boundary - that is, what is the value of α which minimizes the probability of misclassification? What is the resulting probability of misclassification with this optimal value for α? (Hint 1: take advantage of the symmetry around x = 5; Hint 2: express the probability of misclassification as a sum of integrals, each representing a specific range of x.)

3. (**70 points**) In this programming exercise you will implement three multivariate Gaussian classifiers for two classes. The dataset you will use for this exercise is provided in `data.csv`. This dataset contains d-dimensional data points $\mathbf{X} \in \mathbb{R}^{n\times d}$ and their corresponding labels $\mathbf{y} \in \{1, 2\}^n$. Given n training samples, the classifiers will compute their corresponding parameters of both classes for the purpose of classifying new data (i.e., test data). We will use $\mathbf{m_i} \in \mathbb{R}^d$, $\mathbf{S_i} \in \mathbb{R}^{d\times d}$, and $p_i \in \mathbb{R}$ to denote the mean, covariance matrix, and prior, respectively, of class i. These three classifiers have different assumptions as follows:

Classifiers	means ($\mathbf{m}_1$, $\mathbf{m}_2$)	covariance matrices ($\mathbf{S}_1$, $\mathbf{S}_2$)	priors (p_1, p_2)
C1	$\mathbf{m}_1 \neq \mathbf{m}_2$	$\mathbf{S}_1 \neq \mathbf{S}_2$	$p_1 \neq p_2$
C2	$\mathbf{m}_1 \neq \mathbf{m}_2$	$\mathbf{S}_1 = \mathbf{S}_2$	$p_1 \neq p_2$
C3	$\mathbf{m}_1 \neq \mathbf{m}_2$	$\mathbf{S}_1 = \mathbf{S}_2$, $\mathbf{S}_1$ and $\mathbf{S}_2$ are diagonal	$p_1 \neq p_2$

Table 1: Parameter Assumptions of Three Classifiers.

We will next discuss how to compute the parameters.

- **C1** assumes that all the parameters (means, covariance matrices, priors) are learned independently for both classes. In other words, we will first split the training data into two sets (($\mathbf{X}_1$, $\mathbf{y}_1$), ($\mathbf{X}_2$, $\mathbf{y}_2$)) according to the labels (Class 1/Class 2), and then estimate the corresponding parameters following the formulas learnt from lectures.
- **C2** follows the same steps to compute the means and priors as C1. However, instead of directly utilizing independent covariance matrices, C2 computes a covariance matrix of all the data (*i.e.*, Class 1 and Class 2) to obtain a shared covariance matrix $\mathbf{S}$.
- **C3** follows the same steps to compute the parameters as C2. However, C3 additionally assumes that each feature is irrelevant to others (*e.g.*, the "size" and "color" of the "apple"), making the shared covariance matrix $\mathbf{S}$ diagonal. In other words, we will only consider the variance of each feature and set the covariance between any two different features as 0.

After computing all the parameters for three classifiers, you can finally predict the labels of test data based on the discriminant functions. The functions are defined in a *scikit-learn* convention, where you have a *fit* function for model training and a *predict* function for generating predictions on given samples.

You are provided with two jupyter notebook files – `sample.ipynb` and `hw1.ipynb`. `sample.ipynb` contains an example Gaussian discriminant classifier for the univariate case including solution. You will implement the three multivariate Gaussian discriminant classifiers described above and test their performance in `hw1.ipynb`.

Submission

- **Things to submit:**
  1. hw1_written.pdf: a document containing all your answers for Problems 1 & 2.
  2. `hw1.ipynb`: a Jupyter notebook containing three python classes for Problem 3. Use the skeleton file and fill in the missing parts. Make sure to answer the questions at the end using markdown.
- **Submit**: All material must be submitted electronically via Canvas. Please **Do NOT** submit `zip` files. Instead, submit two separate files, hw1_written.pdf and `hw1.ipynb`.