线性分类器（逻辑回归 & GDA 高斯判别分析）

在这个问题中，我们将讨论迄今为止课上涉及到的两种概率线性分类器。第一种是判别式线性分类器：逻辑回归，第二种是生成式线性分类器：高斯判别分析（GDA）。这两种算法都能找到将数据分为两类的线性决策边界，但各自具有不同的假设条件。在这个问题中，我们将深入了解这两种算法的异同，以及它们的优缺点。

对于这个问题，我们考虑以下两个数据集：

i. data/ds1_{train,valid}.csv

ii. data/ds2_{train,valid}.csv

每个文件包含 m 个示例，每行表示一个示例 $(x^{(i)}, y^{(i)})$。其中，第 i 行包含 $x_0^{(i)} \in R$，$x_1^{(i)} \in R$ 和 $y^{(i)} \in \{0,1\}$ 三列。在下列问题中，我们将使用逻辑回归和高斯判别分析（GDA）对这两个数据集进行二元分类。

1. 在课上，我们学习了逻辑回归的平均经验损失：

$$J(\theta) = -\frac{1}{m}\sum_{i=1}^{m} y^{(i)}\log(h_\theta(x^{(i)})) + (1-y^{(i)})\log(1-h_\theta(x^{(i)})),$$

其中，$y^{(i)} \in \{0,1\}$，$h_\theta(x) = g(\theta^T x)$，$g(z) = 1/(1+e^{-z})$。
求该函数的 Hessian H，并证明对于任意向量 z，以下条件成立：

$$z^T H z \geq 0$$

提示：可以先证明 $\sum_i \sum_j z_i x_i x_j z_j = (x^T z)^2 \geq 0$，$g'(z) = g(z)(1-g(z))$。
备注：这是证明矩阵 H 是正半定矩阵的标准方法之一，记作 "H ≽ 0"。这意味着 J 是凸的，除了全局最小值外，没有局部最小值。当然也可以选择使用其他方法来证明 H ≽ 0。

2. 【代码题】根据 src/p01b_logreg.py 中的要求，使用牛顿法训练逻辑回归分类器。从 $\theta = \vec{0}$ 开始，使用牛顿法，直到 θ 的更新很小为止。具体来说，训练到第 k 轮，使得 $\|\theta_k - \theta_{k-1}\|_1 < \varepsilon$，$\varepsilon = 1 \times 10^{-5}$。请确保将模型预测结果写入代码中所指定的文件中。

3. 在 GDA 中，我们用以下方程对(x, y)的联合分布进行建模：

$$\begin{aligned}
p(y) &= \begin{cases} \phi & \text{if } y = 1 \\ 1-\phi & \text{if } y = 0 \end{cases} \\
p(x|y=0) &= \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu_0)^T \Sigma^{-1}(x-\mu_0)\right) \\
p(x|y=1) &= \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu_1)^T \Sigma^{-1}(x-\mu_1)\right)
\end{aligned}$$

其中，$\phi, \mu_0, \mu_1, \Sigma$ 为模型参数。
假设我们已经对参数 $\phi, \mu_0, \mu_1, \Sigma$ 进行了拟合，现在想对一个新的数据点 x 所对应的 y 进行预测。为了证明 GDA 所产生的分类器具有线性判定边界，试说明后验分布可以写成：

$$p(y=1 \mid x; \phi, \mu_0, \mu_1, \Sigma) = \frac{1}{1+\exp(-(\theta^T x + \theta_0))}$$

其中，$\theta \in R^n$，$\theta_0 \in R$，且为 $\phi, \mu_0, \mu_1, \Sigma$ 的函数。

4. 对这一小问，假设 n（x 的维数）为 1，那么 $\Sigma = [\sigma^2]$ 为实数，并且 Σ 的行列式可以由 $|\Sigma| = \sigma^2$ 得到。基于给定的数据集，可以得到参数 $\phi, \mu_0, \mu_1, \Sigma$ 的最大似然估计值为：

$$
\begin{aligned}
\phi &= \frac{1}{m} \sum_{i=1}^{m} 1\{y^{(i)} = 1\} \\
\mu_0 &= \frac{\sum_{i=1}^{m} 1\{y^{(i)} = 0\} x^{(i)}}{\sum_{i=1}^{m} 1\{y^{(i)} = 0\}} \\
\mu_1 &= \frac{\sum_{i=1}^{m} 1\{y^{(i)} = 1\} x^{(i)}}{\sum_{i=1}^{m} 1\{y^{(i)} = 1\}} \\
\Sigma &= \frac{1}{m} \sum_{i=1}^{m} (x^{(i)} - \mu_{y^{(i)}})(x^{(i)} - \mu_{y^{(i)}})^T
\end{aligned}
$$

对数似然函数为：

$$
\begin{aligned}
\ell(\phi, \mu_0, \mu_1, \Sigma) &= \log \prod_{i=1}^{m} p(x^{(i)}, y^{(i)}; \phi, \mu_0, \mu_1, \Sigma) \\
&= \log \prod_{i=1}^{m} p(x^{(i)} | y^{(i)}; \mu_0, \mu_1, \Sigma) p(y^{(i)}; \phi)
\end{aligned}
$$

通过最大化 ℓ，证明 $\phi, \mu_0, \mu_1, \Sigma$ 的最大似然估计值确实可以由上述公式表示。
(可以假设至少有一个正例和一个负例，从而确保上述对 μ_0 和 μ_1 的定义中分母不为零)

5. **【代码题】** 在 src/p01e_gda.py 中填写代码计算 $\phi, \mu_0, \mu_1, \Sigma$，推导 θ，并使用得到的 GDA 模型对验证集进行预测。

6. 对于第一个数据集，绘制训练数据图，横轴为 x_1，纵轴为 x_2，并使用不同的符号对 $y^{(i)} = 0$ 和 $y^{(i)} = 1$ 两个类别的示例 $x^{(i)}$ 进行表示。在同一张图上，绘制第 2 问中通过逻辑回归找到的决策边界。此外，再绘制一张同样的训练数据图，绘制第 5 问中通过 GDA 找到的决策边界。

7. 对于第二个数据集，重复第 6 问中的步骤。在哪个数据集上，GDA 的表现不如逻辑回归？为什么会出现这种情况？

8. 对于 GDA 在第 6 问和第 7 问中表现较差的数据集，你能否找到一种对 $x^{(i)}$ 的变换，使 GDA 的表现明显变得更好？这种变换是什么？