机器学习实验三：贝叶斯分类器

学生姓名：廖望
学号：2210556

一、实验目标

这一实验围绕贝叶斯分类方法展开，核心目标是在亲手实现高斯朴素贝叶斯分类器的过程中，把课上相对抽象的概率公式和实际的分类任务对应起来。我希望通过这次实验，真正理解贝叶斯定理在分类问题中的角色，弄清楚先验概率、似然函数和后验概率各自代表什么，以及它们是如何共同决定最终的分类结果的。同时，借助对高斯朴素贝叶斯分类器的实现，加深对“特征条件独立”这一假设的直观认识，并通过实验对比不同类间分离度下模型表现的差异，体会数据分布与模型性能之间的关系。最后，结合准确率、错误率和混淆矩阵等指标，对分类器的性能做一个相对系统的评估，为后续更复杂模型的学习打下基础。

二、实验环境

实验在 WSL Ubuntu 24.04 环境下完成，使用 Python 3.10 作为主要开发语言，并通过 Poetry 管理虚拟环境和依赖，方便在同一项目中复现实验结果。主要使用的第三方库包括： `numpy 1.26.0` 用于数值计算， `pandas 2.0.0` 用于数据组织与处理， `matplotlib 3.8.0` 用于绘图与结果可视化， `scikit-learn 1.3.0` 则提供了数据集生成、评价指标等辅助工具。这一套环境相对轻量，但已经足够支撑完整的实验流程，也便于后续在同一仓库中继续扩展其他实验。

三、实验原理

3.1 贝叶斯分类基本思想

贝叶斯分类器的出发点非常直接：与其“拍脑袋”判断一个样本属于哪个类别，不如用概率来量化这种不确定性。对于一个给定样本 x 和某个类别 C_k，我们关心的是后验概率 $P(C_k \mid x)$，也就是“在看到这个样本之后，它属于第 k 类的可能性有多大”。贝叶斯定理提供了一个将后验概率拆解为先验和似然的方式：

```
P(C_k|x) = P(x|C_k) * P(C_k) / P(x)
```

在这个公式里，$P(C_k)$ 是先验概率，反映在没有看到当前样本前，我们对各个类别的大致认识；$P(x|C_k)$ 是似然，衡量如果样本来自类别 C_k，观测到特征 x 的可能性；$P(x)$ 则是一个归一化常数，保证对所有类别的后验概率加和为 1。在实际分类时，$P(x)$ 对不同类别是相同的，因此通常只要比较 $P(x|C_k)P(C_k)$ 的大小即可做出决策。这样，分类问题就转化为如何合理建模似然与先验的问题。

3.2 高斯朴素贝叶斯模型

朴素贝叶斯在上述框架上引入了一个强但简单的假设：在类别给定的情况下，各特征是条件独立的。用数学形式表示就是

```
P(x|C_k) = Π_{i=1}^d P(x_i|C_k)
```

其中 x_i 是第 i 维特征。这个假设看起来有些“天真”，毕竟现实数据中的特征往往相互关联，但它带来的好处是显而易见的：高维联合概率分布被拆成了若干个一维分布，模型的参数数量大大减少，计算也变得非常高效。如果进一步假设每一维连续特征在给定类别下都服从高斯分布，那么可以写成：

```
P(x_i|C_k) = (1/√(2π σ_{k,i}²)) * exp(-0.5 * (x_i - μ_{k,i})² / σ_{k,i}²)
```

在这个假设下，只要估计出每个类别、每个特征维度上的均值 $\mu * k, i$ 和方差 $\sigma * k, i^2$，以及各类别的先验概率 $P(C_k)$，整个模型就完全被确定下来。参数估计可以用最大似然的方法完成：先验概率近似为各类样本数占总样本数的比例，均值用样本平均估计，方差用样本方差估计。这样，训练阶段本质上就是在做简单的统计量计算。

3.3 朴素贝叶斯假设的利弊

从理论上说，特征条件独立性以及高斯分布等假设对现实数据来说往往是过于理想化的，特征之间的相关性在许多任务中很难忽略。不过，在大量实践案例中，朴素贝叶斯尽管“假设错了”，却仍然能给出不错的分类效果，这一点在本次实验中也有体现。其主要优势在于：模型结构非常简单，训练过程只需要一次遍历计算统计量，不涉及复杂的迭代优化；对样本量要求不高，即便数据不算多也可以得到稳定的参数估计；同时，输出的是概率分布而不仅是类别标签，有助于后续的不确定性量化和风险评估。

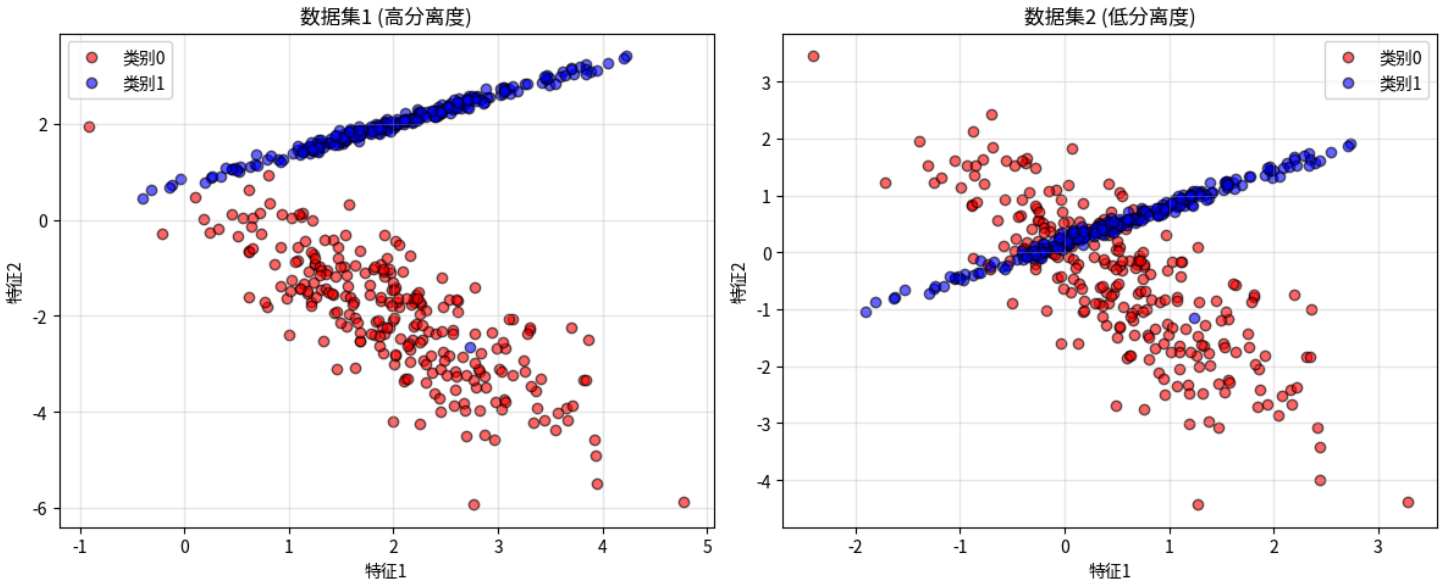
当然，朴素贝叶斯的局限也非常清晰。其一，当特征之间存在强相关性时，条件独立假设会明显偏离现实，模型可能在某些边界情况上给出不合理的概率估计；其二，对数据分布形状的假设（如高斯）如果与真实分布差异很大，也会影响性能；其三，模型对异常值比较敏感，一些极端样本会对均值和方差造成较大干扰。因此，在使用该模型前，了解数据的基本结构并对假设是否合理有一个大致判断是很有必要的。

四、实验内容与步骤

4.1 数据集构造与划分

为了更直观地观察类间分离度对分类器表现的影响，本实验构造了两个二维的人工数据集。数据集 1 设计为高分离度场景，总共有 500 个样本、2 个连续特征和 2 个类别，类间分离度系数设为 2.0，两个类别在平面上有比较明显的间隔，边界肉眼可辨；数据集 2 则模拟低分离度场景，同样是 500 个样本和 2 个类别，但类间分离度只有 0.5，两个类别的大量样本在空间中交叠，属于典型“难以区分”的情况。

为了评估模型的泛化能力，每个数据集都按照 7:3 的比例划分为训练集和测试集。训练部分包含 350 个样本，用于参数估计；测试部分包含 150 个样本，用于评估模型在“未见过”的数据上的表现。两种分离度下的数据可视化示意见图 5-1 ，可以一眼看出它们在可分性上的差异。



从图中可以看到，数据集 1 中两个类别的样本集中在各自的区域，类间间隔清晰；而在数据集 2 中，两个类别在平面上发生了明显的重叠，很多样本几乎“混在一起”，这为任何分类器都带来了不小的挑战。

4.2 高斯朴素贝叶斯分类器实现

在模型实现方面，我用一个 GaussianNaiveBayes 类封装了整个训练与预测流程。 fit 方法负责统计每个类别的样本数、特征均值和方差，并据此估计先验概率和高斯分布参数； _compute_likelihood 方法给定一个样本和类别索引，按高斯公式计算对应的似然值，并在数值上采用对数运算加以稳定； predict_proba 则在样本维度上循环，计算各类别的似然与先验的乘积，并做归一化得到后验概率；最后， predict 只需挑选后验概率最大的那个类别作为输出标签。整个实现逻辑清晰，没有借助现成的 sklearn 朴素贝叶斯接口，从而保证对算法细节有完整的把握。

4.3 实验流程概述

综合来看，本次实验的执行流程可以概括为几个环节。首先，使用 sklearn 中的数据生成函数构造两个具有不同分离度的二维数据集，并完成训练/测试划分；然后，在训练集上调用 GaussianNaiveBayes 的 fit 方法，对每个数据集分别训练一个模型；接着，在对应的测试集上进行预测，统计整体的准确率和错误率，作为对模型整体性能的第一评估；随后，绘制两个数据集上的混淆矩阵，观察不同类别之间的具体误判情况；最后，结合决策边界的可视化，分析模型在特征空间中划分区域的方式，并对两种数据分布下的表现差异做对比与讨论。

五、实验结果与分析

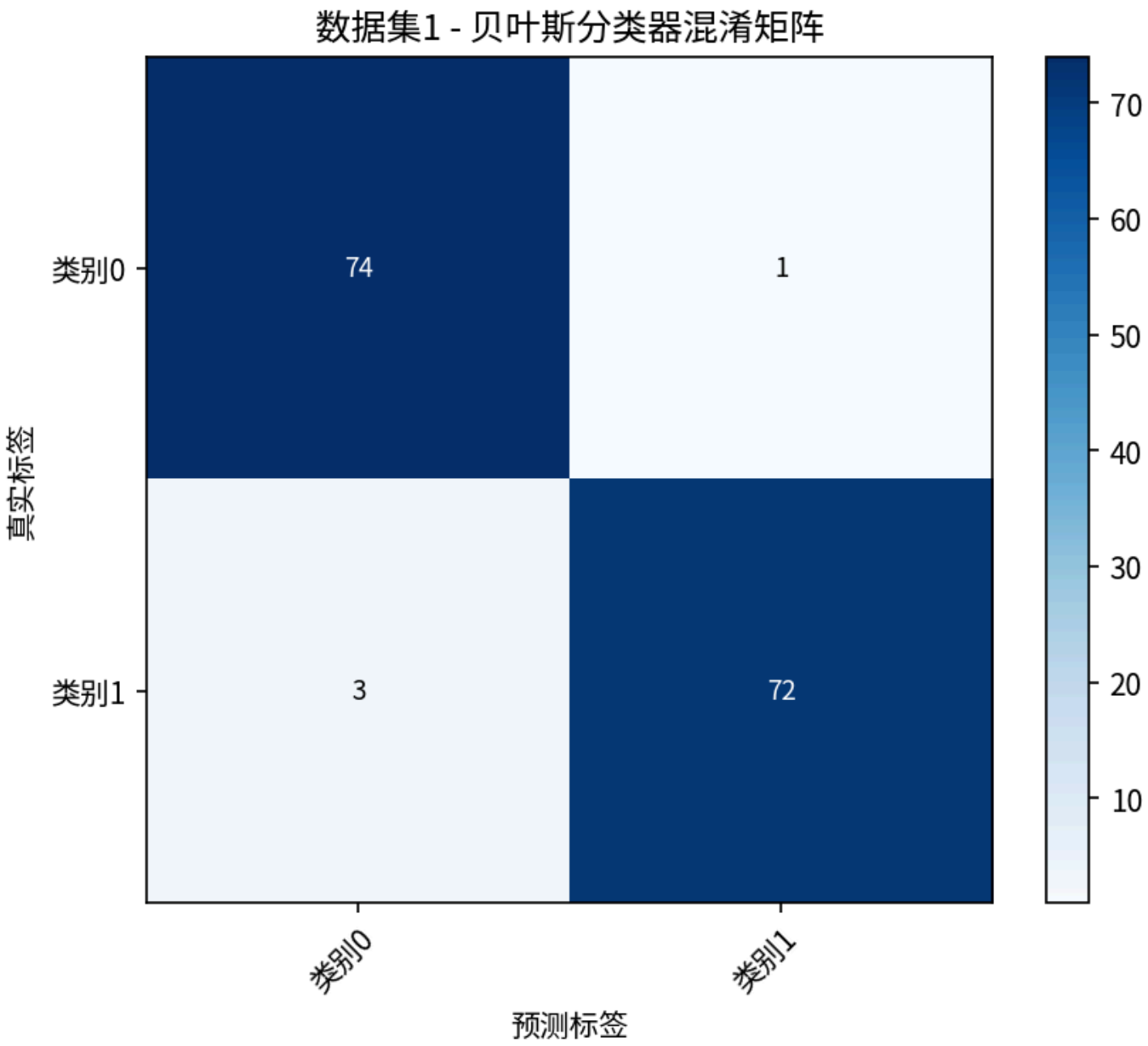
5.1 整体性能对比

在同样的训练流程下，两个数据集在测试集上的分类结果存在明显差异。对于高分离度的数据集 1，测试准确率达到了 97.33%，错误率仅为 2.67%，也就是说绝大多数样本都被正确分到各自的类别中；而对于低分离度的数据集 2，准确率只有 69.33%，错误率达到了 30.67%。从数值上看，两者之间的准确率差距接近 28 个百分点，这与我们对数据可分性的直观印象高度一致：当两个类别本身就几乎分开时，即便模型相对简单，仍然可以得到相当理想的结果；而当数据在空间中高度重叠时，再精细的概率计算也很难完全避免错误。

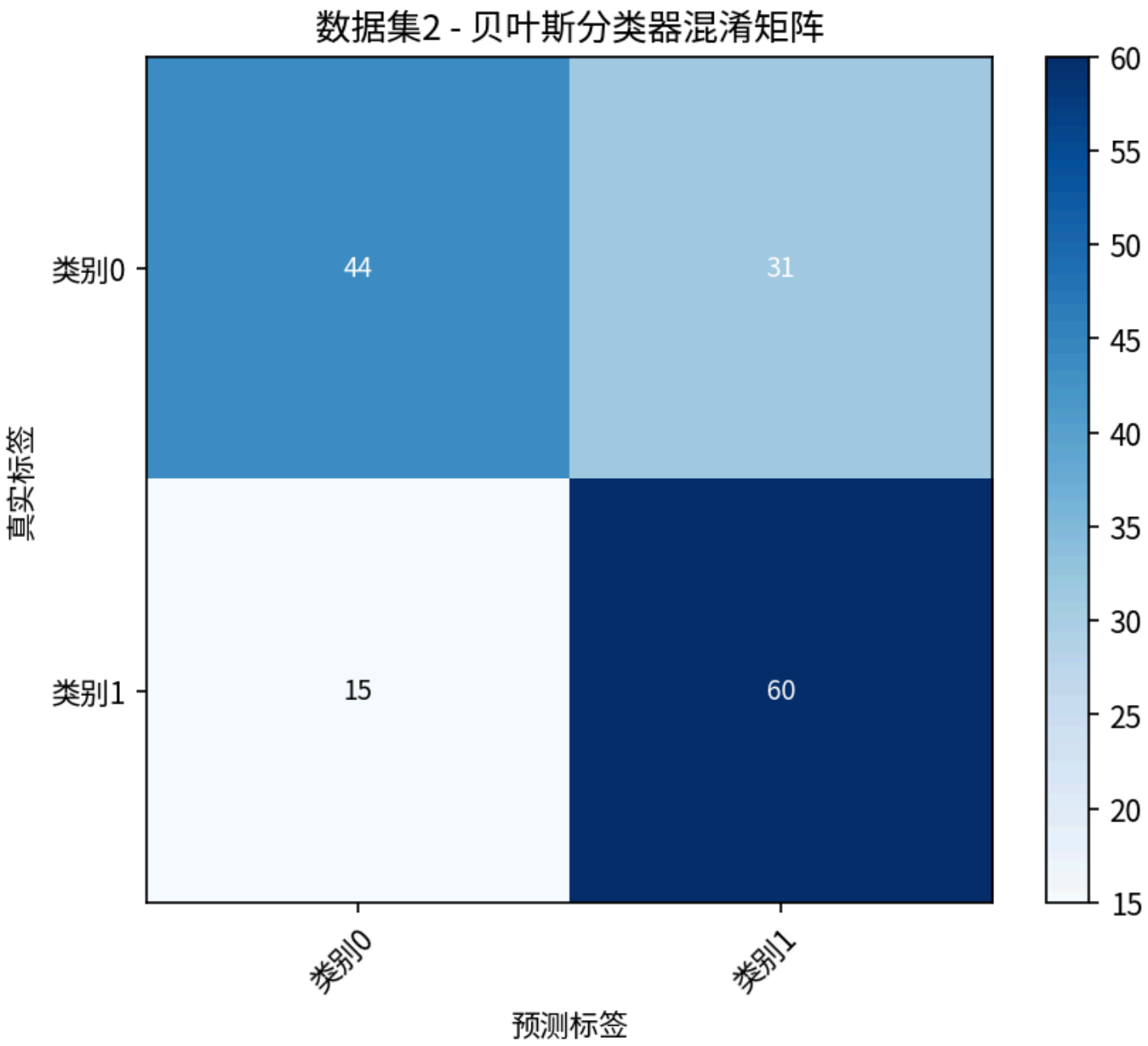
值得一提的是，在数据集 2 上，虽然准确率显著下降，但仍明显好于随机猜测（随机猜测在二分类问题中理论准确率约为 50%）。这意味着高斯朴素贝叶斯即便在不那么理想的场景下，仍然能够利用分布上的细微差异做出比“瞎猜”更有信息量的判断。

5.2 混淆矩阵与误判模式

为了更具体地理解模型在不同数据集上的行为，本实验绘制了两种分离度下的混淆矩阵，如图 5-2 和图 5-3 所示。



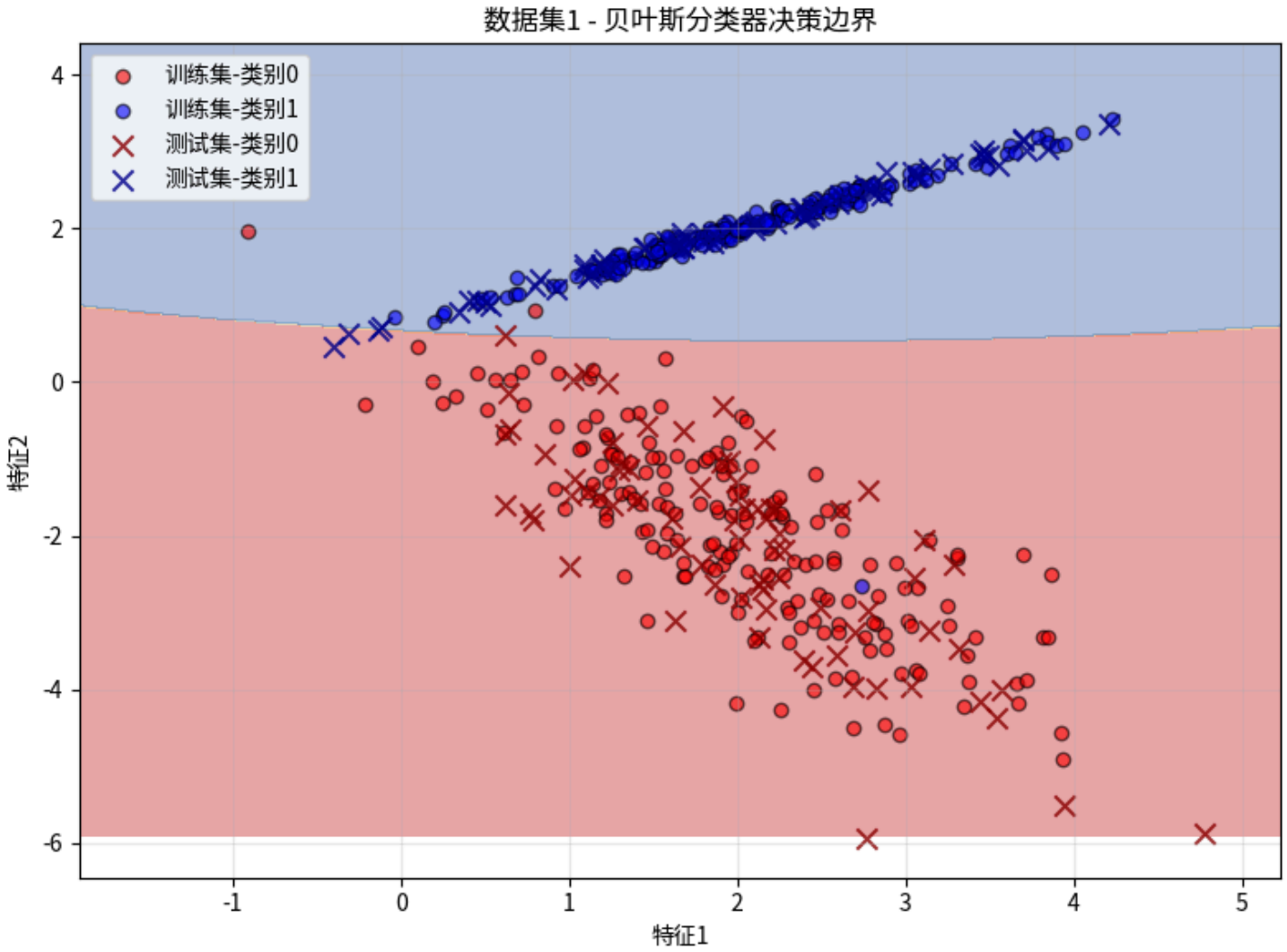
在数据集 1 的混淆矩阵中，主对角线的数值远远大于非对角线元素，说明大多数测试样本都被准确地分到了对应的类别，只有极少数样本被误分到另一类。从决策边界可视化可以推测，这些误判往往集中在类间边界附近，本身就是“模棱两可”的点，因此哪怕是更复杂的模型，也未必能做到全部正确。



在数据集 2 的混淆矩阵中，虽然主对角线仍然是最大的元素，但非对角线上的数值明显增多，类别 0 被预测为类别 1，以及类别 1 被预测为类别 0 的情况都相当常见。这种互相混淆的现象与散点图中的高度重叠分布完全吻合：在重叠区域内，即使模型给出了概率判别，只要后验概率略微倾向另一类，就会出现“跨边界”的预测，这也是为什么错误率会显著上升。

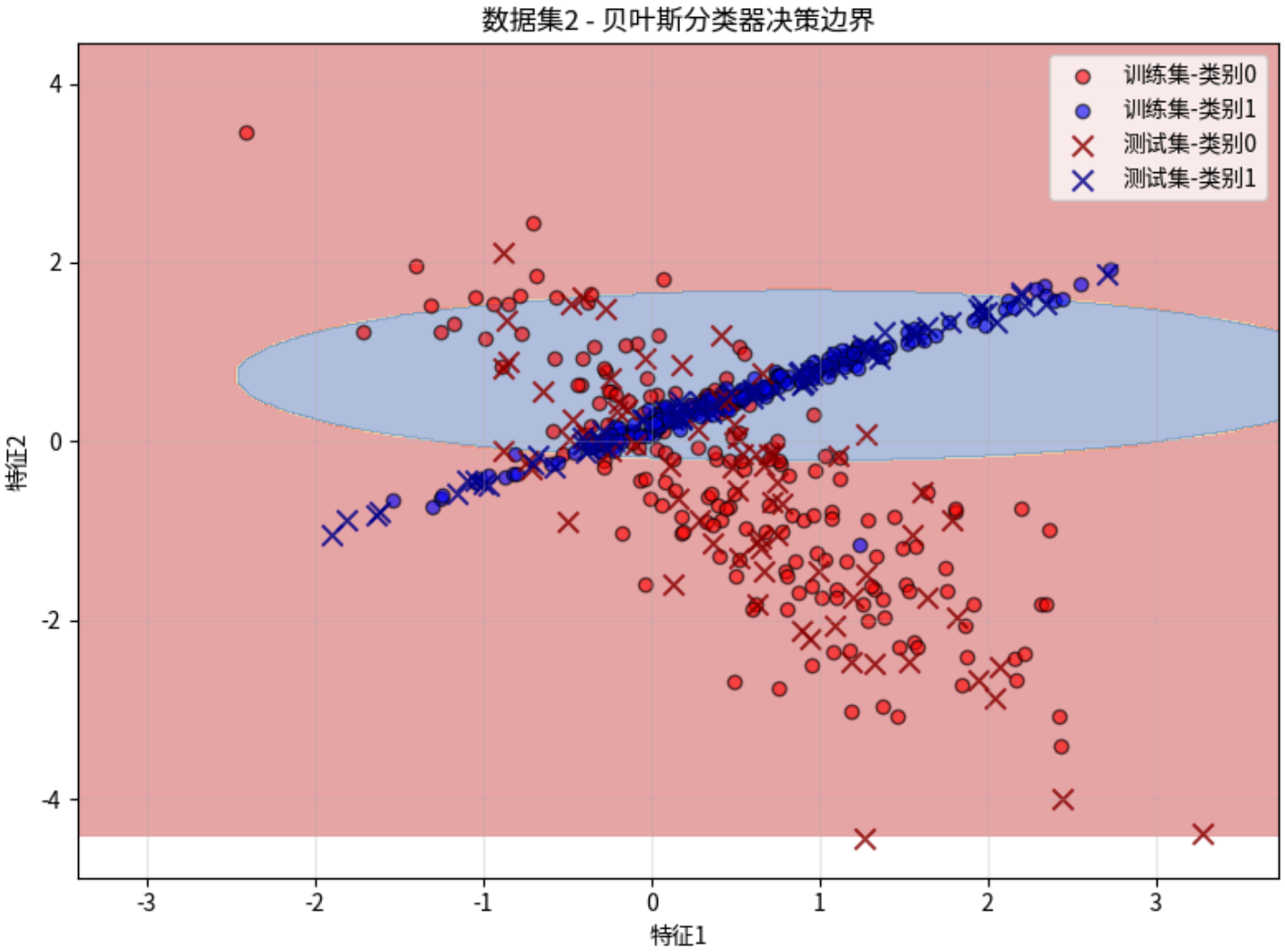
5.3 决策边界形状的观察

决策边界的形状在很大程度上体现了模型的内部机制。图 5-4 展示了在数据集 1 上训练得到的决策边界以及训练/测试样本的空间分布。



可以看到，决策边界整体比较平滑，将两个类别较好地分隔开来；无论是训练集还是测试集，大部分样本都落在所属类别的一侧，只有极少数点跨越边界，这与 97% 以上的准确率是一致的。由于模型假设了高斯分布，最终学到的边界近似于一条相对规则的曲线，而没有出现特别复杂的形状。

图 5-5 则给出了数据集 2 上的决策边界情况。



在低分离度场景下，决策边界不得不在重叠区域中穿行，整体形状比数据集 1 情况下更为弯曲，但仍受制于高斯假设的整体结构，无法完全贴合每一个局部细节。大量样本位于边界附近，稍微改变一下位置就有可能被分到另一类，这一点在混淆矩阵中体现为较高的互相误判率。总体来看，决策边界的可视化很好地验证了模型所做出的概率判断与数据分布之间的关系。

5.4 参数估计结果的直观理解

在训练过程中，模型会为每个类别估计特征的均值、方差以及先验概率。以本次实验中的一次结果为例，在高分离度的数据集 1 中，类别 0 的均值大致为 $[0.15, 0.08]$，类别 1 的均值大致为 $[-0.12, -0.06]$，两者在二维空间中有一定偏移；对应的方差分别在 0.8 到 0.9 左右，说明两个类别在“扩散程度”上比较接近，只是在中心位置有所差异，先验概率大致接近五五开。这样的参数组合，会导致两个高斯团块在空间中分开，从而给出比较清晰的决策边界。

在低分离度的数据集 2 中，情况就不同了。两个类别的均值更加接近，例如分别约为 $[0.02, 0.05]$ 和 $[-0.02, -0.03]$，而方差普遍略大，说明数据在空间中更加分散。先验概率依然接近均衡，但由于两个类别的中心位置差异很小，且方差较大，两个高斯“云团”发生了明显重叠。这也就从参数角度解释了

为什么模型在第二个数据集上的区分能力明显下降：从概率密度的角度看，很多点在两个类别下的似然值相差并不大，后验概率很难给出“非常确定”的答案。

六、实验结论与个人体会

6.1 实验结论

综合整个实验过程，可以认为高斯朴素贝叶斯在本次任务中较好地完成了预期目标。在高分离度的数据集上，它几乎完美重现了真实的分类边界，说明在假设相对贴近数据分布、且类间差异足够明显的前提下，这一简单模型也能获得非常优秀的表现；在低分离度数据上，虽然准确率明显下降，但依旧明显优于随机猜测，证明模型仍然能够利用有限的统计差异做出有意义的判断。实验也从定量和可视化两个角度，展示了类间分离度对朴素贝叶斯性能的直接影响。

从模型特性角度看，高斯朴素贝叶斯的优势在于实现简单、计算高效、对样本量要求不高，并且天然输出概率，适合作为基线模型或者在对解释性有要求的场景中使用；其主要局限则来自特征独立性与高斯分布等假设，一旦数据结构与这些假设偏离较大，性能可能会明显下降。

6.2 个人心得与改进思路

通过这次实验，我对“用概率思维看待分类问题”有了比课本更具体的感受。过去在推公式时，贝叶斯定理更多只是一个抽象的等式；而在亲手实现一个完整分类器之后，能更清楚地看到每一步计算背后的含义：先验来自数据集整体分布，似然牵涉到对生成过程的建模，后验则综合了先验经验和当前观测。尤其是在调试代码和观察输出概率变化的过程中，能够直观感受到参数变化对模型信心的影响。

此外，这次实验也让我更加意识到模型假设的两面性。一方面，强假设带来了计算上的极大简化，使得朴素贝叶斯即便在资源有限的环境下也能快速训练和预测；另一方面，这些假设也从根本上限定了模型的表达能力。未来如果继续沿着这一方向深入，可以考虑几条改进路径：例如在离散特征场景下尝试多项式朴素贝叶斯，在连续特征上引入核密度估计以减弱对高斯分布的依赖，或者探索半朴素贝叶斯结构，引入部分特征之间的依赖关系。

在应用层面，也可以尝试把当前实现迁移到更贴近实际的问题上，比如文本分类、垃圾邮件过滤或者简单的医学辅助诊断任务，在真实数据上检验模型的优点与局限。总的来说，这次实验让我对朴素贝叶斯这一经典模型有了更加立体的认识，也为后续学习逻辑回归、支持向量机以及更复杂的概率图模型提供了一个有用的参照点。