STAT-6655 HWK 2

Cody Grogan

January 2024

Problem 1

The main point Belkin is trying to convey in this paper is the discrepancy between the generalization curve for classical models and empirical results from modern machine learning models. Belkin posits models that can represent an arbitrary function class experience a second descent in testing error when the number of parameters in the model is greater than the number of data points. Belkin calls this the "double descent curve", or where the assumption that a classical model is in the "sweet spot" or when a model is neither underfit nor overfit to the data. In classical models, this yields a model with the best test error for the training data. In machine learning models with arbitrary complexity, such as neural networks or random forests, increasing the complexity of the model past overfitting yields models with test errors that decrease with increasing parameters. However, this seemingly easy way to minimize test error is made more complex when the complexity of the model surpasses the ability of the optimizer to find low-risk solutions. This implies that the classical "sweet spot" may be optimal for extremely high-dimensional problems.

Problem 2

a)

The pdf of the function is $p(x, \theta) = \frac{1}{\beta} e^{-\frac{x}{\beta}}$ where the parameter to optimize is β. Defining the log-likelihood of the pdf.

$$l(\beta) = \sum_{i=1}^{n} \log(\frac{1}{\beta} e^{-\frac{x_i}{\beta}}) \tag{1}$$

$$= \sum_{i=1}^{n} \log(\frac{1}{\beta}) + \log(e^{-\frac{x_i}{\beta}}) \tag{2}$$

$$= \sum_{i=1}^{n} \log(1) - \log(\beta) - \frac{x_i}{\beta} \tag{3}$$

$$= \sum_{i=1}^{n} - \log(\beta) - \frac{x_i}{\beta} \tag{4}$$

$$= -n \log(\beta) - \frac{1}{\beta} \sum_{i=1}^{n} x_i \tag{5}$$

Now taking the derivative with respect to β to maximize the result,

$$\frac{dl}{d\beta} = -\frac{n}{\beta} + \frac{1}{\beta^2}\sum_{i=1}^{n} x_i \tag{6}$$

$$0 = -\frac{n}{\beta} + \frac{1}{\beta^2}\sum_{i=1}^{n} x_i \tag{7}$$

$$\frac{n}{\beta} = \frac{1}{\beta^2}\sum_{i=1}^{n} x_i \tag{8}$$

$$n\beta = \sum_{i=1}^{n} x_i \tag{9}$$

$$\boxed{\beta = \frac{1}{n}\sum_{i=1}^{n} x_i} \tag{10}$$

This shows the MLE of β is simply the sample mean of the variables. This makes sense because the expectation of this specific exponential distribution is β and the value of beta should be best approximated by the sample mean.

b)

The likelihood of θ can be written as,

$$L(\theta) = \prod_{i=1}^{n} \frac{1}{\theta} \tag{11}$$

$$L(\theta) = (\frac{1}{\theta})^n \tag{12}$$

$$\boxed{L(\theta) = \theta^{-n}} \tag{13}$$

The log-likelihood is then,

$$l(\theta) = \sum_{i=1}^{n} \log(\frac{1}{\theta}) \tag{14}$$

$$l(\theta) = \sum_{i=1}^{n} -\log(\theta) \tag{15}$$

$$\boxed{l(\theta) = -n\log(\theta)} \tag{16}$$

The MLE is then,

$$\frac{dl}{d\theta} = -\frac{n}{\theta} \tag{17}$$

$$0 = -\frac{n}{\theta} \tag{18}$$

We then see the log-likelihood does not allow us to estimate θ. However, the likelihood function is a decreasing function of θ. We also know that the likelihood is maximized when all samples are between 0 and θ. Otherwise, the likelihood would be 0. Thus the MLE of θ is one where θ is only large enough to bound the largest sample.

$$\boxed{\hat{\theta} = max(X_1, ..., X_n)} \tag{19}$$

Problem 3

Using the loss function given our empirical risk is,

$$\hat{R}(f) = \frac{1}{n}\sum_{i=1}^{n} \log(1 + e^{-y_i(w^T x_i + b)}) \tag{20}$$

To start we must begin by defining the likelihood for logistic regression. For this case we want the probability to be η when y = 1 and $1 - \eta$ when y = -1. Looking at η for logistic regression,

$$\eta = \frac{1}{1 + e^{-(w^T x + b)}} \tag{21}$$

$$1 - \eta = \frac{e^{-(w^T x + b)}}{1 + e^{-(w^T x + b)}} \tag{22}$$

Now manipulating $1 - \eta$,

$$1 - \eta = \frac{e^{-(w^T x + b)}}{1 + e^{-(w^T x + b)}} \frac{e^{(w^T x + b)}}{e^{(w^T x + b)}} \tag{23}$$

$$1 - \eta = \frac{1}{1 + e^{(w^T x + b)}} \tag{24}$$

Thus we can define our new eta as $\eta = \frac{1}{1+e^{-y(w^T x+b)}}$ and achieve the same result as labels {0,1} with labels {-1,1}. Now defining the likelihood.

$$L(\theta) = \prod_{i=1}^{n} \eta(x_i, y_i; \theta) \tag{25}$$

Now getting the log-likelihood,

$$l(\theta) = \sum_{i=1}^{n} \log\left(\frac{1}{1 + e^{-y(w^T x + b)}}\right) \tag{26}$$

$$l(\theta) = \sum_{i=1}^{n} \log(1) - \log\left(1 + e^{-y(w^T x + b)}\right) \tag{27}$$

$$l(\theta) = -\sum_{i=1}^{n} \log\left(1 + e^{-y(w^T x + b)}\right) \tag{28}$$

Now the negative log-likelihood is simply negating the log-likelihood,

$$nll(\theta) = \sum_{i=1}^{n} \log\left(1 + e^{-y(w^T x + b)}\right) \tag{29}$$

Which is proportional to the ERM solution and is off by a factor of $\frac{1}{n}$

Problem 4

The squared error loss can be used as a surrogate loss but it won't work very well. This is because in classification the outputs represent the probability of each class which would limit the output of an estimator between 0 and 1. This then limits the values that w and b can take on which may not yield a solution with meaningful accuracy. This is why losses for classifications usually transform $w^T x + b$ into the interval of 0 and 1 so the weights and bias can take on a much wider range of values. One can use an exponential loss function where the classes are $\{-1, 1\}$ and the loss is $e^{-yf(x)}$. This encourages the function to be the same sign as the truth as it will produce small values.

Problem 5

a)

Deriving the gradient of the loss function,

$$l(w, w_0) = -\frac{1}{n}\sum_{i=1}^{n}\left[y_i \log\left(\frac{1}{1+e^{-\tilde{w}^T\tilde{x}}}\right) + (1-y_i)\log\left(\frac{e^{-\tilde{w}^T\tilde{x}}}{1+e^{-\tilde{w}^T\tilde{x}}}\right)\right] + \lambda||w||^2 \tag{30}$$

First splitting this into 2 sums,

$$\nabla_{\tilde{w}} l(w, w_0) = \nabla_{\tilde{w}} - \frac{1}{n}\sum_{i=1}^{n} y_i \log\left(\frac{1}{1+e^{-\tilde{w}^T\tilde{x}}}\right) - \frac{1}{n}\sum_{i=1}^{n}(1-y_i)\log\left(\frac{e^{-\tilde{w}^T\tilde{x}}}{1+e^{-\tilde{w}^T\tilde{x}}}\right) + \lambda||w||^2 \tag{31}$$

$$\nabla_{\tilde{w}} l(w, w_0) = -\frac{1}{n}\sum_{i=1}^{n} y_i \nabla_{\tilde{w}} \log\left(\frac{1}{1+e^{-\tilde{w}^T\tilde{x}}}\right) - \frac{1}{n}\sum_{i=1}^{n}(1-y_i)\nabla_{\tilde{w}}\log\left(\frac{e^{-\tilde{w}^T\tilde{x}}}{1+e^{-\tilde{w}^T\tilde{x}}}\right) + \nabla_{\tilde{w}}\lambda||w||^2 \tag{32}$$

Now defining some variables for chain rule,

$$a = \tilde{w}^T\tilde{x} \tag{33}$$

$$b = \frac{1}{1+e^{-\tilde{w}^T\tilde{x}}} \tag{34}$$

$$c = \frac{e^{-\tilde{w}^T\tilde{x}}}{1+e^{-\tilde{w}^T\tilde{x}}} \tag{35}$$

Now rewriting,

$$\nabla_{\tilde{w}} l(w, w_0) = -\frac{1}{n}\sum_{i=1}^{n} y_i \nabla_{\tilde{w}} \log(b) - \frac{1}{n}\sum_{i=1}^{n}(1-y_i)\nabla_{\tilde{w}}\log(c) + \nabla_{\tilde{w}}\lambda||w||^2 \tag{36}$$

$$\nabla_{\tilde{w}} l(w, w_0) = -\frac{1}{n}\sum_{i=1}^{n} y_i \nabla_{\tilde{w}} a\frac{\partial b}{\partial a}\frac{\partial}{\partial b}\log(b) - \frac{1}{n}\sum_{i=1}^{n}(1-y_i)\nabla_{\tilde{w}} a\frac{\partial c}{\partial a}\frac{\partial}{\partial c}\log(c) + \nabla_{\tilde{w}}\lambda||w||^2 \tag{37}$$

Splitting it into more manageable sections I first take the gradient of a wrt $\tilde{w}$

$$\nabla_{\tilde{w}} a = \nabla_{\tilde{w}}\tilde{w}^T\tilde{x}_i \tag{38}$$

$$\nabla_{\tilde{w}} a = \tilde{x}_i \tag{39}$$

Now evaluating $\frac{\partial b}{\partial a}$,

$$\frac{\partial b}{\partial a} = \frac{\partial}{\partial a}(1+e^{-a})^{-1} \tag{40}$$

$$\frac{\partial b}{\partial a} = \frac{e^{-a}}{(1+e^{-a})^2} \tag{41}$$

$$\frac{\partial b}{\partial a} = \frac{1+e^{-a}-1}{(1+e^{-a})^2} \tag{42}$$

$$\frac{\partial b}{\partial a} = \frac{1}{1+e^{-a}} - \frac{1}{(1+e^{-a})^2} \tag{43}$$

$$\frac{\partial b}{\partial a} = b(1-b) \tag{44}$$

Now evaluating $\frac{\partial c}{\partial a}$

$$\frac{\partial c}{\partial a} = \frac{\partial}{\partial a}\frac{e^{-a}}{1+e^{-a}} \tag{45}$$

$$\frac{\partial c}{\partial a} = \frac{\partial}{\partial a}(1 - \frac{1}{1+e^{-a}}) \tag{46}$$

$$\frac{\partial c}{\partial a} = \frac{\partial}{\partial a} - (1+e^{-a})^{-1} \tag{47}$$

$$\frac{\partial c}{\partial a} = -\frac{e^{-a}}{(1+e^{-a})^2} \tag{48}$$

$$\frac{\partial c}{\partial a} = -b(1-b) \tag{49}$$

Now putting these parts all together,

$$\nabla_{\tilde{w}} l(w, w_0) = -\frac{1}{n}\sum_{i=1}^{n} y_i \tilde{x}_i b(1-b)\frac{1}{b} - \frac{1}{n}\sum_{i=1}^{n}(1-y_i)\tilde{x}_i(-b(1-b))\frac{1}{1-b} + \nabla_{\tilde{w}}\lambda||w||^2 \tag{50}$$

$$\nabla_{\tilde{w}} l(w, w_0) = -\frac{1}{n}\sum_{i=1}^{n}\left[y_i\tilde{x}_i(1-b) - (1-y_i)\tilde{x}_i b\right] + \nabla_{\tilde{w}}\lambda||w||^2 \tag{51}$$

$$\nabla_{\tilde{w}} l(w, w_0) = -\frac{1}{n}\sum_{i=1}^{n}\tilde{x}_i\left[y_i(1-b) - (1-y_i)b\right] + \nabla_{\tilde{w}}\lambda||w||^2 \tag{52}$$

$$\nabla_{\tilde{w}} l(w, w_0) = -\frac{1}{n}\sum_{i=1}^{n}\tilde{x}_i\left[y_i - y_i b - b + y_i b\right] + \nabla_{\tilde{w}}\lambda||w||^2 \tag{53}$$

$$\nabla_{\tilde{w}} l(w, w_0) = -\frac{1}{n}\sum_{i=1}^{n}\tilde{x}_i(y_i - b) + \nabla_{\tilde{w}}\lambda||w||^2 \tag{54}$$

Now since $\tilde{w}$ includes the bias term we can split this gradient into one with respect to the weights and one with respect to the bias,

$$\nabla_{w} l(w, w_0) = -\frac{1}{n}\sum_{i=1}^{n} x_i(y_i - b) + \nabla_{w}\lambda||w||^2 \tag{55}$$

$$\nabla_{w} l(w, w_0) = -\frac{1}{n}\sum_{i=1}^{n} x_i(y_i - b) + \nabla_{w}\lambda w^T w \tag{56}$$

$$\nabla_{w} l(w, w_0) = \frac{1}{n}\sum_{i=1}^{n} x_i(\frac{1}{1+e^{-(w^T x_i+\beta)}} - y_i) + 2\lambda w \tag{57}$$

Now for the bias,

$$\nabla_{\beta} l(w, w_0) = -\frac{1}{n}\sum_{i=1}^{n}(y_i - b) + \nabla_{\beta}\lambda||1||^2 \tag{58}$$

$$\nabla_{\beta} l(w, w_0) = \frac{1}{n}\sum_{i=1}^{n}(\frac{1}{1+e^{-(w^T x_i+\beta)}} - y_i) \tag{59}$$

This can also be equivalently expressed in terms of the augmented vectors,

$$\nabla_{\tilde{w}} l(w, w_0) = \frac{1}{n}\sum_{i=1}^{n}\tilde{x}_i\left(\frac{1}{1+e^{-\tilde{w}^T\tilde{x}_i}} - y_i\right) + 2\lambda[0\ w^T]^T \tag{60}$$

Now defining h as a column vector of the sigmoid evaluation and defining λ as $\frac{\lambda}{2}$,

$$\boxed{\nabla_{\tilde{w}} l(w, w_0) = \frac{1}{n}\tilde{X}^T(h - Y) + \lambda[0\ w^T]^T} \tag{61}$$

b)

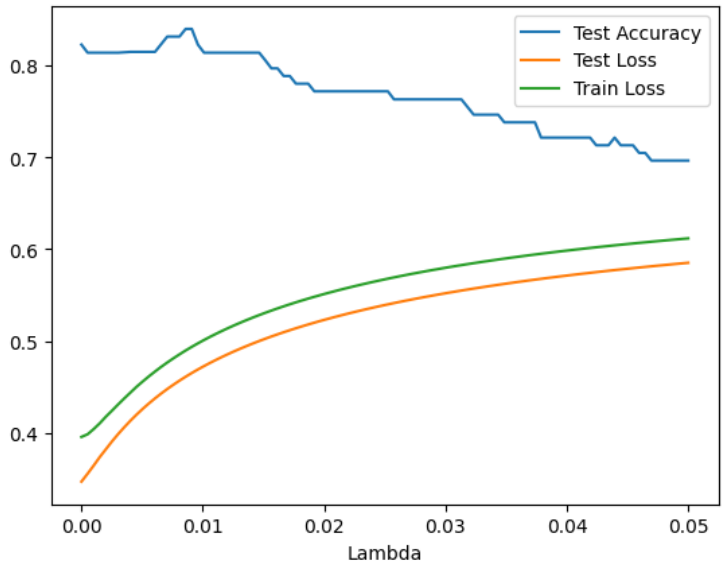
After running the algorithm a few times I found that simply splitting the data set into a test and train set produced wildly different accuracies for different random splits. As a result I employed 6-Fold cross-validation to determine the accuracy of my lambda values.

d)

For optimal parameters I found the best lambda was 0.006, learning rate of 0.1, and Gaussian initialization with mean 0 and stdev $\frac{1}{len(\theta)}$. For Test error I on average I get 83% with a test loss of 0.4460.

e)

For my handmade logistic regression the scale of the lambda values was not in line with what is supplied in the question. As a result I show a plot of lambda values that produce good results,



The plot shows the model is overfit for minimal values of lambda below 0.005 and underfit for values larger than 0.02. However, choosing a lambda value too small has very little effect on the accuracy whereas too large of lambda drastically affects the accuracy.

Problem 6

Using the class conditional density,

$$P(x|C_k) = \frac{1}{(2\pi)^{\frac{p}{2}}|\Sigma_k|^{\frac{1}{2}}} e^{-\frac{1}{2}(x-u_k)^T\Sigma_k^{-1}(x-u_k)} \tag{62}$$

Now forming the likelihood function,

$$L(C_k) = \prod_{i=1}^{n} \frac{1}{(2\pi)^{\frac{p}{2}}|\Sigma_k|^{\frac{1}{2}}} e^{-\frac{1}{2}(x-u_k)^T\Sigma_k^{-1}(x-u_k)} \tag{63}$$

Now the log-likelihood,

$$l(C_k) = \sum_{i=1}^{n} \log\left(\frac{1}{(2\pi)^{\frac{p}{2}}|\Sigma_k|^{\frac{1}{2}}} e^{-\frac{1}{2}(x-u_k)^T\Sigma_k^{-1}(x-u_k)}\right) \tag{64}$$

$$l(C_k) = \sum_{i=1}^{n} \log\left((2\pi)^{-\frac{p}{2}}|\Sigma_k|^{-\frac{1}{2}}\right) + \log\left(e^{-\frac{1}{2}(x-u_k)^T\Sigma_k^{-1}(x-u_k)}\right) \tag{65}$$

$$l(C_k) = \sum_{i=1}^{n} -\frac{p}{2}\log(2\pi) - \frac{1}{2}\log(|\Sigma_k|) + -\frac{1}{2}(x-u_k)^T\Sigma_k^{-1}(x-u_k) \tag{66}$$

Now taking the gradient with respect to u_k,

$$\nabla_{u_k} l(C_k) = \nabla_{u_k} \sum_{i=1}^{n} -\frac{p}{2}\log(2\pi) - \frac{1}{2}\log(|\Sigma_k|) - \frac{1}{2}(x_i-u_k)^T\Sigma_k^{-1}(x_i-u_k) \tag{67}$$

$$\nabla_{u_k} l(C_k) = \sum_{i=1}^{n} \nabla_{u_k} - \frac{1}{2}(x_i-u_k)^T\Sigma_k^{-1}(x_i-u_k) \tag{68}$$

$$0 = \sum_{i=1}^{n} \Sigma_k^{-1}(x_i-u_k) \tag{69}$$

$$0 = \sum_{i=1}^{n} (\Sigma_k^{-1}x_i - \Sigma_k^{-1}u_k) \tag{70}$$

$$0 = \sum_{i=1}^{n} \Sigma_k^{-1}x_i - n\Sigma_k^{-1}u_k \tag{71}$$

$$n\Sigma_k^{-1}u_k = \sum_{i=1}^{n} \Sigma_k^{-1}x_i \tag{72}$$

$$\boxed{u_k = \frac{1}{n}\sum_{i=1}^{n} x_i} \tag{73}$$

Now taking the gradient with respect to Σ^{-1},

$$\nabla_{\Sigma_k} l(C_k) = \nabla_{\Sigma_k}\left(\frac{n}{2}\log(|\Sigma_k|^{-1}) - \sum_{i=1}^{n}\frac{1}{2}(x_i-u_k)^T\Sigma_k^{-1}(x_i-u_k)\right) \tag{74}$$

Using the property $|A| = \frac{1}{|A^{-1}|}$ and $\frac{1}{|A|} = |A^{-1}|$

$$\nabla_{\Sigma_k} l(C_k) = \nabla_{\Sigma_k}\left(\frac{n}{2}\log(|\Sigma_k^{-1}|) - \sum_{i=1}^{n}\frac{1}{2}(x_i-u_k)^T\Sigma_k^{-1}(x_i-u_k)\right) \tag{75}$$

Now using the property $\nabla_A \log(|A|) = (A^{-1})^T$ and $\Sigma = \Sigma^T$

$$\nabla_{\Sigma_k} l(C_k) = \frac{n}{2}\Sigma_k - \frac{1}{2}\sum_{i=1}^{n}\nabla_{\Sigma_k}(x_i-u_k)^T\Sigma_k^{-1}(x_i-u_k) \tag{76}$$

Now since $(x_i-u_k)^T\Sigma_k^{-1}(x_i-u_k)$ is a scalar, the trace is not changed by cyclic permutations (Or moving an end matrix to the other side $tr[ABC] = tr[BCA] = tr[CAB]$).

$$\nabla_{\Sigma_k} l(C_k) = \frac{n}{2}\Sigma_k - \frac{1}{2}\sum_{i=1}^{n} \nabla_{\Sigma_k} tr[(x_i - u_k)(x_i - u_k)^T \Sigma_k^{-1}] \tag{77}$$

$$\nabla_{\Sigma_k} l(C_k) = \frac{n}{2}\Sigma_k - \frac{1}{2}\sum_{i=1}^{n} (x_i - u_k)(x_i - u_k)^T \tag{78}$$

$$\nabla_{\Sigma_k} l(C_k) = \frac{n}{2}\Sigma_k - \frac{1}{2}\sum_{i=1}^{n} (x_i - u_k)(x_i - u_k)^T \tag{79}$$

$$0 = \frac{n}{2}\Sigma_k - \frac{1}{2}\sum_{i=1}^{n} (x_i - u_k)(x_i - u_k)^T \tag{80}$$

$$\frac{n}{2}\Sigma_k = \frac{1}{2}\sum_{i=1}^{n} (x_i - u_k)(x_i - u_k)^T \tag{81}$$

$$\boxed{\Sigma_k = \frac{1}{n}\sum_{i=1}^{n} (x_i - u_k)(x_i - u_k)^T} \tag{82}$$

Now forming the Bayes classifier for this conditional distribution,

$$f(x) \begin{cases} 1 & P(x|C_1)P(C_1) \geq P(x|C_2)P(C_2) \\ 0 & \text{otherwise} \end{cases} \tag{83}$$

Now taking the log of our decision for 1,

$$\log(P(x|C_1)) + \log(P(C_1)) \geq \log(P(x|C_2)) + \log(P(C_2)) \tag{84}$$

$$\log(P(x|C_1)) - \log(P(x|C_2)) \geq \log(P(C_2)) - \log(P(C_1)) \tag{85}$$

Now defining in the conditional probabilities,

$$\log(P(x|C_1)) = -\frac{p}{2}\log(2\pi) - \frac{1}{2}\log(|\Sigma_1|) - \frac{1}{2}(x - u_1)^T \Sigma_1^{-1}(x - u_1) \tag{86}$$

$$\log(P(x|C_0)) = -\frac{p}{2}\log(2\pi) - \frac{1}{2}\log(|\Sigma_0|) - \frac{1}{2}(x - u_0)^T \Sigma_0^{-1}(x - u_0) \tag{87}$$

Now if the covariance matrices are the same or classifier inequality is,

$$-\frac{1}{2}(x - u_1)^T \Sigma_1^{-1}(x - u_1)) + \frac{1}{2}(x - u_0)^T \Sigma_0^{-1}(x - u_0)) \geq \log(P(C_2)) - \log(P(C_1)) \tag{88}$$

Now expanding the quadratic term individually,

$$(x - u_1)^T \Sigma_1^{-1}(x - u_1)) \tag{89}$$

$$(x^T - u_1^T)(\Sigma_1^{-1}x - \Sigma_k^{-1}u_1) \tag{90}$$

$$x^T \Sigma_1^{-1} x - u_1^T \Sigma_1^{-1} x - x^T \Sigma_1^{-1} u_1 + u_1^T \Sigma_1^{-1} u_1 \tag{91}$$

$$x^T \Sigma_1^{-1} x - 2x^T \Sigma_1^{-1} u_1 + u_1^T \Sigma_1^{-1} u_1 \tag{92}$$

Now substituting this result in I get,

$$\frac{1}{2}\left(-x^T \Sigma_1^{-1} x + 2x^T \Sigma_1^{-1} u_1 - u_1^T \Sigma_1^{-1} u_1 + x^T \Sigma_1^{-1} x - 2x^T \Sigma_0^{-1} u_0 + u_0^T \Sigma_0^{-1} u_0\right) \tag{93}$$

$$\frac{1}{2}\left(2x^T \Sigma_1^{-1} u_1 - u_1^T \Sigma_1^{-1} u_1 - 2x^T \Sigma_0^{-1} u_0 + u_0^T \Sigma_0^{-1} u_0\right) \tag{94}$$

$$\frac{1}{2}\left(2x^T \Sigma_1^{-1}(u_1 - u_0) - u_1^T \Sigma_1^{-1} u_1 + u_0^T \Sigma_0^{-1} u_0\right) \tag{95}$$

Now bringing the the inequality back in,

$$\frac{1}{2}\left(2x^T\Sigma^{-1}(u_1-u_0)-u_1^T\Sigma^{-1}u_1+u_0^T\Sigma^{-1}u_0\right)\geq \log(P(C_2))-\log(P(C_1)) \tag{96}$$

Where the estimated values would be plugged in for each value of Σ and u

$$\boxed{f(x)=\begin{cases}1 & \frac{1}{2}\left(2x^T\Sigma^{-1}(u_1-u_0)-u_1^T\Sigma^{-1}u_1+u_0^T\Sigma^{-1}u_0\right)\geq \log(P(C_0))-\log(P(C_1))\\ 0 & otherwise\end{cases}} \tag{97}$$

b)

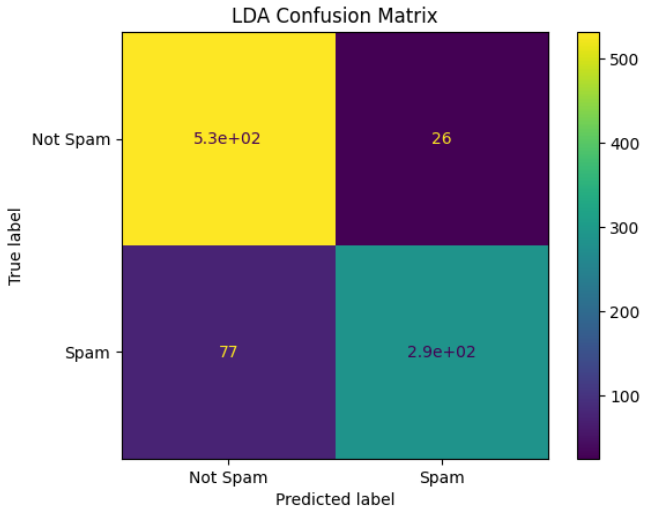
If we relax the constraint that $\Sigma_0=\Sigma_1$ we get a classifier of the form,

$$\boxed{f(x)=\begin{cases}1 & \frac{1}{2}\left(-x^T\Sigma_1^{-1}x+2x^T\Sigma_1^{-1}u_1-u_1^T\Sigma_1^{-1}u_1+x^T\Sigma_1^{-1}x-2x^T\Sigma_0^{-1}u_0+u_0^T\Sigma_0^{-1}u_0\right)\geq \log(P(C_0))-\log(P(C_1))\\ 0 & otherwise\end{cases}} \tag{98}$$

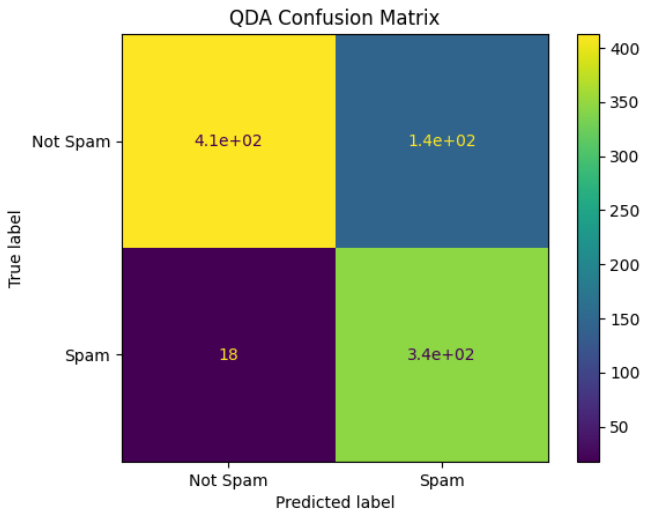
It gets this name because the $x^T\Sigma^{-1}x$ terms can no longer be canceled out. This then results in a quadratic matrix multiplication leading to Quadratic Discriminant analysis.

Problem 7

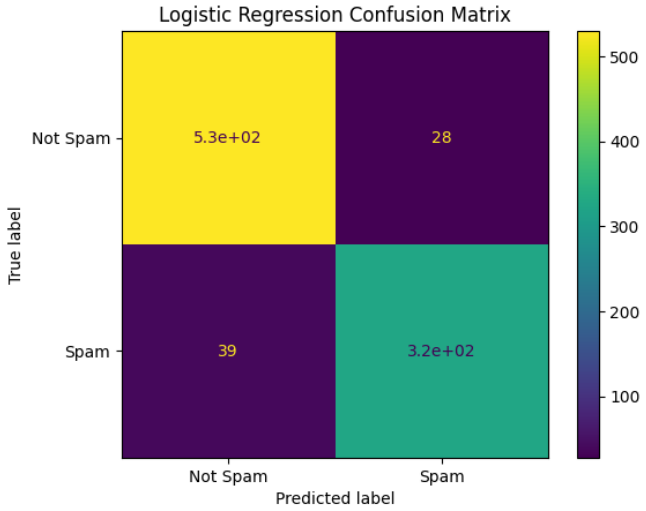
LDA Result



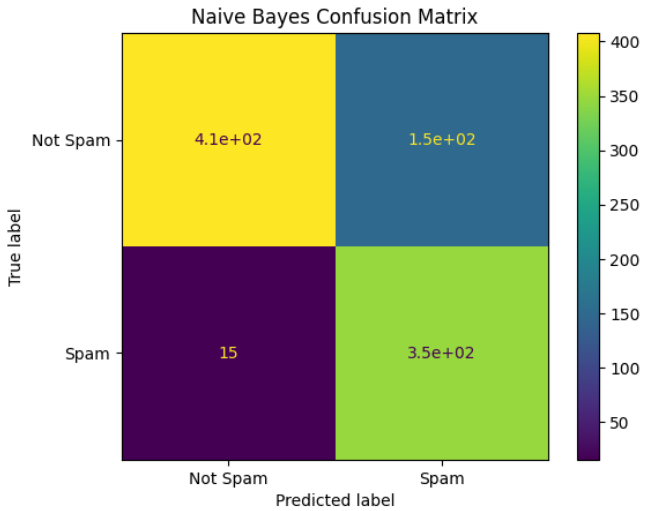
QDA Result



Logisitic Regression Result



Naive Bayes Result



From these results, we see the logistic regressor fits the data the best by a very small margin over LDA. For a sanity check, we would expect predicting only the most frequent class to yield an accuracy of 61%.