

```
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import accuracy_score

# Generate data from different continuous distributions
def generate_data(n_samples, class_probs=[0.5, 0.5]):
    data = []
    labels = []
    for i, prob in enumerate(class_probs):
        # Generate data for class i
        if i == 0:
            # Normal distribution with mean 0, std 1
            data_class = np.random.normal(0, 1, int(n_samples * prob))
        else:
            # Normal distribution with mean 2, std 1
            data_class = np.random.normal(2, 1, int(n_samples * prob))
        data.append(data_class)
        labels.append([i] * len(data_class))

    # Combine data and labels
    data = np.concatenate(data)
    labels = np.concatenate(labels)

    # Shuffle data
    indices = np.random.permutation(len(data))
    data = data[indices]
    labels = labels[indices]

    return data, labels


# Generate training data
X, y = generate_data(1000)

# Split data into train and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Train a Gaussian Naive Bayes classifier
model = GaussianNB()
model.fit(X_train.reshape(-1, 1), y_train)

# Make predictions
y_pred = model.predict(X_test.reshape(-1, 1))

# Evaluate accuracy
print("Accuracy:", accuracy_score(y_test, y_pred))
```

 Accuracy: 0.8