Question 2

2(e)

Case 1: $P(C=1|x)=0.5$ Given: $P(C=1|x) = \frac{1}{1+e^{-z}}$, $z = w^T x + w_0$

Let $y = \frac{1}{1+e^{-z}} \Rightarrow y = \frac{1}{1+\frac{1}{e^z}}$

$= \frac{1}{\frac{e^z+1}{e^z}} = \boxed{\frac{e^z}{e^z+1} = y}$

$\ln\left(\frac{y}{1-y}\right) = \ln\left(\frac{\frac{e^z}{e^z+1}}{1-\frac{e^z}{e^z+1}}\right)$ (log-odds ratio from slides)

$= \ln\left(\frac{\frac{e^z}{e^z+1}}{\frac{(e^z+1)-e^z}{e^z+1}}\right) = \ln\left(\frac{\frac{e^z}{e^z+1}}{\frac{1}{e^z+1}}\right) = \ln(e^z)$

① $\boxed{\ln\left(\frac{y}{1-y}\right) = z}$

$\ln\left(\frac{0.5}{1-0.5}\right) = w_0 + w_1 x_1 + w_2 x_2$

$0 = w_0 + w_1 x_1 + w_2 x_2$

$w_2 x_2 = -w_0 - w_1 x_1$

$\boxed{x_2 = \frac{-w_0 - w_1 x_1}{w_2}}$, for $P(C=1|x)=0.5$

Case 2: $P(C=1|x)=0.6$

$\ln\left(\frac{0.6}{1-0.6}\right) = w_0 + w_1 x_1 + w_2 x_2$ (reuse eqn. ①)

$0.405465 = w_0 + w_1 x_1 + w_2 x_2$

$w_2 x_2 = -w_0 + 0.405465 - w_1 x_1$

$\boxed{x_2 = \frac{-w_0 + 0.405465 - w_1 x_1}{w_2}}$, for $P(C=1|x)=0.6$

Case 3: $P(C=1|x)=0.05$

$\ln\left(\frac{0.05}{1-0.05}\right) = w_0 + w_1 x_1 + w_2 x_2$ (reuse eqn. ①)

$-2.9444\ldots = w_0 + w_1 x_1 + w_2 x_2$

$w_2 x_2 = -w_0 - 2.9444 - w_1 x_1$

$\boxed{x_2 = \frac{-w_0 - 2.944 - w_1 x_1}{w_2}}$, for $P(C=1|x)=0.05$

Question 3

3(a)

$p(y)$ = prior

Assume $p(t|\phi) = \phi^t (1-\phi)^{1-t}$ (prior is Bernoulli)

Prove $\phi = \frac{1}{N} \sum_{n=1}^{N} \mathbb{I}[t^{(n)}=1]$

$$l(\phi, \mu_0, \mu_1, \Sigma) = -\ln \prod_{n=1}^{N} p(x^{(n)}|t^{(n)}, \mu_0, \mu_1, \Sigma) p(t^{(n)}|\phi) \quad \text{(slide 17)}$$

Choose arbitrary class $k \in \{0,1\}$, preserving generality, Assume $\Sigma_1 = \Sigma_2$ (classes share same covar. matrix)

$$= -\ln \prod_{n=1}^{N} p(x^{(n)}|t^{(n)}, \mu_k, \Sigma) p(t^{(n)}|\phi)$$

(Eqn from slide 6)

$$= -\ln \prod_{n=1}^{N} \left(\frac{1}{(2\pi)^{d/2} |\Sigma_k|^{1/2}} \cdot e^{(-(x-\mu_k)^T \Sigma_k^{-1} (x-\mu_k))} \right) \cdot \phi^t (1-\phi)^{1-t}$$

$$= -\left[\sum_{n=1}^{N} \ln(1)^{\to 0} - \ln\left((2\pi)^{\frac{d}{2}} |\Sigma_k|^{\frac{1}{2}}\right) + \ln\left(e^{-(x-\mu_k)^T \Sigma_k^{-1} (x-\mu_k)}\right) + t \ln\phi + (1-t)\ln(1-\phi) \right]$$

$$= \sum_{n=1}^{N} \frac{d}{2} \ln(2\pi) + \frac{1}{2} \ln|\Sigma_k| + (x-\mu_k)^T \Sigma_k^{-1} (x-\mu_k) + t \ln\phi + (1-t)\ln(1-\phi) \quad \text{Eqn (1)}$$

Now equate derivative to 0

$$\frac{d}{d\phi} l(\phi, \mu_k, \Sigma_k) = 0 = \sum_{n=1}^{N} 0 + 0 + 0 + \frac{d}{d\phi} t^{(n)} \ln\phi + \frac{d}{d\phi} (1-t^{(n)}) \ln(1-\phi)$$

$$= \sum_{n=1}^{N} \frac{t^{(n)}}{\phi} + \frac{d}{d\phi}(1-t^{(n)})^{\to 0} + 0 \cdot \ln(1-\phi)^{\to 0} + (1-t^{(n)}) \cdot \frac{1}{(1-\phi)}(-1) \quad \text{(chain rule)}$$

$$= \sum_{n=1}^{N} \frac{t^{(n)}}{\phi} - \frac{(1-t^{(n)})}{(1-\phi)} = \sum_{n=1}^{N} \frac{t^{(n)}(1-\phi) - \phi(1-t^{(n)})}{\phi(1-\phi)} = 0$$

$$0 = \sum_{n=1}^{N} t^{(n)} - t\phi - \phi + t\phi = \sum_{n=1}^{N} (t^{(n)} - \phi) = \sum_{n=1}^{N} t^{(n)} - N\phi$$

$$\Rightarrow \phi = \frac{1}{N} \sum_{n=1}^{N} t^{(n)}$$

$$\Rightarrow \phi = \frac{1}{N} \sum_{n=1}^{N} \mathbb{I}[t^{(n)}=1] \quad \text{(since prior is Bernoulli, } t^{(n)} \in \{0,1\})$$

$$\therefore \phi = \frac{1}{N} \sum_{n=1}^{N} \mathbb{I}[t^{(n)}=1]$$

∎

3(b)

Given $\Sigma = \frac{1}{N}\sum_{n=1}^{N}\left(x^{(n)}-\mu_{t^{(n)}}\right)^T\left(x^{(n)}-\mu_{t^{(n)}}\right)$ ①, where $x^{(n)}$ is row vector

Let X_0 be the data matrix for class 0, X_1 for class 1.

Prove $\Sigma = \left[Y_0^T Y_0 + Y_1^T Y_1\right]/N$, where $Y_i = X_i - \vec{1}\mu_i^T$

where μ_i is column vector

and $\vec{1}$ is column vector

$$Y_0 = X_0 - \vec{1}\mu_0^T$$

$$= \begin{bmatrix} x^{(1)} \\ x^{(2)} \\ \vdots \\ x^{(n)} \end{bmatrix}_{nd} - \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}_{n\times 1} \begin{bmatrix} \mu_{01} & \mu_{02} & \cdots & \mu_{0d} \end{bmatrix}_{1\times d}$$

$$= \begin{bmatrix} x^{(1)} \\ x^{(2)} \\ \vdots \\ x^{(n)} \end{bmatrix}_{nd} - \begin{bmatrix} \mu_{01}^{(1)} & \mu_{02}^{(1)} & \cdots & \mu_{0d}^{(1)} \\ \mu_{01}^{(2)} & \mu_{02}^{(2)} & \cdots & \mu_{0d}^{(2)} \\ \vdots & & & \\ \mu_{01}^{(n)} & \mu_{02}^{(n)} & \cdots & \mu_{0d}^{(n)} \end{bmatrix}_{nd}$$

$$Y_0 = \begin{bmatrix} x^{(1)} - \mu_0 \\ x^{(2)} - \mu_0 \\ \vdots \\ x^{(n)} - \mu \end{bmatrix}_{nd} \quad ②$$

$$\Sigma = \frac{1}{N}\left[\sum_{n=1}^{N_0}\left(x^{(n)}-\mu_{t^{(n)}}\right)^T\left(x^{(n)}-\mu_{t^{(n)}}\right) + \sum_{n=1}^{N_1}\left(x^{(n)}-\mu_{t^{(n)}}\right)^T\left(x^{(n)}-\mu_{t^{(n)}}\right)\right]$$ (split given eqn. ① by class)

$$= \frac{1}{N}\left[\begin{bmatrix}(x^{(1)}-\mu_0) & (x^{(2)}-\mu_0) & \cdots & (x^{(n)}-\mu_0)\end{bmatrix}\begin{bmatrix} x^{(1)}-\mu_0 \\ x^{(2)}-\mu_0 \\ \vdots \\ x^{(n)}-\mu_0 \end{bmatrix} + \sum_{\cdots}^{N_1}\cdots\right]$$

$$= \frac{1}{N}\left[Y_0^T Y_0 + \sum_{n=1}^{N_1}\left(x^{(n)}-\mu_{t^{(n)}}\right)^T\left(x^{(n)}-\mu_{t^{(n)}}\right)\right]$$ (plug in eqn. ②)

$$\therefore \Sigma = \frac{1}{N}\left[Y_0^T Y_0 + Y_1^T Y_1\right]$$ ■ (plug in eqn. ② but symmetrically for Y_1 and Y_1^T)

3(c)

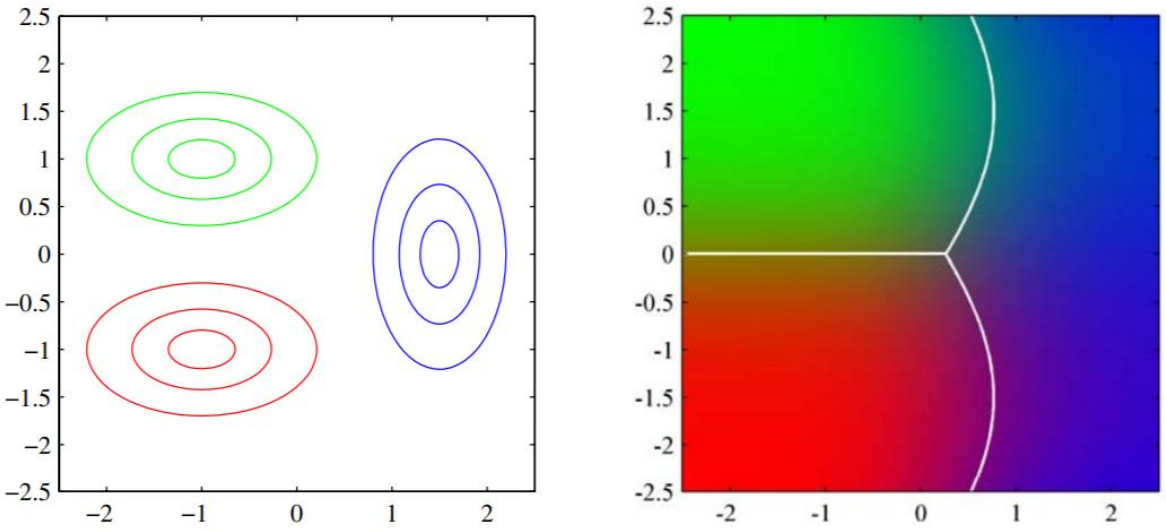
"I don't know"

Question 4

4(b)

Quadratic Discriminant Analysis, or QDA, does not assume that the covariance matrices are the same between different classes. In the case that the covariance matrices are the same, then the decision boundary between the classes which share the covariance matrix is linear. On the other hand, as we see in the Figure for 4(a), when the covariance matrices are different, QDA is free to learn quadratic boundaries. In particular, this is due to the fact that when you are calculating the decision boundaries with different covariance matrices, then the squared terms will not cancel out and you will receive quadratic decision boundaries.

The recommended text by *Bishop* demonstrates this in Figure 4.11:



Question 5

5(e)

To put it briefly, adding noise prevents the classifiers from learning parts of the whitespace surrounding the digits. This is related to the error message received about collinear variables because the classifications without noise experience a phenomenon called multicollinearity where one predictor variable can be linearly predicted from the others fairly accurately – leading to large changes in the results from small changes in the model or data. This is reflected as low accuracies in my printouts for models trained without noise.

5(f)

The similar accuracies for Naïve Bayes can be explained because Naïve Bayes relies on the conditional independence assumption that all the variables in the dataset are not correlated to each other. When this assumption holds, Naïve Bayes will converge quicker than discriminative models so you can get away with having less data to achieve similar accuracies. This is evident when comparing the Naïve Bayes accuracies between part e) and part f).

The performance for full Gaussian Bayes has plummeted because it does not converge as quickly and since we have only 90% of the data in part f) as we did in part e). The full Gaussian Bayes model has not yet converged and it is reflected by a drop in the accuracies.

5(g)

To put it simply and briefly, Gaussian Naïve Bayes predicts the probability of each class based on the feature vectors. A Gaussian is estimated for each feature, for each class. The class likelihoods are then used along with the priors and evidence to predict the posteriors for each class. The class with the largest posterior gets returned.

The background noise is no longer visible in the mean vectors because it got averaged out.