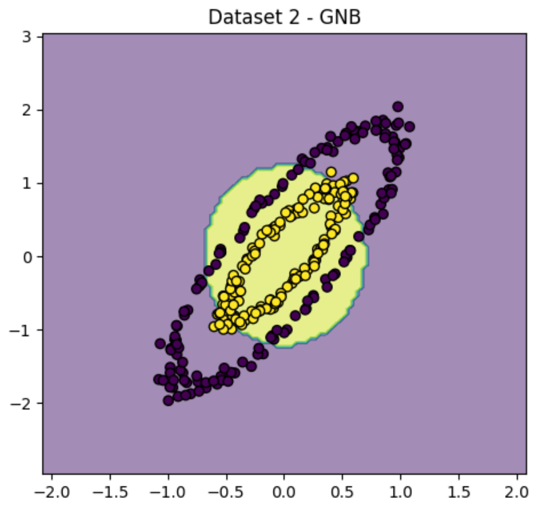
2 4

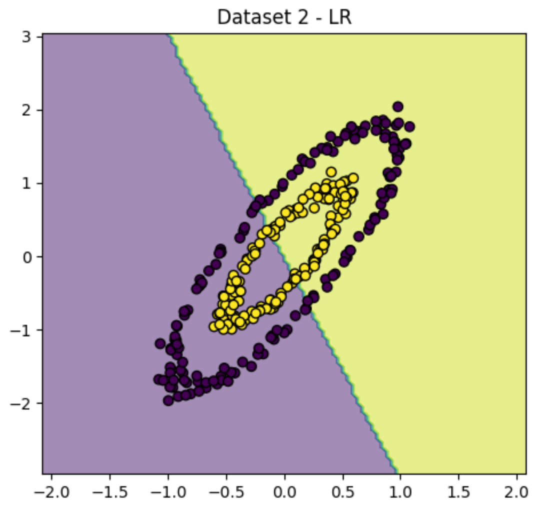
[14]:
```
from sklearn.naive_bayes import GaussianNB
from sklearn.linear_model import LogisticRegression

data1 = np.load('data-1.npy')
```

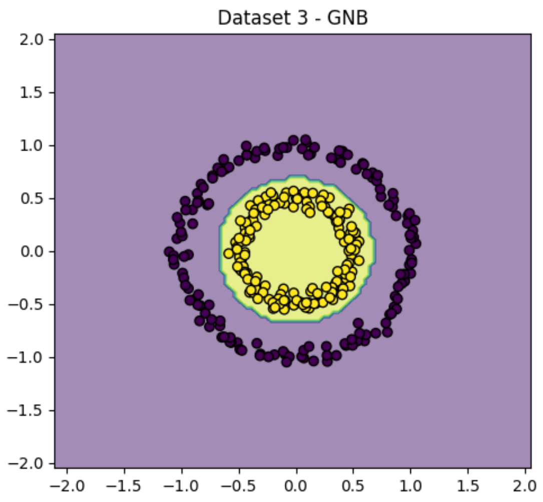
```
data2 = np.load('data-2.npy')
data3 = np.load('data-3.npy')

datasets = [data1, data2, data3]
for i, data in enumerate(datasets):
    X = data[:, :2]
    y = data[:, 2]

    # Fit Gaussian Naive Bayes
    gnb = GaussianNB()
    gnb.fit(X, y)

    # Fit Logistic Regression
    lr = LogisticRegression()
    lr.fit(X, y)

    # Plotting
    plt.figure(figsize=(12, 5))

    # Plot decision boundaries
    x_min, x_max = X[:, 0].min() - 1, X[:, 0].max() + 1
    y_min, y_max = X[:, 1].min() - 1, X[:, 1].max() + 1
    xx, yy = np.meshgrid(np.linspace(x_min, x_max, 100), np.linspace(y_min,␣
 ↪y_max, 100))

    Z_gnb = gnb.predict(np.c_[xx.ravel(), yy.ravel()])
    Z_gnb = Z_gnb.reshape(xx.shape)

    Z_lr = lr.predict(np.c_[xx.ravel(), yy.ravel()])
    Z_lr = Z_lr.reshape(xx.shape)

    plt.subplot(1, 2, 1)
    plt.contourf(xx, yy, Z_gnb, alpha=0.5, cmap='viridis')
    plt.scatter(X[:, 0], X[:, 1], c=y, edgecolor='k', marker='o')
    plt.title(f'Dataset {i+1} - GNB')

    plt.subplot(1, 2, 2)
    plt.contourf(xx, yy, Z_lr, alpha=0.5, cmap='viridis')
    plt.scatter(X[:, 0], X[:, 1], c=y, edgecolor='k', marker='o')
    plt.title(f'Dataset {i+1} - LR')
    plt.show()
```

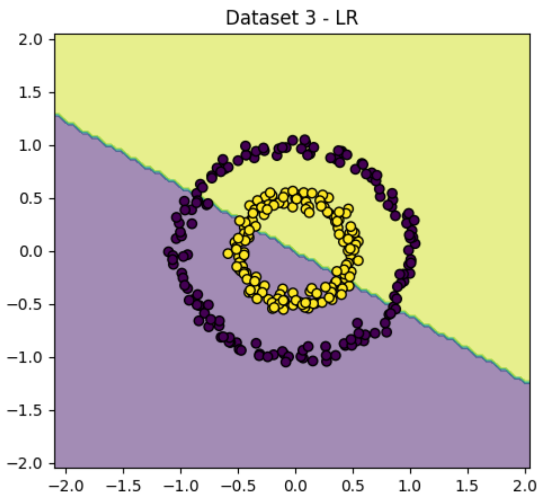

Dataset 1 - GNB
6
4
2
0
−2
−4
−6
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0


Dataset 1 - LR
6
4
2
0
−2
−4
−6
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0


Dataset 2 - GNB
3
2
1
0
−1
−2
−2.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0


Dataset 2 - LR
3
2
1
0
−1
−2
−2.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0


Dataset 3 - GNB
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−2.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0


Dataset 3 - LR
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−2.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0