实验一：多元正态分布下的贝叶斯决策

一、实验任务

分别生成两类数据集，每类数据集的样本是3000个，有2个特征，服从不同的二维高斯分布。对其随机划分成训练集（4000个样本）和测试集（2000个样本），在不同的前提下分别进行贝叶斯决策。

（1） 每类的协方差矩阵相等，且2个特征相互独立（对角线矩阵），类间均值不同。

（2） 每类的协方差矩阵相等，且2个特征不相互独立（非对角线矩阵），类间均值不同。

（3） 每类的协方差矩阵不相等，类间均值不同。

要求：

（1） 画出两个不同高斯分布的等概率密度线以及概率密度图；

（2） 计算测试集的准确率，并画出决策面；

二、算法原理概述

若变量x服从多元正态分布$\mathcal{N}(\mu,\Sigma)$，那么

$$p(x)=\frac{1}{(2\pi)^{\frac{d}{2}}(\det\Sigma)^{\frac{1}{2}}}\exp\{-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\}$$

判别函数（对数形式）为：

$$\begin{aligned}g_i(x)&=\ln(p(x|\omega_i)p(\omega_i))=\ln(p(x|\omega_i))+\ln p(\omega_i)\\&=-\frac{1}{2}(x-\mu_i)^T\Sigma_i^{-1}(x-\mu_i)+\ln\frac{1}{(2\pi)^{\frac{d}{2}}\det\Sigma_i^{\frac{1}{2}}}+\ln p(\omega_i)\\&=-\frac{1}{2}(x-\mu_i)^T\Sigma_i^{-1}(x-\mu_i)-\frac{d}{2}\ln 2\pi-\frac{1}{2}\ln\det\Sigma_i+\ln p(\omega_i)\end{aligned}$$

决策面方程为

$$g_i(x)=g_j(x)\rightarrow$$
$$-\frac{1}{2}[(x-\mu_i)^T\Sigma_i^{-1}(x-\mu_i)-(x-\mu_j)^T\Sigma_j^{-1}(x-\mu_j)]-\frac{1}{2}\ln\frac{\det\Sigma_i}{\det\Sigma_j}+\ln\frac{p(\omega_i)}{p(\omega_j)}=0$$

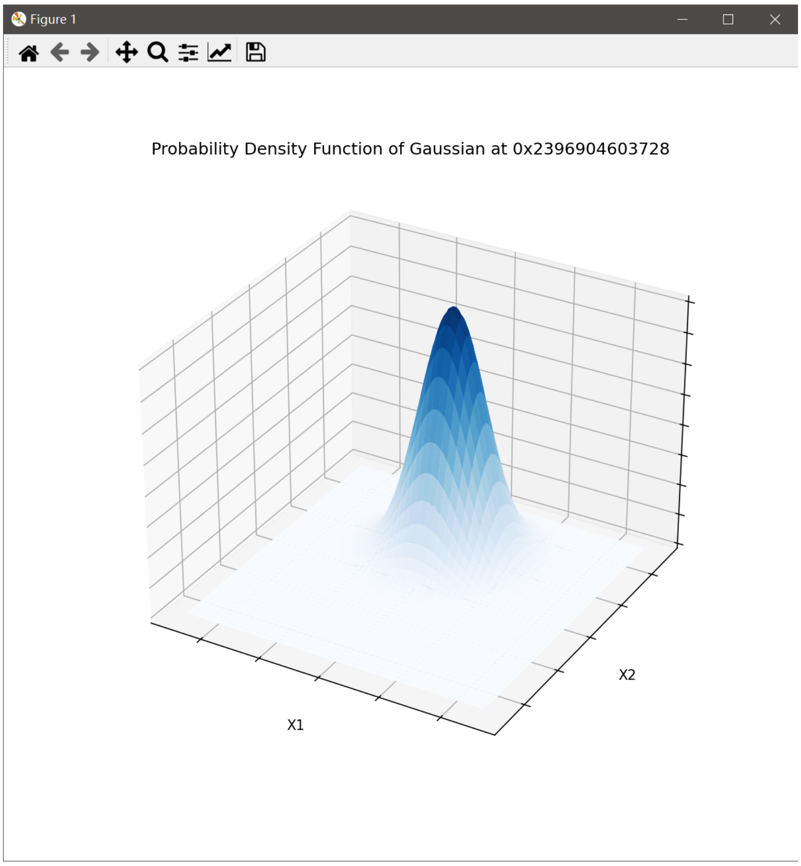
三、测试步骤

直接运行`util.py`文件即可，运行结果为所有的图例。

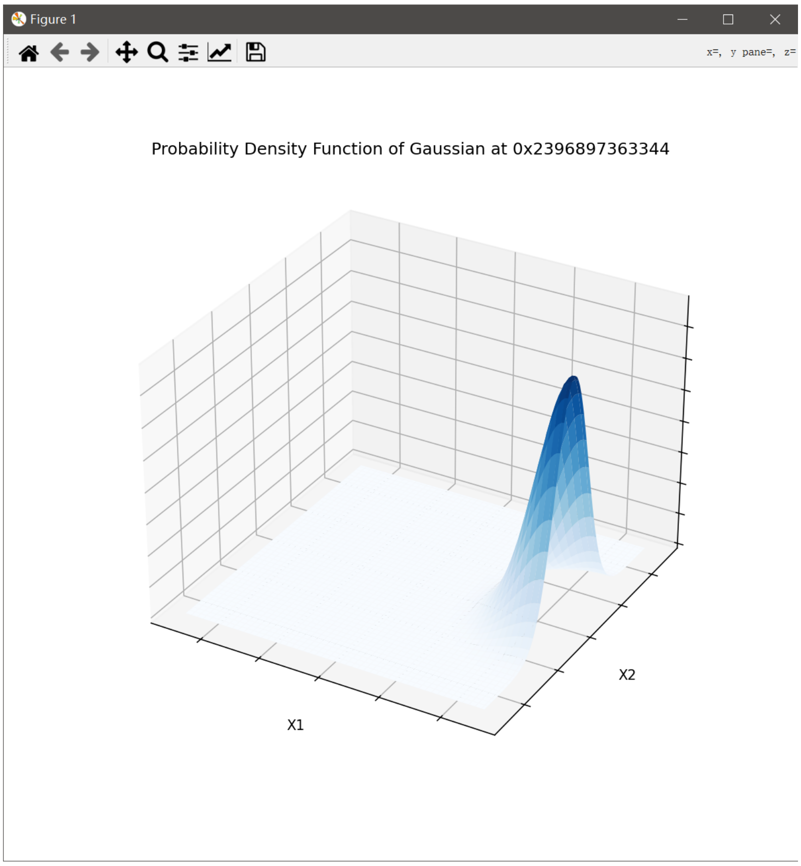
四、实验结果与讨论

1. 当每类的协方差矩阵都为$\begin{bmatrix}1 & 0\\0 & 1\end{bmatrix}$，第一类多元正态分布的均值为$[0 \quad 2]$，第二类多元正态分布的均值为$[5 \quad 0]$时，运行 `util.py` 文件，程序依次弹出 两类高斯分布的**概率密度函数图**，两类高斯分布的**等概率密度线**，两个高斯分布的**决策面**，最后在终端打印预测准确率。

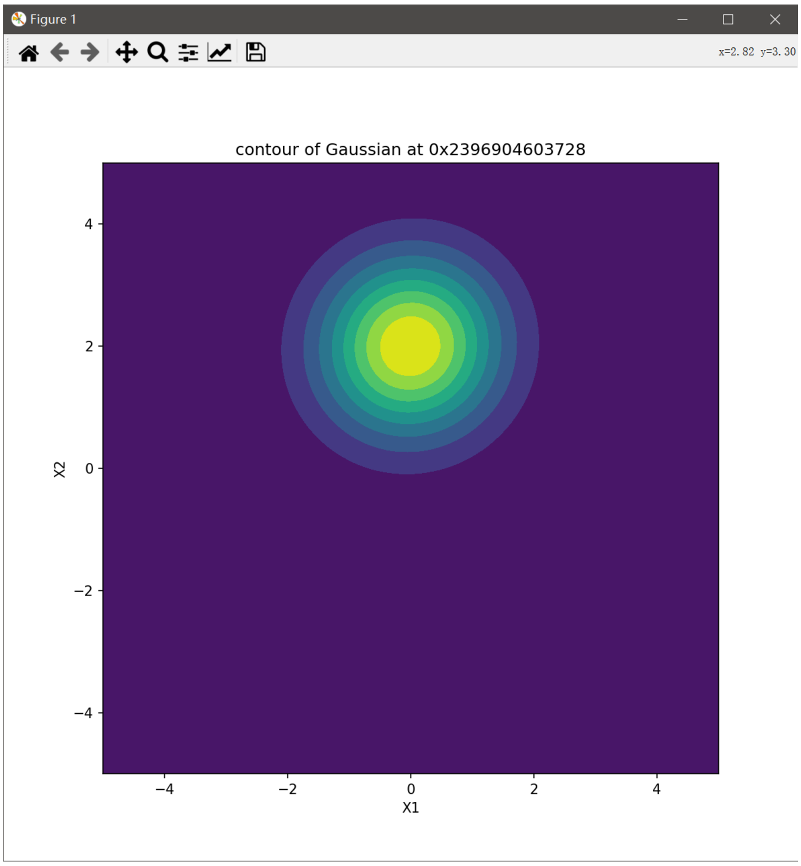
   第一类正态分布的概率密度函数图



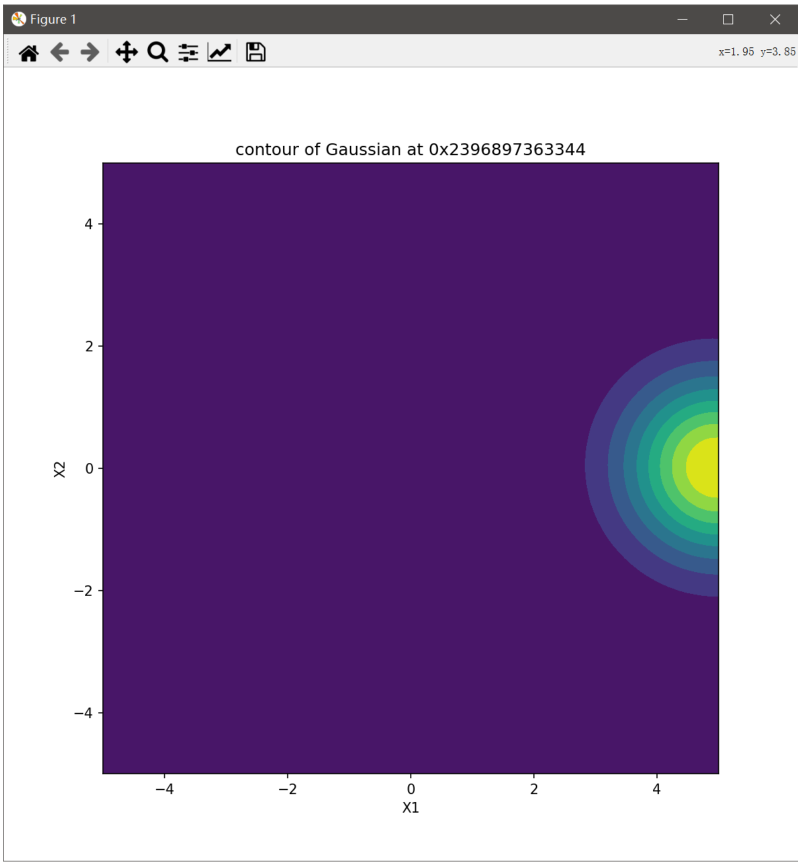
   第二类正态分布的概率密度函数图



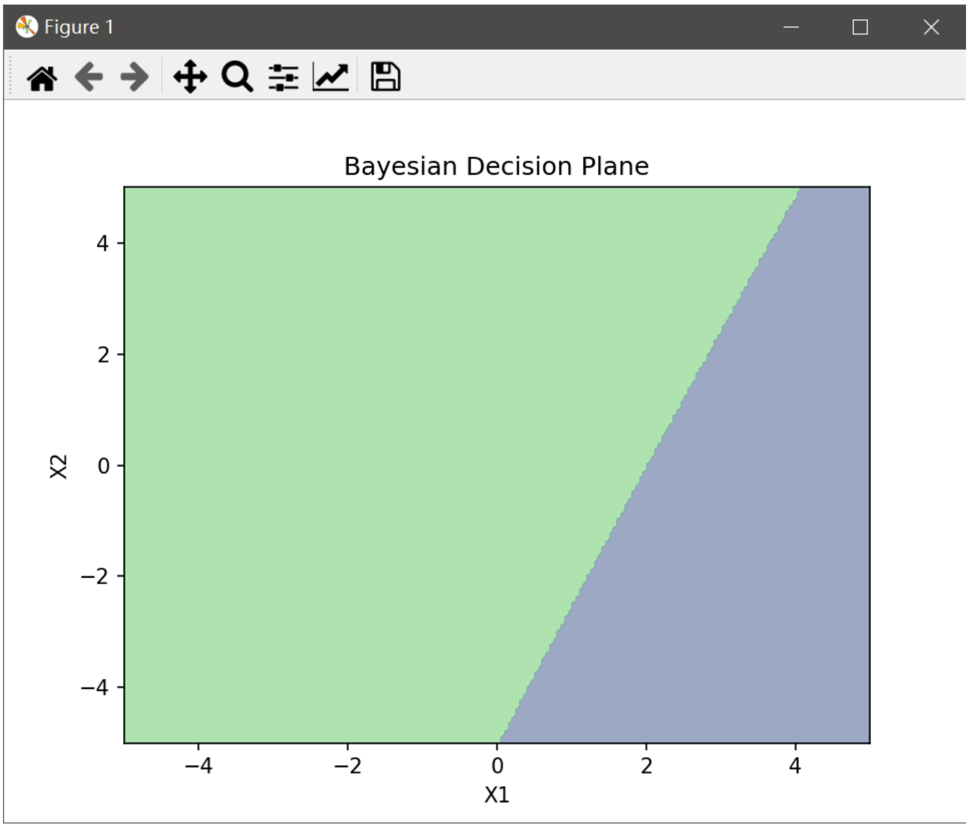
   第一类正态分布的等概率密度图



第二类正态分布的等概率密度图



贝叶斯决策面：



模型预测准确率

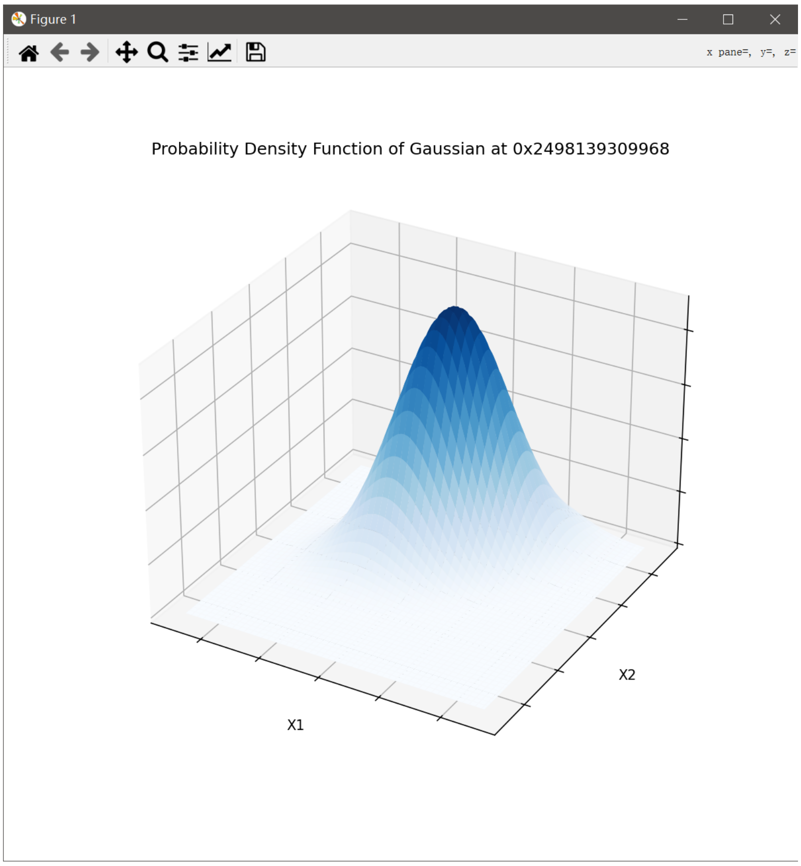
```
D:\anaconda\envs\pr\python.exe "E:\23FA\F
prior is 0.497
accuracy is 0.9955

Process finished with exit code 0
```

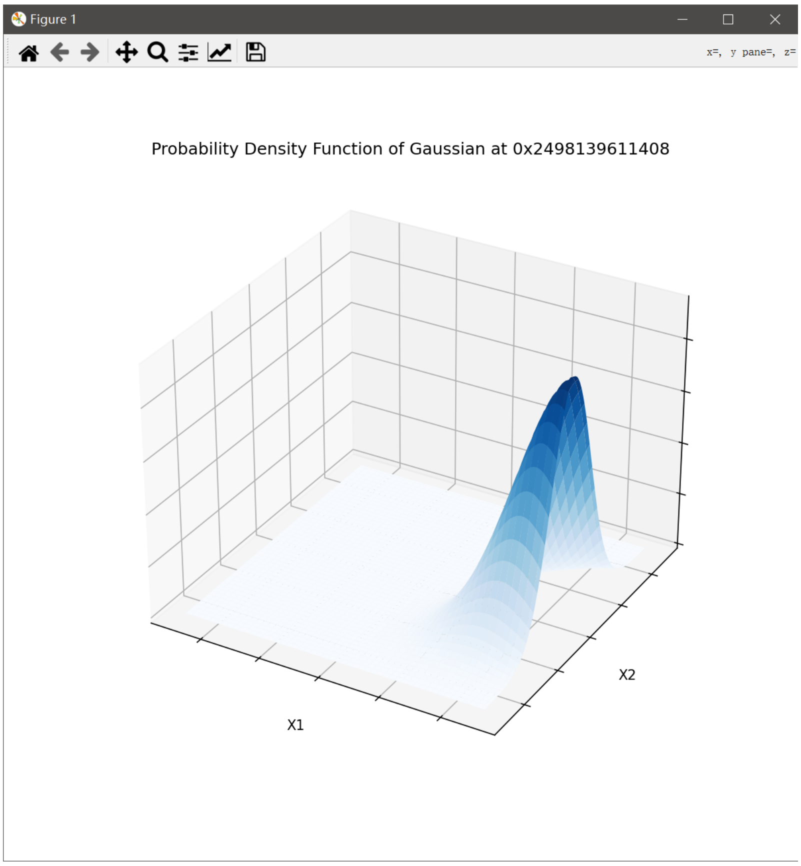
根据正态分布时的贝叶斯决策，当各类协方差矩阵相等，且各特征独立、方差相等：$\Sigma_i = \sigma^2 I$ 时，决策曲线时一个线性判别函数，向先验概率小的方向偏移。由上图可见，第1类正态分布的先验概率为0.497，因此线性判别函数会像第1类正态分布的中心偏移.

2. 当各类协方差矩阵取$\begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$时，第一类多元正态分布的均值为$\begin{bmatrix} 0 & 2 \end{bmatrix}$，第二类多元正态分布的均值为$\begin{bmatrix} 5 & 0 \end{bmatrix}$时，运行 `util.py` 文件，程序依次弹出 两类高斯分布的**概率密度函数图**，两类高斯分布的**等概率密度线**，两个高斯分布的**决策面**，最后在终端打印预测准确率。

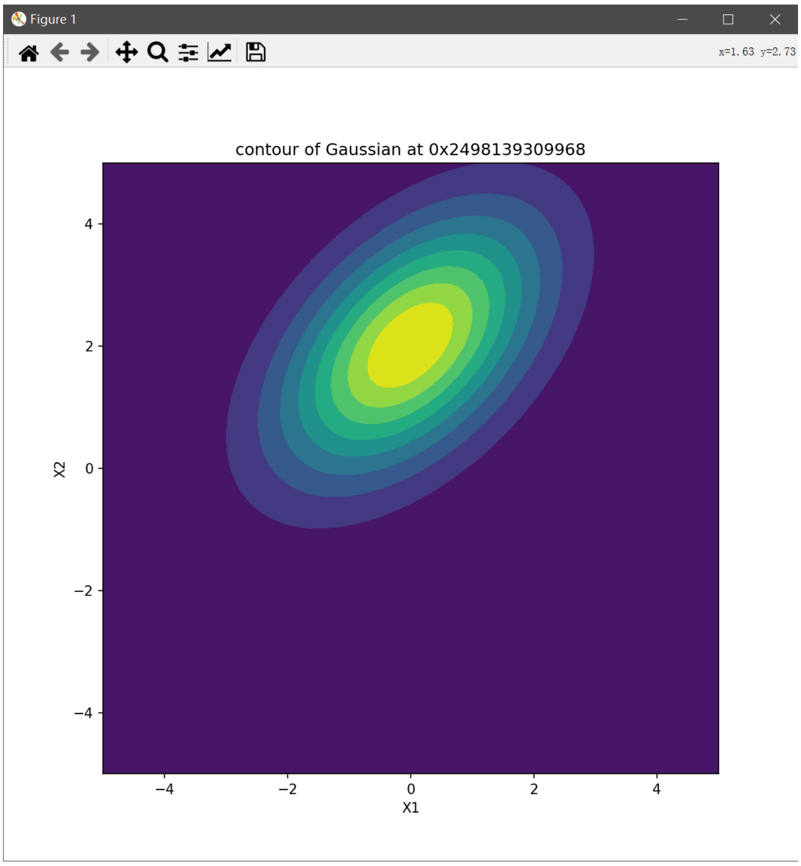
第一类正态分布的概率密度函数图



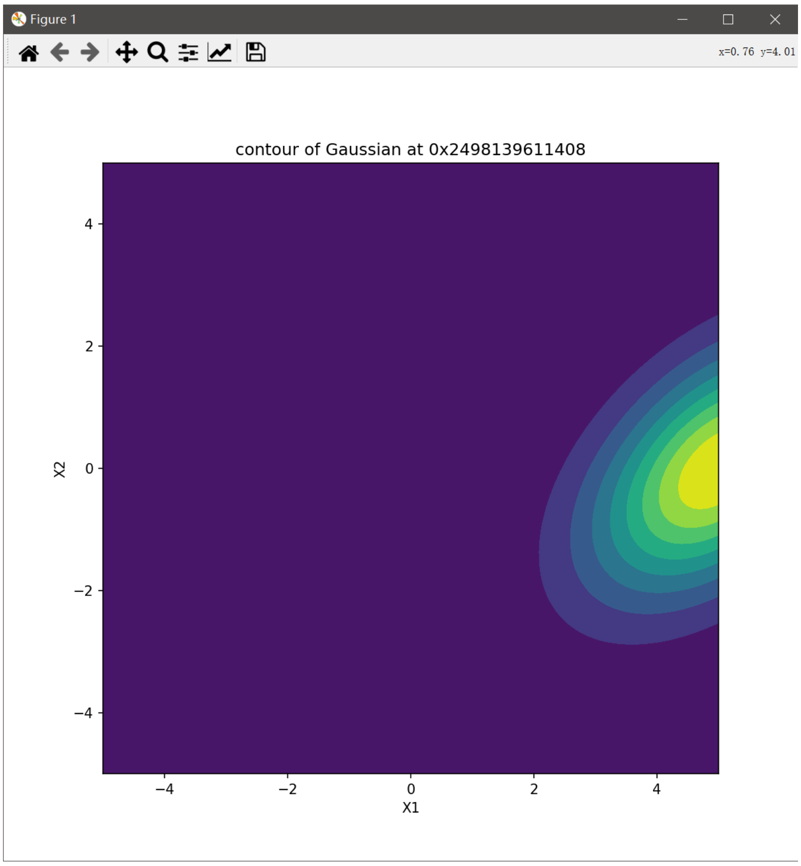
第二类正态分布的概率密度函数图



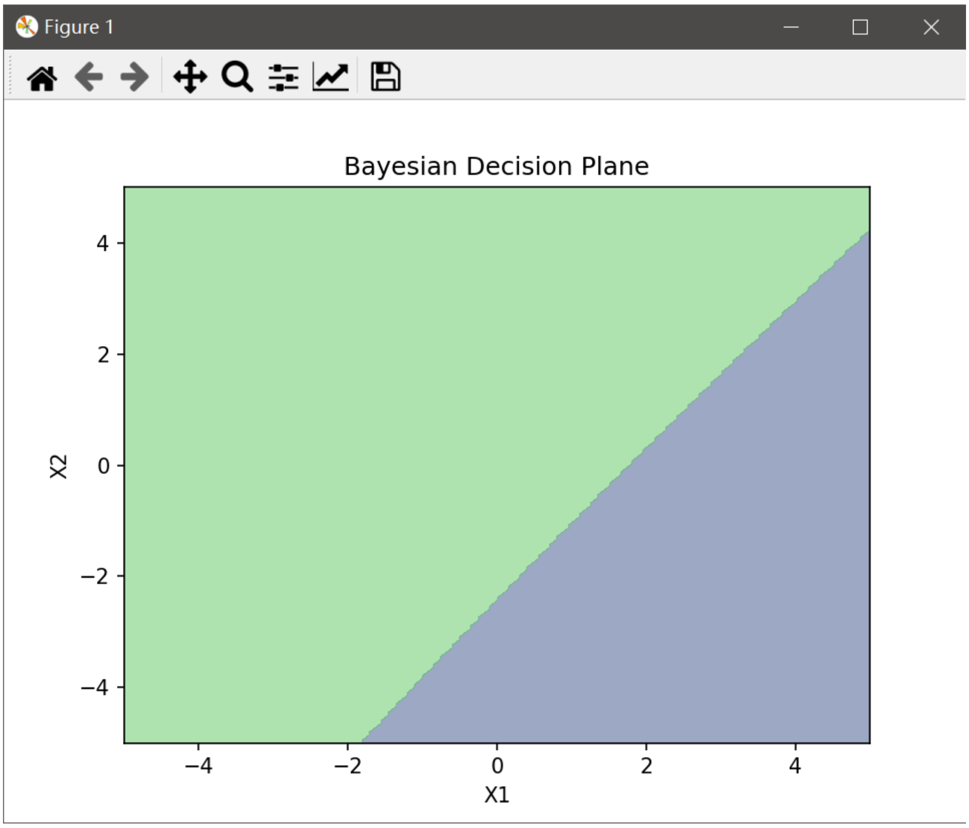
第一类正态分布的等概率密度图



第二类正态分布的等概率密度图



贝叶斯决策面



先验概率和预测准确率

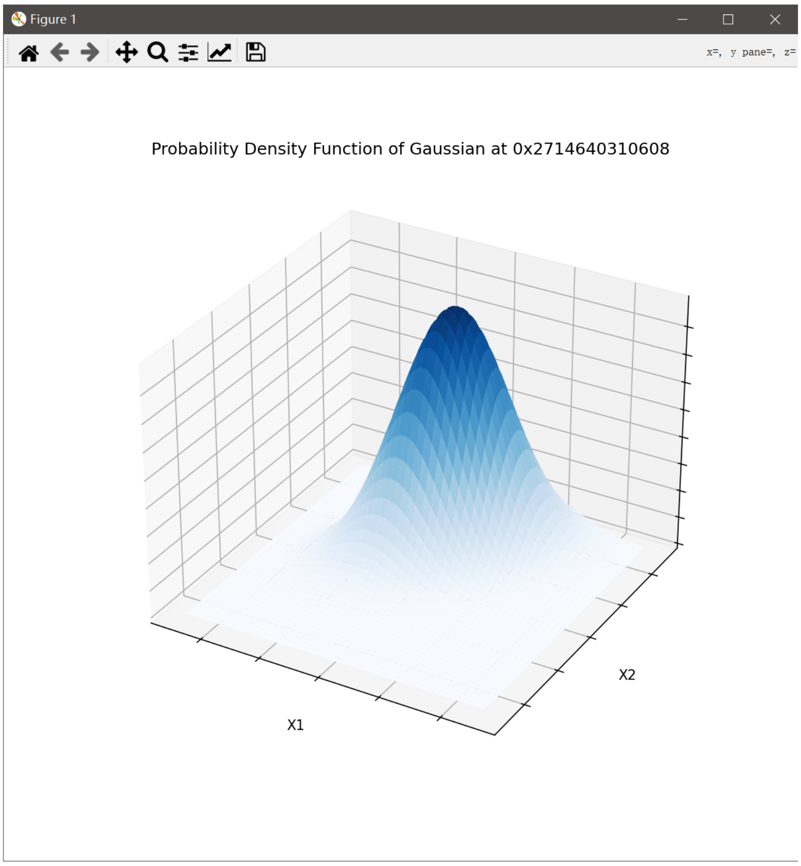
```
D:\anaconda\envs\pr\python.exe "E:\23FA\Pattern Recognition
prior is 0.511
accuracy is 0.995

Process finished with exit code 0
```

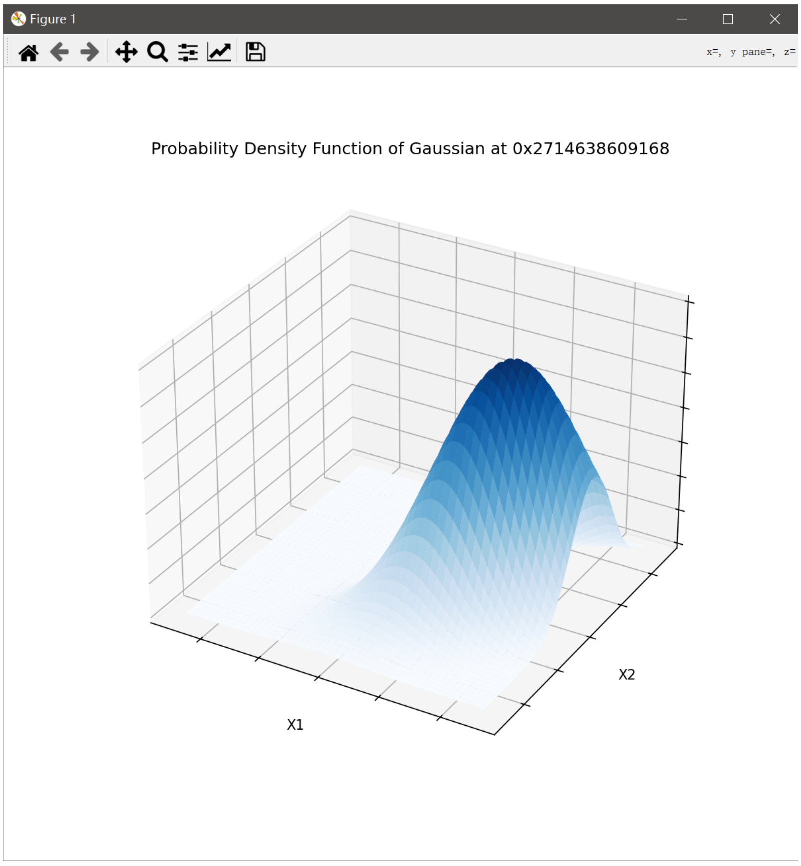
根据正态分布时的贝叶斯决策，当各类协方差矩阵相等，$\Sigma_i = \Sigma$ 时，决策曲线时一个线性判别函数，向先验概率小的方向偏移。由上图可见，第1类正态分布的先验概率为0.511，因此线性判别函数会像第二类正态分布的中心偏移.

3. 当第一类正态分布协方差矩阵取$\begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$时，第一类多元正态分布的均值为$[0 \quad 2]$，当第二类正态分布协方差矩阵取$\begin{bmatrix} 3 & 1 \\ 1 & 3 \end{bmatrix}$第二类多元正态分布的均值为$[3 \quad 0]$时，运行 `util.py` 文件，程序依次弹出 两类高斯分布的**概率密度函数图**，两类高斯分布的**等概率密度线**，两个高斯分布的**决策面**，最后在终端打印预测准确率。

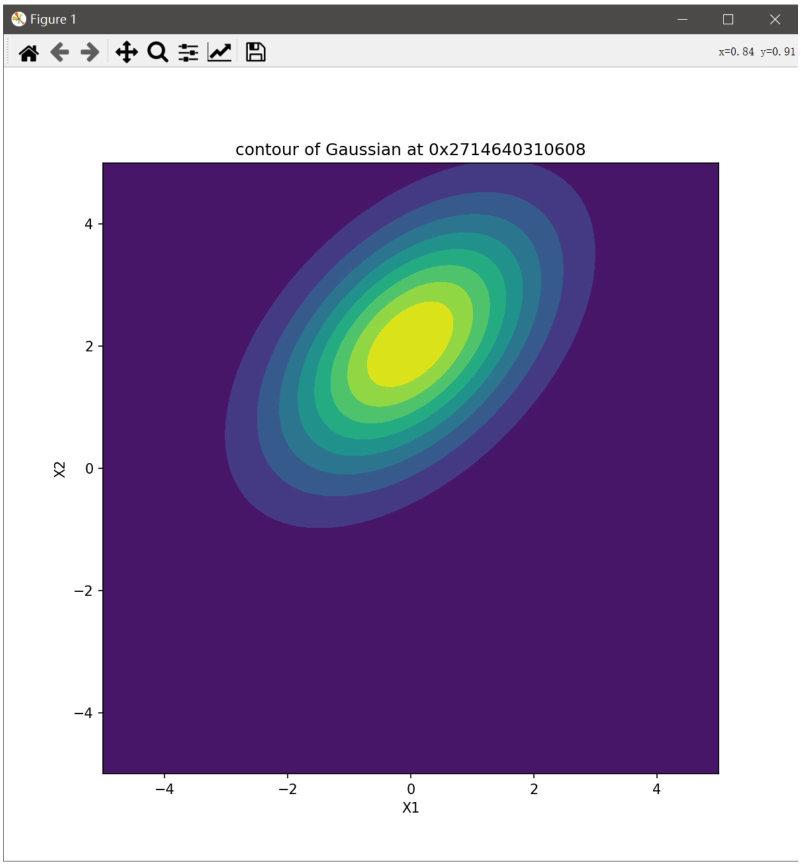
第一类正态分布的概率密度函数图



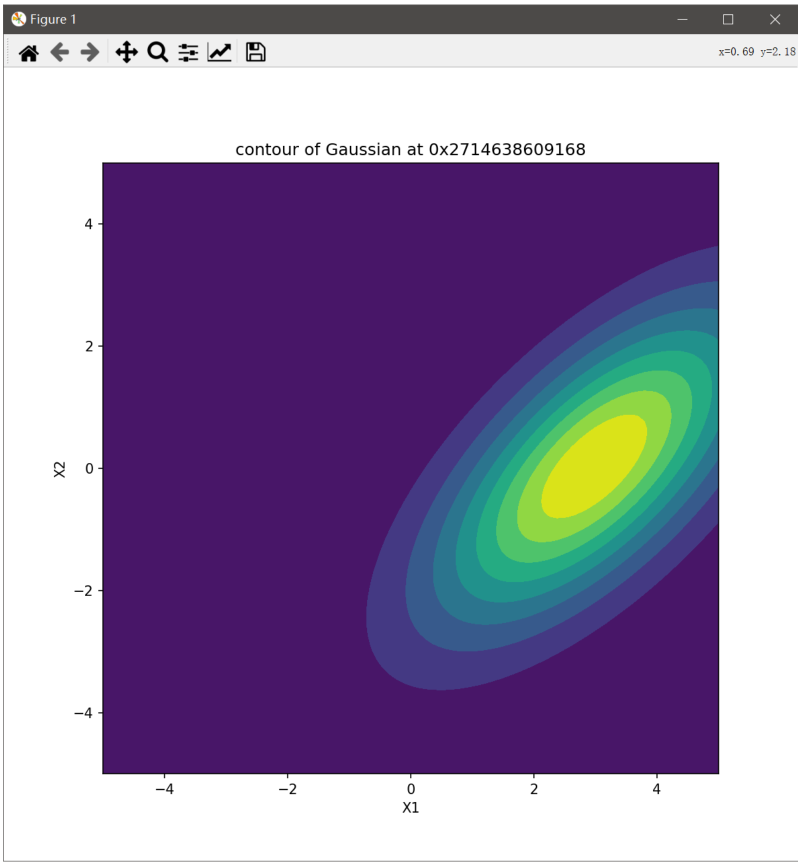
第二类正态分布的概率密度函数图



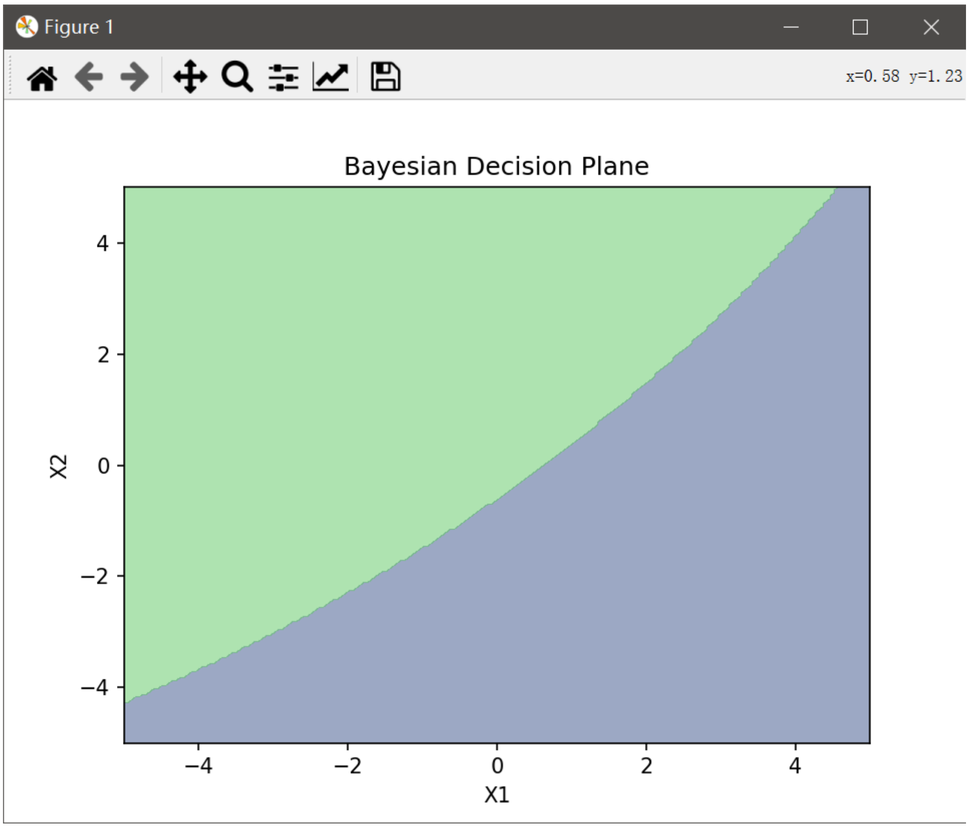
第一类正态分布的等概率密度图



第二类正态分布的等概率密度图



贝叶斯决策面



先验概率和预测准确率

```
D:\anaconda\envs\pr\python.exe "E:\23FA\Pattern Reco
prior is 0.507
accuracy is 0.9615

Process finished with exit code 0
```

此时决策面是曲线。

五、实验总结

1. 本次实验中我自定义实现了一个`Gaussian`多元高斯分布类，通过`pdf`，`loglikelihood`，`plot_pdf`，`plot_identical_pdf`等方法实现了计算pdf，对数似然，绘制pdf和绘制等概率密度图等功能。
2. 代码性能可优化的部分在于将`pdf`和`loglikelihood`等方法**向量化**来加速程序运行。其根本方法在于将**二次型的计算向量化**。x^TAx的计算时x^T是一个列向量，二次型计算得到一个scalar值。如果此时$x^T = X, X \in \mathbb{R}^{(1000,2)}$，那么$XAX^T$的计算得到一个$1000 \times 1000$的矩阵，但我们需要的是一个$(1000, 1)$的列向量。~~思考如何优化此部分。~~
3. 代码逻辑上可优化的部分在于matplotlib包中`Axes`对象的优化。当前版本的代码中`Gaussian`类的每一个方法都新建了一个figure对象，导致在程序运行时出现的窗口太多。可以尝试将pdf和contour绘制在一个figure对象中。可以通过在`Gaussian`类中声明一个 ax **成员变量**实现。每个方法实现对 ax 的修改。通过添加此成员变量也能增加`Gaussian.plot_pdf()`和`Gaussian.plot_identical_pdf()`方法中的代码复用。
4. 程序在`util.py`文件中实现了`gen_dataset()`，`split_dataset()`，`fit()`，`bayesian_decision()`，`plot_desicion_hyperplane()`等方法，实现了数据集的生成和划分，模型的拟合，贝叶斯决策过程，绘制贝叶斯决策面等功能。在主函数中只需要更改生成数据集的`Gaussian`对象即可（取消和添加注释），即可运行程序。

5. 学习和掌握了利用Matplotlib绘制3d图像，根据matplotlib官方文档，通过`numpy.meshgrid()` 得到x，y坐标，计算z坐标在通过`plot_surface()`或`plot_contourf()`等方法绘制。注意这里x, y, z都是2-d的向量。