1.a

Compute the gradient of the cost function $J(\theta)$, we have:

$$\nabla_\theta J(\theta) = \frac{1}{n}\sum_{i=1}^{n}\left(-y^{(i)}\nabla_\theta \log\left(g(\theta^T x^{(i)})\right) - (1-y^{(i)})\nabla_\theta \log\left(1-g(\theta^T x^{(i)})\right)\right)$$

$$= \frac{1}{n}\sum_{i=1}^{n}\left(-y^{(i)}\left(1-g(\theta^T x^{(i)})\right)x^{(i)} + (1-y^{(i)})\,g(\theta^T x^{(i)})x^{(i)}\right)$$

$$= \frac{1}{n}\sum_{i=1}^{n}\left(g(\theta^T x^{(i)}) - y^{(i)}\right)x^{(i)}$$

Taking the second derivative of the gradient, we can obtain the Hessian matrix:

$$H(J)(\theta) = \frac{1}{n}\sum_{i=1}^{n} g(\theta^T x^{(i)})\left(1-g(\theta^T x^{(i)})\right)x^{(i)}x^{(i)T}$$

We know that $h_\theta(x^{(i)}) = g(\theta^T x^{(i)})$. For any vector $z \in \mathbb{R}^{d+1}$, let's consider:

$$z^T H(J)(\theta) z = \frac{1}{n} \sum_{i=1}^{n} h_\theta(x^{(i)})(1 - h_\theta(x^{(i)}))(z^T x^{(i)})^2$$

All the terms are non-negative as $h_\theta(x^{(i)})$ is a probability and squares are non-negative. It means that $z^T H(J)(\theta) z \geq 0$ for all vector $z \in \mathbb{R}^{d+1}$, hence $H(J)(\theta) \succeq 0$.

1c.

We have:

$$p(y=1 \mid x; \theta, \mu_0, \mu_1, \Sigma) = \frac{p(y=1)\,p(x \mid y=1)}{p(y=0)\,p(x \mid y=0) + p(y=1)\,p(x \mid y=1}$$

$$= \frac{1}{1 + \dfrac{p(y=0)\,p(x \mid y=0)}{p(y=1)\,p(x \mid y=1)}}$$

Let's consider the fraction in the dominator, we have:

$$\frac{p(y=0)\,p(x \mid y=0)}{p(y=1)\,p(x \mid y=1}$$

=

$$= \frac{\frac{1-\phi}{(2\pi)^{d/2}|\Sigma|^{1/2}} \exp\left((x-\mu_1)^T \Sigma^{-1}(x-\mu_1)\right)}{\frac{\phi}{(2\pi)^{d/2}|\Sigma|^{1/2}} \exp\left((x-\mu_0)^T \Sigma^{-1}(x-\mu_0)\right)}$$

$$= \left(\frac{1-\phi}{\phi}\right) \exp\left(\frac{1}{2}(x-\mu_1)^T\Sigma^{-1}(x-\mu_1) - \frac{1}{2}(x-\mu_0)^T\Sigma^{-1}(x-\mu_0)\right)$$

We can re-write $x-\mu_1$ as $x-\mu_0+\mu_0-\mu_1$, the exponent in the expression above can be derived as follows:

$$\frac{1}{2}(x-\mu_1)^T\Sigma^{-1}(x-\mu_0+\mu_0-\mu_1) - \frac{1}{2}(x-\mu_0)^T\Sigma^{-1}(x-\mu_0)$$

$$= -\frac{1}{2}(\mu_1-\mu_0)^T\Sigma^{-1}(x-\mu_0) - \frac{1}{2}(\mu_1-\mu_0)^T\Sigma^{-1}(x-\mu_1)$$

$$= -(\mu_1-\mu_0)^T\Sigma^{-1}\left(x - \frac{\mu_0+\mu_1}{2}\right)$$

By combining all the elements, we obtain the above fraction equal to:

$$\exp\left\{-(\mu_1-\mu_0)^T\Sigma^{-1}x-\left(\log\left(\frac{\phi}{1-\phi}\right)-\frac{(\mu_1-\mu_0)^T\Sigma^{-1}(\mu_0+\mu_1)}{2}\right)\right\}$$

if we let $\theta=(\mu_1-\mu_0)^T\Sigma^{-1}x$,

$$\theta_0=\log\left(\frac{\phi}{1-\phi}\right)-\frac{(\mu_1-\mu_0)^T\Sigma^{-1}(\mu_0+\mu_1)}{2}$$

so the posterior probability can be written as $\frac{1}{1+\exp(-\theta^T-\theta_0)}$

If the threshold to make prediction is 0.5, the decision boundary will have an equation $\theta^Tx+\theta_0=0$.

1. d.

The log-likelihood is given by:

$$\ell(\phi, \mu_0, \mu_1, \Sigma) = \log \prod_{i=1}^{n} p(x^{(i)} | y^{(i)}) p(y^{(i)})$$

$$= \sum_{i=1}^{n} \left(\log p(x^{(i)} | y^{(i)}) \log(y^{(i)}) \right)$$

$$= \sum_{i=1}^{n} \left(-\log(2\pi)^{d/2} - \frac{1}{2} \log|\Sigma| - \frac{1}{2} (x^{(i)} - \mu_{y^{(i)}})^T \Sigma^{-1} (x^{(i)} - \mu_{y^{(i)}}) \right.$$

$$\left. + y^{(i)} \log \phi + (1 - y^{(i)}) \log(1 - \phi) \right)$$

The maximum likelihood estimate of $\theta = (\phi, \mu_0, \mu_1, \Sigma)$ are the values which maximizes the log-likelihood function solving the equation $\nabla \ell(\theta) = 0$ in order to obtain them.

We've got:

$$\nabla_\phi \ell(\theta) = \sum_{i=1}^{n} \left(\frac{y^{(i)}}{\theta} - \frac{1 - y^{(i)}}{1 - \phi} \right) = 0$$

solution of the equation above gives:

$$\hat{\phi} = \frac{1}{n} \sum_{i=1}^{n} y^{(i)} = \frac{1}{n} \sum_{i=1}^{n} 1\{y^{(i)} = 1\}$$

Consider now:

$$\nabla_{\mu_0} \ell(\theta) = \nabla_{\mu_0} \left(\sum_{i=1}^{n} -\frac{1}{2} 1\{y^{(i)} = 0\} (x^{(i)} - \mu_0)^T \Sigma^{-1} (x^{(i)} - \mu_0) \right)$$

$$= -\Sigma^{-1} \sum_{i=1}^{n} 1\{y^{(i)} = 0\} (\mu_0 - x^{(i)})$$

Solve the equation $\nabla_{\mu_0} \ell(\theta) = 0$ gives:

$$\hat{\mu}_0 = \frac{\sum_{i=1}^{n} 1\{y^{(i)} = 0\} x^{(i)}}{\sum_{i=1}^{n} 1\{y^{(i)} = 1\}}$$

Similarly, we obtain $\hat{\mu}_1 = \frac{\sum_{i=1}^{n} 1\{y^{(i)}=1\} x^{(i)}}{\sum_{i=1}^{n} 1\{y^{(i)}=1\}}$

when we solve equation $\nabla_{\mu_1} l(\theta) = 0$.

Finally, we computing:

$$\nabla_{\Sigma} l(\theta) = \nabla_{\Sigma} \sum_{i=1}^{n} \left(-\frac{1}{2}\log|\Sigma| - \frac{1}{2}(x^{(i)} - \mu_{y^{(i)}})^T \Sigma^{-1} (x^{(i)} - \mu_{y^{(i)}}) \right)$$

$$= \sum_{i=1}^{n} \left(-\frac{1}{2}\Sigma^{-1} + \frac{1}{2}\Sigma^{-1}(x^{(i)} - \mu_{y^{(i)}})(x^{(i)} - \mu_{y^{(i)}})^T \Sigma^{-1} \right)$$

$$= \frac{1}{2}\Sigma^{-1} \left(\sum_{i=1}^{n} (x^{(i)} - \mu_{y^{(i)}})(x^{(i)} - \mu_{y^{(i)}})^T \Sigma^{-1} - n \right)$$

Solve the equation $\nabla_{\Sigma} l(\theta) = 0$, we get:

$$\hat{\Sigma} = \frac{1}{n} \sum_{i=1}^{n} (x^{(i)} - \mu_{y^{(i)}})(x^{(i)} - \mu_{y^{(i)}})^T$$

In derivation of the above equation $\nabla_\Sigma \ell(\theta)$, we used 2 identities:

$$\nabla_\Sigma \log |\Sigma| = (\Sigma^{-1})^T$$

$$\nabla_\Sigma z^T \Sigma^{-1} z = -\Sigma^{-1} z z^T \Sigma^{-1}$$

where z isn't a function of Σ. To prove the second identity, we take the differential:

$$\begin{aligned} d(z^T \Sigma^{-1} z) &= d(\mathrm{Tr}[z^T \Sigma^{-1} z]) \\ &= \mathrm{Tr}[z z^T d\Sigma^{-1}] \\ &= \mathrm{Tr}[-z z^T \Sigma^{-1} (d\Sigma) \Sigma^{-1}] \\ &= \mathrm{Tr}[-\Sigma^{-1} z z^T \Sigma^{-1} d\Sigma] \end{aligned}$$

We replace $d\Sigma^{-1} = -\Sigma^{-1}(d\Sigma)\Sigma^{-1}$ (derived from identity $\Sigma\Sigma^{-1} = I$).

Thus $\nabla_{\Sigma} z^{\top} \Sigma^{-1} z = -\Sigma^{-1} z z^{\top} \Sigma^{-1}$

1.f.

GDA seems to be affected more towards the outlier in the validation dataset 1. This indicates the GDA may have a higher bias compared to logistic regression.

1.g.

Regarding the validation dataset 2, both models gave rather comparable outcomes, though the logistic regression was slightly better. All in all, both models learned decision boundaries are good on this data set.

Based on the previous question, it was evident that GDA underperformed compared to the logistic regression on dataset 1. Furthermore, it seems that the two classes in dataset 2 do not look like a

Gaussian distribution.

1.h.

The z-transformation given by:

$$z^{(i)} = \frac{x^{(i)} - \mu}{s}, \quad i = 1, \ldots, n$$

Where μ is the mean of the dataset and s is the standart deviation of the dataset might help improve the GNN's performance with Dataset 1

2.c $$p(t^{(i)}=1 \mid y^{(i)}=1, x^{(i)}) =$$

$$= \frac{p(t^{(i)}=1 \mid x^{(i)})\, p(y^{(i)} \mid t^{(i)}=1, x^{(i)})}{p(t^{(i)}=1 \mid x^{(i)})\, p(y^{(i)}=1 \mid t^{(i)}=1, x^{(i)}) + p(t^{(i)}=0 \mid x^{(i)})\, p(y^{(i)}=1 \mid t^{(i)}=0, x^{(i)})} =$$

$$= 1$$

The term $p(y^{(i)}=1 \mid t^{(i)}=0, x^{(i)}) = 0 \;\; \forall x^{(i)}$.

2. d

We get:

$$p(y^{(i)}=1|x^{(i)}) = p(t^{(i)}=1|x^{(i)})\,p(y^{(i)}=1|t^{(i)}=1, x^{(i)}) + p(t^{(i)}=0|x^{(i)})\,p(y^{(i)}=1|t^{(i)}=0, x^{(i)})$$

$$= \alpha \cdot p(t^{(i)}=1|x^{(i)})$$

Thus:

$$p(t^{(i)}=1|x^{(i)}) = \frac{1}{\alpha} \cdot p(y^{(i)}=1|x^{(i)}).$$

2 e.

We have $p(y^{(i)}=1|x^{(i)})$, then

$$\begin{aligned}
h(x^{(i)}) &= p(t^{(i)}=1|x^{(i)})\, p(y^{(i)}=1|t^{(i)}=1, x^{(i)}) && (1)\\
&\quad + p(t^{(i)}=0|x^{(i)})\, p(y^{(i)}=1|t^{(i)}=0, x^{(i)}) \\
&= p(t^{(i)}=1|x^{(i)}) \cdot c + p(t^{(i)}=0|x^{(i)}) \cdot 0 && (2)\\
&= 1 \cdot c && (3)\\
&= c && (4)
\end{aligned}$$

When $y^{(i)}$ is already set to 1 in line (3), the label for $x^{(i)}$ becomes 1 as well. As per our critical assumption $p(t^{(i)}=1|x^{(i)})=1$ and $p(t^{(i)}=0|x^{(i)})=0$. In the scenario where $y^{(i)}=1$, $h(x^{(i)})$ will constantly have a value of c, leading to its conditional expectation being c.

Hence, we can say that the conditional expectation of $d(x^{(i)})$ when $y^{(i)}$ is equal to 1 is α. Which can be represented as $E[d(x^{(i)}) | y^{(i)} = 1] = \alpha$.

3. a (i)

Consider a classifier that always predicts the majority class label, which in this case is the negative label with frequency $1-p$. This classifier will predict negative for all examples, getting a correct prediction rate of $1-p$ for negative class examples. For positive class examples, it will make incorrect prediction for all examples. The overall accuracy of this classifier will be:

$$\frac{TP+TN}{TP+TN+FP+FN} = \frac{TN}{TN+FP} = 1-p.$$

Thus, for any dataset with p fraction of positive examples and $1-p$ fraction of negative examples, there exists a trivial classifier that achieves accuracy of at least $1-p$.

To verify that the accuracy is equal to $A = \frac{TP+TN}{TP+TN+FP+FN}$:

as we know $TP+TN$ represents the number of examples that are correctly classified by the model and $TP+TN+FP+FN$ represents the total number of examples in the dataset. Therefore, the accuracy can be expressed as:

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN}.$$

Dividing numerator and denominator by $TP+TN$, we get:

$$\text{Accuracy} = \frac{(TP/(TP+TN))+(TN/(TP+TN))}{((TP+FP)/(TP+TN))+((FN/(TP+TN))+((FP/(TP+TN))}$$

The terms in the numerator of the right-hand side of the equation are equal to A_1 and A_0, respectively:

$$A_1 = \frac{TP}{TP+FN} = \frac{TP}{TP+TN+FN-TN} =$$

$$= \frac{TP}{TP+TN-FP} = \frac{TP}{TP+FP} =$$

$$= \frac{(TP/(TP+TN))}{((TP+FP)/(TP+TN))}$$

$$A_0 = \frac{TN}{TN+FP} = \frac{TN}{TP+TN+FP-TP} =$$

$$= \frac{TN}{TN+FN+TN} = \frac{TN}{TN+FN} =$$

$$= \frac{(TN/(TP+TN))}{((TN+FN)/(TP+TN))}$$

Substituting these values back into the equation, we obtain:

$$\text{Accuracy} =$$

$$= \frac{A_1 \cdot (TP+FN)}{(TP+TN+FN)} + \frac{A_0 \cdot (TN+FP)}{(TP+TN+FP)}$$

Multiplying both terms by $(TP+TN+FP+FN)/(TP+TN+FP+FN)$ we obtain:

$$\text{Accuracy} = \frac{A_1 \cdot (TP + FN) + A_0 \cdot (TN + FP)}{TP + TN + FP + FN}.$$

so we can simply re-write it as

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} = A$$

Thus we have verified that accuracy is equal to the balanced accuracy A, which takes into account both positive and negative class prediction.

3a (ii)

We can first observe that $TP+FN$ is the total number of positive examples in the validation set, and $TN+FP$ is the total number of negative examples. Therefore, the total number of examples is $TP+TN+FP+FN$.

We can rewrite p as:

$$p = \frac{\#\text{ positive examples}}{\#\text{ total examples}}$$

$$= \frac{TP+FN}{TP+TN+FP+FN}.$$

To show that $A = p \cdot A_1 + (1-p) A_0$, we can start by substituting the definitions of A, A_0 and A_1

$$A = \frac{1}{2}(A_0 + A_1)$$

$$= \frac{1}{2}\left(TN/(TN+FP) + TP/(TP+FN)\right)$$

$$=$$

$$= \frac{1}{2}\left(\left(TN/(TP+FN)\right)/\left((TN+FP)(TP+FN)\right)+\left(TP(TN+FP)\right)/\left((TP+FN)(TN+FP)\right)\right)$$

$$= 1/2\left(\left(TN\cdot TP+TN\cdot FP+TP\cdot FN+TP\cdot TN\right)/\left((TP+TN)(FP+FN)\right)\right)$$

$$= (TP+TN)/\left(2(TP+TN+FP+FN)\right)$$

Then, we can substitute the expression for p:

$$A = (TP+TN)/\left(2(TP+TN+FP+FN)\right)$$

$$= \left[(TP+FN)/(TP+TN+FP+FN)\right]\cdot\left[(TP/(TP+FN))+(TN/(TN+FP))\right]$$

$$= p\cdot A_1 + (1-p)\cdot A_0$$

3a (iii)

Since the classifier predicts the majority class for every example, it will predict negative for all positive examples, resulting in $TP = 0$ and FN = number of positive examples.

We can calculate balanced accuracy as:

$$A = \frac{1}{2}(A_0 + A_1)$$

$$= \frac{1}{2}(TN/(TN+FP)) + \frac{1}{2}(TN/(TN+FP) + 0/(0+FN))$$

$$= \frac{1}{2}(TN/(TN+FP))$$

$$= \frac{1}{2}((TP+TN+FP+FN-(1-p)(TP+TN+FP+FN)+FP)/(TP+TN+FP+FN))$$

$$= \frac{1}{2}((p\cdot(TP+TN+FP+FN)+TN-TP-FP)/(TP+TN+FP+FN))$$

$$= \frac{1}{2}\left(\left(p(TP+TN+FP+FN) - (1-p)(TP+TN+FP+FN)\right) / (TP+TN+FP+FN)\right)$$

$$= \frac{1}{2}(2 \cdot p - 1)$$

$$= p - \frac{1}{2}$$

Since the trivial classifier predicts the majority class for every example, it will have $p \geq \frac{1}{2}$. Therefore the balanced accuracy A will be:

$A = p - \frac{1}{2} \geq 0$. However, it's not necessarily equal to 50%. It will be 50% only if dataset is perfectly balanced $p = \frac{1}{2}$. In general, the balanced accuracy will be greater than 50% if classifier performs better than trivial classifier and less than 50% if it performs worse.

Therefore, the trivial classifier constructed for part (Accuracy = $\frac{TP+TN}{TP+TN+FP+FN} = A$) will have a balanced accuracy of at least 50%, but it could be greater than 50% if the data is imbalanced.

3.C.

To prove that the balanced accuracy on $\mathcal{D}$ is equal to the accuracy on $\mathcal{D}'$, we need to show that the number of true positives, true negatives, false positive, false negatives are the same for both datasets. Let TP, TN, FP, FN denote the number of true positives, true negatives false positives and false negatives on $\mathcal{D}$, and let TP', TN', FP' and FN' denote on $\mathcal{D}'$.

First denote that $TN = TN'$ since each negative example in $\mathcal{D}$ appears once in $\mathcal{D}'$. Similarly, $FP = FP'$ since each negative example is still classified as negative. Also $TP' = kTP$ since each positive example ~~in~~ in $\mathcal{D}$ appears $\frac{1}{k}$ times in $\mathcal{D}'$. Finally, $FN' = kFN$ since

each false negative in D appears $\frac{1}{k}$ times in D'.
Therefore, the balanced accuracy on D is

$$A = \frac{1}{2}\left(\frac{TP + TN}{TP + FN + TN + FP}\right)$$

The accuracy on D' is the proportion of the correct predictions, which is:

$$A' = \frac{TP' + TN'}{TP' + TN' + FP' + FN'} = \frac{kTP + TN}{kTP + TN + FP + kFN}$$

We can simplify A' as follows:

$$A' = \frac{kTP}{kTP + kFN + FP + TN} + \frac{TN}{kTP + kFN + FP + TN}$$

$$= \frac{kTP}{(k(TP + FN) + FP + TN)} + \frac{TN}{(k(TP + FN) + FP + TN)}$$

$$= \frac{k\left(TP/(TP+FN)+FP/(TP+FN)\right)}{k+TN/(TP+FN+FP+TN)}$$

$$= pA_1 + (1-p)A_0$$

where A_1 and A_0 are the accuracies for the positive and negative examples respectively.

Thus, we have shown that $A = A'$. This implies that the logistic regression trained on the new dataset $\mathcal{D}'$ has the same balanced accuracy as the original dataset $\mathcal{D}$.

The empirical loss for logistic regression is given by

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

where $h_\theta(x) = \frac{1}{1+\exp(-\theta^T x)}$ is the sigmoid function. To account for the reweighting of the dataset, we can modify the empirical loss as follows:

$$J'(\theta) = -\frac{1}{n}\sum_{i=1}^{n} w^{(i)}\left(y^{(i)}\log(h_\theta(x^{(i)})) + (1-y^{(i)})\log(1-h_\theta(x^{(i)}))\right),$$

note that $w^{(i)} = \frac{1}{k}$ if $y^{(i)} = 1$, and $w^{(i)} = 1$ otherwise. Let's now calculate the average empirical loss for logistic regression on the dataset $\mathcal{D}$:

$$J'(\theta) = -\frac{1}{|\mathcal{D}|}\sum_{(x^{(i)},y^{(i)})\in\mathcal{D}'}\left[y^{(i)}\log(h_\theta(x^{(i)})) + (1-y^{(i)})\log(1-h_\theta(x^{(i)}))\right]$$

Since $\mathcal{D}'$ contains each negative example once and $\frac{1}{k}$ repetitions of each positive example in $\mathcal{D}$, we can

rewrite $J'(\theta)$ as:

$$J'(\theta) = -\frac{1}{n} \sum_{i=1}^{n} w^{(i)} \left(y^{(i)} \log\left(h_\theta(x^{(i)})\right) + (1-y^{(i)}) \log\left(1-h_\theta(x^{(i)})\right) \right)$$

where $w^{(i)} = 1$ if $y^{(i)} = 0$ and $w^{(i)} = \frac{1}{2}$ if $y^{(i)} = 1$

Therefore, we have shown that the average loss for logistic regression on the dataset $\mathcal{D}'$ is equal to J.