1. (Classification using GDA) Your task is to use Gaussian Discriminant Analysis (GDA) to build a classification model. To complete this assignment, make sure you:
   a) Write your own code to implement the GDA algorithm. **(Do not use built-in classification functions.)**
   b) Clearly explain how the GDA model works and why it can be used for classification, in particular this data set.
   c) Train your model on the given dataset and report its accuracy. Be explicit about how you measure performance (e.g., accuracy on a test set, cross-validation, etc.).
   d) Plot the decision boundary of your model and include the visualization in your report.

1(a)

Estimate class prior:
$\phi = \frac{1}{m}\sum_{i=1}^{m} \mathbb{1}\{y^{(i)} = 1\}$
Estimate class means:
$$\mu_0 = \frac{\sum_{i:y^{(i)}=0} x^{(i)}}{\sum_{i=1}^{m} \mathbb{1}\{y^{(i)}=0\}}, \quad \mu_1 = \frac{\sum_{i:y^{(i)}=1} x^{(i)}}{\sum_{i=1}^{m} \mathbb{1}\{y^{(i)}=1\}}$$
Estimate shared covariance matrix:
$\Sigma = \frac{1}{m}\sum_{i=1}^{m} \left(x^{(i)} - \mu_{y^{(i)}}\right)\left(x^{(i)} - \mu_{y^{(i)}}\right)^T$
Compute the decision boundary:
$\theta^T x + \theta_0 = 0$
$\theta = \Sigma^{-1}(\mu_1 - \mu_0)$
$\theta_0 = \frac{1}{2}(\mu_0^T \Sigma^{-1}\mu_0 - \mu_1^T \Sigma^{-1}\mu_1) + \log\left(\frac{1-\phi}{\phi}\right)$

1(b)

GDA is a generative learning algorithm. It models the probability distribution of each class and then uses Bayes' Rule to compute the posterior probability for classification.
It assumes data is generated from a multivariate normal distribution per class.
It works well when class-conditional distributions are close to Gaussian.

1(c)

1. Split the dataset into training and test sets.

2. Train using the steps from part (a).
3. Use $\hat{y} = \begin{cases} 1 & \text{if } \theta^T x + \theta_0 \geq 0 \\ 0 & \text{otherwise} \end{cases}$ to predict.
   $\Rightarrow \text{Accuracy} = \frac{\text{Correct Predictions}}{\text{Total Predictions}}$

1(d)

To visualize, we plot in tha data points in different color

2. (Regression) Your task is to build a regression model that represents a piecewise smooth function. To do this, combine the two models from Assignment 4 into a single function. Specifically, let
   - $C(\vec{x})$ be your classification model, and
   - $R(\vec{x})$ be your regression model.

   Then construct a model $h(\vec{x})$ defined as

$$h(\vec{x}) = \begin{cases} R(\vec{x}), & \text{if } C(\vec{x}) = 1 \\ -999, & \text{if } C(\vec{x}) = 0. \end{cases}$$

   To complete this assignment, make sure you:
   a) Implement this combined model in code.
   B) Apply your model to the dataset and verify that the piecewise definition works as expected.
   c) Briefly explain how you built the combined function.
   d) Include plots or tables that demonstrate the behavior of your model.

2(a)

- Train GDA model for classification.
- Train regression model $R(\vec{x})$ on class 1 data.
- Define function h(x) that checks class using $C(\vec{x})$, then outputs either $R(\vec{x})$ or -999.

2(b)

- Apply h(x) to test points.
- Check that values are:

- From regression if classified as class 1
- -999 if classified as class 0

2(c)

- Used GDA to determine class of each input.
- Applied regression model only to class 1.
- Used piecewise logic to switch output accordingly.