1. For part A, we are given the basic normal conditional density function for a random vector $\boldsymbol{X}$ and its classification to class $y_i,\ i = 1,2$ (binary case):

$$P(\boldsymbol{x}|y_i) = p(\boldsymbol{x}) = \frac{1}{(2\pi)^{\frac{d}{2}}|\boldsymbol{\Sigma}|^{\frac{1}{2}}}\exp\left[-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_i)^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_i)\right]$$

Where $\boldsymbol{\mu}_i$ is the mean vector and $\boldsymbol{\Sigma}$ is the covariance matrix, in the case that the covariance matrices for all q classes are equal. The minimum error-rate classification can be achieved by use of the discriminant function $g_i(\boldsymbol{x})$:

$$g_i(\boldsymbol{x}) = \ln p(\boldsymbol{x}|y_i) + \ln P(y_i)$$

Since we assume that the densities $p(\boldsymbol{x}|y_i)$ are multivariate normal, then this function can be expanded into:

$$g_i(\boldsymbol{x}) = -\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_i)^T\boldsymbol{\Sigma}_i^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_i) - \frac{d}{2}\ln 2\pi - \frac{1}{2}\ln|\boldsymbol{\Sigma}_i| + \ln P(y_i)$$

Now we are also given that the covariance matrices of (both) corresponding classes are equal, i.e. $\Sigma_i = \Sigma_1 = \Sigma_2 = \Sigma$. In this case, the cluster for the ith class is centered on the mean vector $\boldsymbol{\mu}_i$. In this case, the terms $-\frac{d}{2}\ln 2\pi$ and $-\frac{1}{2}\ln|\boldsymbol{\Sigma}_i| = -\frac{1}{2}\ln|\boldsymbol{\Sigma}|$ are constants which are independent of i, reducing the equation to:

$$g_i(\boldsymbol{x}) = -\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_i)^T\boldsymbol{\Sigma}_i^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_i) + \ln P(y_i)$$

The term of interest in this equation, $(\boldsymbol{x}-\boldsymbol{\mu}_i)^T\boldsymbol{\Sigma}_i^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_i)$, is also known as the Mahalanobis distance. Rearranging the terms using the rules of transpose in this equation gives us our desired result, the final linear discriminant function:

$$g_i(\boldsymbol{x}) = \boldsymbol{\mu}_i^T\boldsymbol{\Sigma}^{-1}\boldsymbol{x} - \frac{1}{2}\boldsymbol{\mu}_i^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_i + \ln P(y_i)$$

Expansion of this gives us a quadratic term, $\boldsymbol{x}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{x}$, which is independent of i, thus giving us an equivalent linear equation:

$$g_i(\boldsymbol{x}) = \boldsymbol{w}_i^T\boldsymbol{x} + w_{i0}$$

For part B, to show $g(\boldsymbol{x}) = g_1(\boldsymbol{x}) - g_2(\boldsymbol{x})$, we start by using the given equation:

$$g(\boldsymbol{x}) = \boldsymbol{w}^T\boldsymbol{x} + w_o = (\boldsymbol{\mu}_1^T - \boldsymbol{\mu}_2^T)\boldsymbol{\Sigma}^{-1}\boldsymbol{x} - \frac{1}{2}(\boldsymbol{\mu}_1^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_2) + \ln\frac{P(y_1)}{P(y_2)}$$

Notice that this can be rearranged and rewritten as:

$$g(\boldsymbol{x}) = [\boldsymbol{\mu}_1^T\boldsymbol{\Sigma}^{-1}\boldsymbol{x} - \boldsymbol{\mu}_2^T\boldsymbol{\Sigma}^{-1}\boldsymbol{x}] + \left[-\frac{1}{2}\boldsymbol{\mu}_1^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_1 - \left(-\frac{1}{2}\boldsymbol{\mu}_2^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_2\right)\right] = 0$$

Notice that if we set $\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_i = \boldsymbol{w}_i$ and $-\frac{1}{2}(\boldsymbol{\mu}_i^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_i) + \ln P(y_i) = w_{i0}$, then we can rewrite $g(\boldsymbol{x})$ in the more simple and familiar form:

$$g(\boldsymbol{x}) = [\boldsymbol{w}_1^T\boldsymbol{x} - \boldsymbol{w}_2^T\boldsymbol{x}] + \left[w_{1,0} - w_{2,0}\right] = 0$$

Further rearranging of the terms leaves us with:

$$g(\boldsymbol{x}) = \boldsymbol{w}_1^T\boldsymbol{x} + w_{1,0} - \boldsymbol{w}_2^T\boldsymbol{x} - w_{2,0} = 0$$

If $g_i(\boldsymbol{x}) = \boldsymbol{w}_i^T\boldsymbol{x} + w_{i0}$, then this gives the final result:

$$g(\boldsymbol{x}) = g_1(\boldsymbol{x}) - g_2(\boldsymbol{x}) = 0$$

Since we showed above that $g_i(\boldsymbol{x})$ is linear, then $g_1(\boldsymbol{x}) - g_2(\boldsymbol{x}) = g(\boldsymbol{x})$ is also linear.

2. For Problem 2, the gradient descent algorithm problem, we are given:

$$E(\boldsymbol{w}) = 2w_1^2 + 2w_1w_2 + 5w_2^2$$

Since our starting point is $\boldsymbol{w} = [\mathbf{2} \quad \mathbf{-2}]^{\boldsymbol{T}}$, we plug in these values to get:

$$E(\boldsymbol{w}) = 2(2)^2 + 2(2)(-2) + 5(-2)^2 = 8 - 8 + 20 = \mathbf{20}$$

Then, we take the partial derivatives with respect to both components of $\boldsymbol{w}$:

$$\frac{\partial E(\boldsymbol{w})}{\partial w_1} = 4w_1 + 2w_2, \qquad \frac{\partial E(\boldsymbol{w})}{\partial w_2} = 10w_2 + 2w_1$$

And for the first iteration we will use both these results and a value $\alpha = 0.1$ to compute both components of w':

$$w_1' = w_1 - \alpha\frac{\partial E(\boldsymbol{w})}{\partial w_1} = 2 - 0.1[4(2) + 2(-2)] = \mathbf{1.6}$$

$$w_2' = w_2 - \alpha\frac{\partial E(\boldsymbol{w})}{\partial w_2} = -2 - 0.1[10(-2) + 2(2)] = \mathbf{-0.4}$$

$$\begin{aligned} E(\boldsymbol{w}') &= 2(w_1')^2 + 2w_1'w_2' + 5(w_2')^2 \\ &= 2(1.6)^2 + 2(1.6)(-0.4) + 5(-0.4)^2 \\ &= 5.12 - 1.28 + 0.8 = \mathbf{4.64} \end{aligned}$$

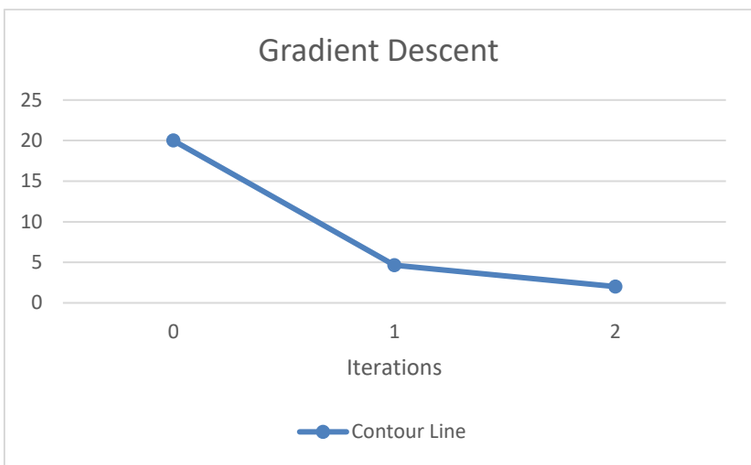
And the process is repeated for the second iteration, called w'':

$$w_1'' = w_1 - \alpha\frac{\partial E(\boldsymbol{w})}{\partial w_1} = 2 - 0.1[4(1.6) + 2(-0.4)] = \mathbf{1.04}$$

$$w_2'' = w_2 - \alpha\frac{\partial E(\boldsymbol{w})}{\partial w_2} = -2 - 0.1[10(1.6) + 2(-0.4)] = \mathbf{-0.32}$$

$$\begin{aligned} E(\boldsymbol{w}'') &= 2(w_1'')^2 + 2w_1''w_2'' + 5(w_2'')^2 \\ &= 2(1.04)^2 + 2(1.04)(-0.32) + 5(-0.32)^2 \\ &= 2.1632 - 0.6656 + 0.512 = \mathbf{2.0096} \end{aligned}$$

And we can now plot the contour lines:



Further iterations of this algorithm will approach a local minimum. Recall that since $E(\boldsymbol{w}) = 2w_1^2 + 2w_1w_2 + 5w_2^2$ is a quadratic equation with two variables, this will form an elliptic paraboloid when plotted. We can use the formula $4AB - C^2$ to find the minimum, since $E(\boldsymbol{w})$ is of the form $Ax^2 + By^2 + Cxy$. Plugging in values shows $4AB - C^2 = 4(2)(5) - 2^2 = 36 > 0$, thus there exists one minimum. Therefore, the value that this algorithm approaches will be the absolute maximum.

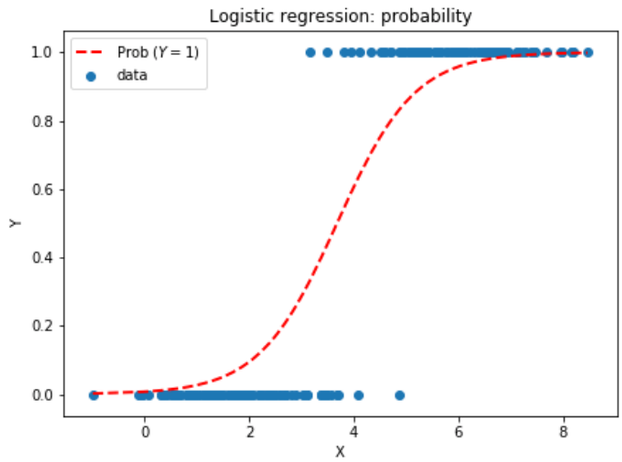
3. To show that estimation of parameters in logistic regression is nonlinear, we first write out the general regression formula:

$$y = \boldsymbol{\beta X} + \varepsilon$$

Where $\boldsymbol{\beta}$ is our coefficient parameter matrix, $\boldsymbol{X}$ is our input data matrix, y is the regressand and ε represents some random error. Our goal in regression is to optimize $\boldsymbol{\beta}$, and this is the same in both the linear and nonlinear sense. However, when our regressand takes on a binary response variable (that is, can only be 0 or 1), it is more appropriate to use logistic regression. Linear regression necessitates extrapolation of y to values beyond 0 or 1, and thus we instead use the following probability density function:

$$P(Y = 1) = \frac{\exp\left(\beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_k x_k + \varepsilon\right)}{1 + \exp\left(\beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_k x_k + \varepsilon\right)}$$

Where k is the number of parameters in $\boldsymbol{X}$. Thus, coupled with the fact that the density function is a fraction whose components are exponential functions, we can also visually see the graph of $p(y|\boldsymbol{X})$ is clearly nonlinear:



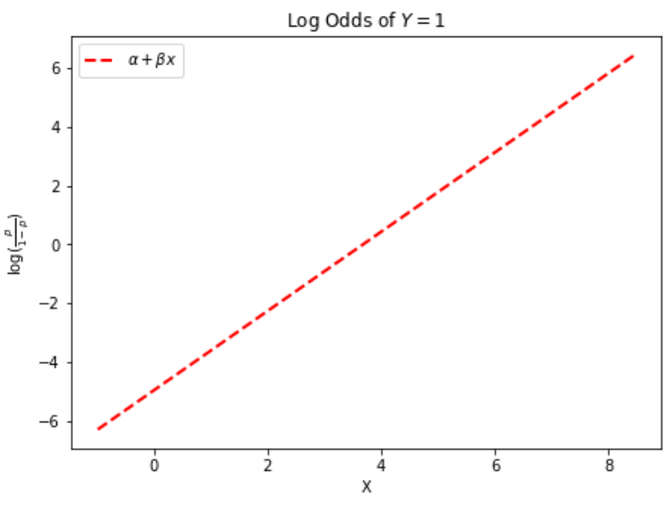
However, it is possible to treat logistic regression in the terms of a linear regression optimization problem. If we define the odds of Y as:

$$\left(\frac{p}{1-p}\right) = \exp\left(\beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_k x_k + \varepsilon\right)$$

If we take the logarithm of both sides of this equation:

$$\ln\left(\frac{p}{1-p}\right) = \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_k x_k + \varepsilon$$

Now this is in the form of a linear equation in the k-dimensional space, as shown by a sample plot:



References:

Mimi, M. (2017). *Discriminant Functions For The Normal(Gaussian) Density - Part 2 - Rhea*. Project Rhea. https://www.projectrhea.org/rhea/index.php/Discriminant_Functions_For_The_Normal(Gaussian)_Density_-_Part_2

Nihy, K. (2011). Bayesian decision theory. In *Pattern Recognition* (pp. 13–29). https://cw.fel.cvut.cz/old/_media/courses/a3b33kui/knihy/pattern_classification_chapter_02.pdf?cache=nocache

Veksler, O. (2013, October). *Pattern Recognition* [Slides]. Https://Www.Csd.Uwo.Ca/. https://www.csd.uwo.ca/~oveksler/Courses/CS434a_541a/Lecture9.pdf