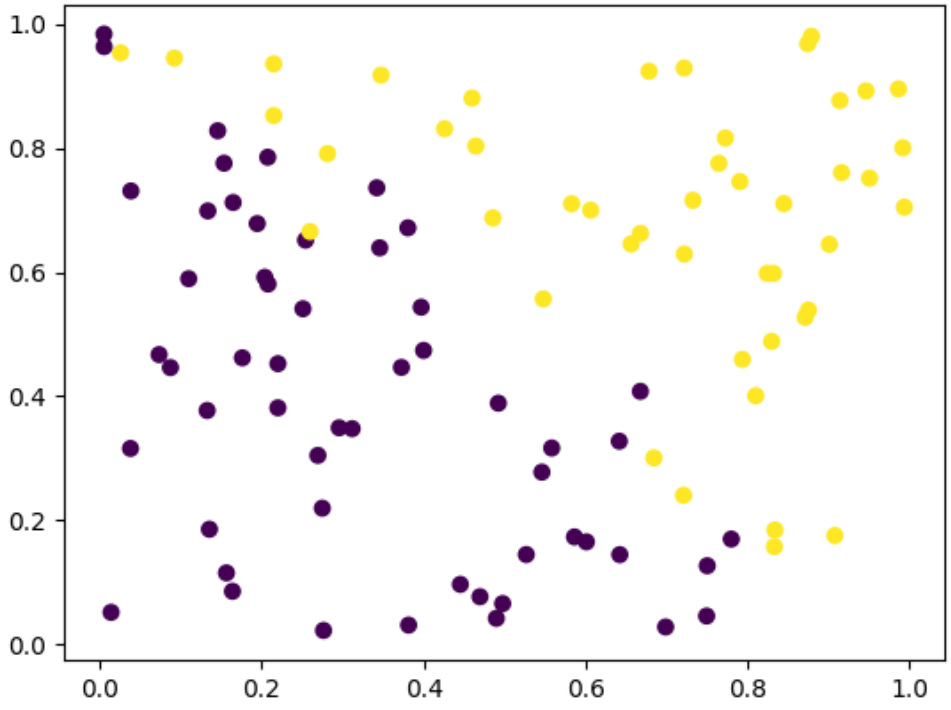
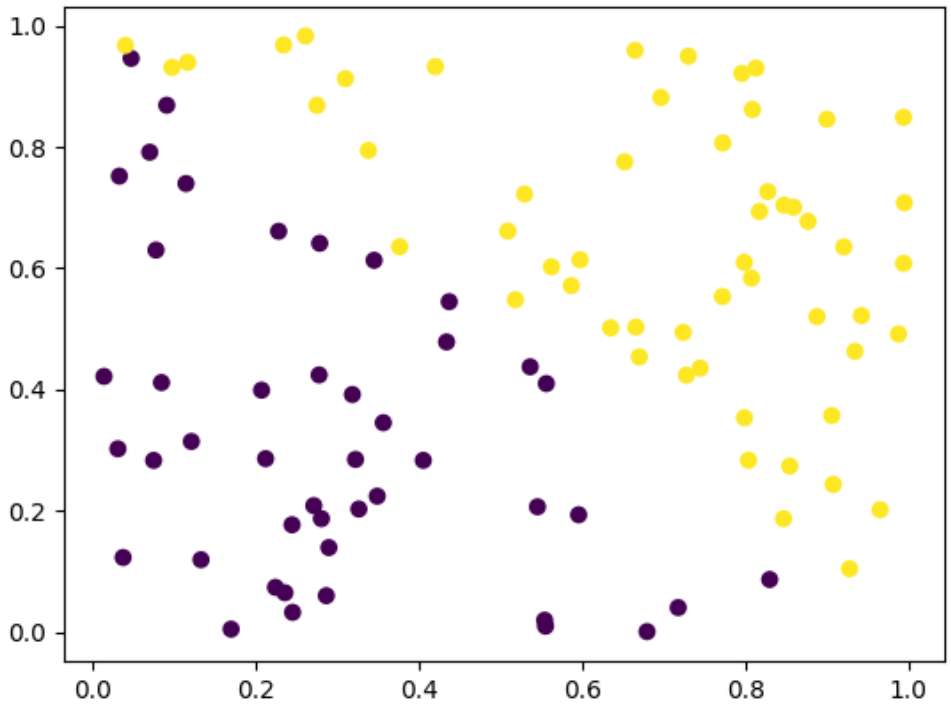
1)

a) La diferencia más notable a la hora de correr el código y entrenar el modelo es que en el caso del dataset A la convergencia llega rápidamente y en cambio cuando se entrena sobre el conjunto B este tarda demasiado en converger.

b) Lo realizado para intentar comprender esta diferencia fue graficar mediante gráficos de puntos ambos datasets. Al observar los gráficos lo más notable al momento de comparar es el hecho de que en el conjunto B podríamos clasificar casi perfectamente los datos con un correcto borde de decisión lineal. A continuación los gráficos de puntos:





Si observamos las figuras con detalle podríamos notar que en el dataset “b” a diferencia del dataset “a” se podría separar/clasificar casi perfectamente los datos con un correcto y justo borde de decisión lineal. Esta diferencia podría ser la causa por la cual la convergencia tarda tanto en el conjunto “b”.

c) Lo que determina el valor de la taza de aprendizaje es que tan rápido puede convergerse hacia los valores óptimos de los parámetros del modelo. Lo que se establece es que un LR que es muy chico el modelo puede converger lento y tomarse más iteraciones para llegar a una solución óptima y por otra parte un LR grande puede hacer que el modelo no converja, ósea los parámetros nunca llegan a valores óptimos. En el caso para el trabajos se probó estableciendo la variable de “learnig rate” tanto con valores altos, bajos y otros varios y no fue solución para que el dataset “b”.

En lo que es escalado de datos, es importante ya que esto ayuda a que los modelos converjan más rápido y además de manera más estable. Si las características no están en la misma escala, algunas de estas pueden dominar sobre otras en términos de contribución a la función de coste, lo

que puede hacer que el proceso de optimización sea más lento. Se ha probado con escalar las Xs previo a realizar el entrenamiento pero tampoco tuvo resultados positivos en la velocidad de convergencia.

Al aplicar un término de regularización a la función de costo se penalizan los valores grandes de los coeficientes del modelo. Lo que se hizo fue aplicar dicho término a la función de costo como un valor lambda según vimos en clase y probando con varios valores se logró que el dataset “b” converja y a su vez que esto no afecte gravemente en el conjunto “a”.

d) No, las SVM en realidad son menos propensas a sufrir problemas de convergencia ya que son menos sensibles a la elección de los hiperparametros o a la calidad de los datos (en comparación de la regresión logística). La función de pérdida de Hinge, utilizada por las SVM, están diseñadas para maximizar el margen entre las clases, lo que significa que se enfoca en los datos cercanos al borde de decisión que están clasificados incorrectamente.

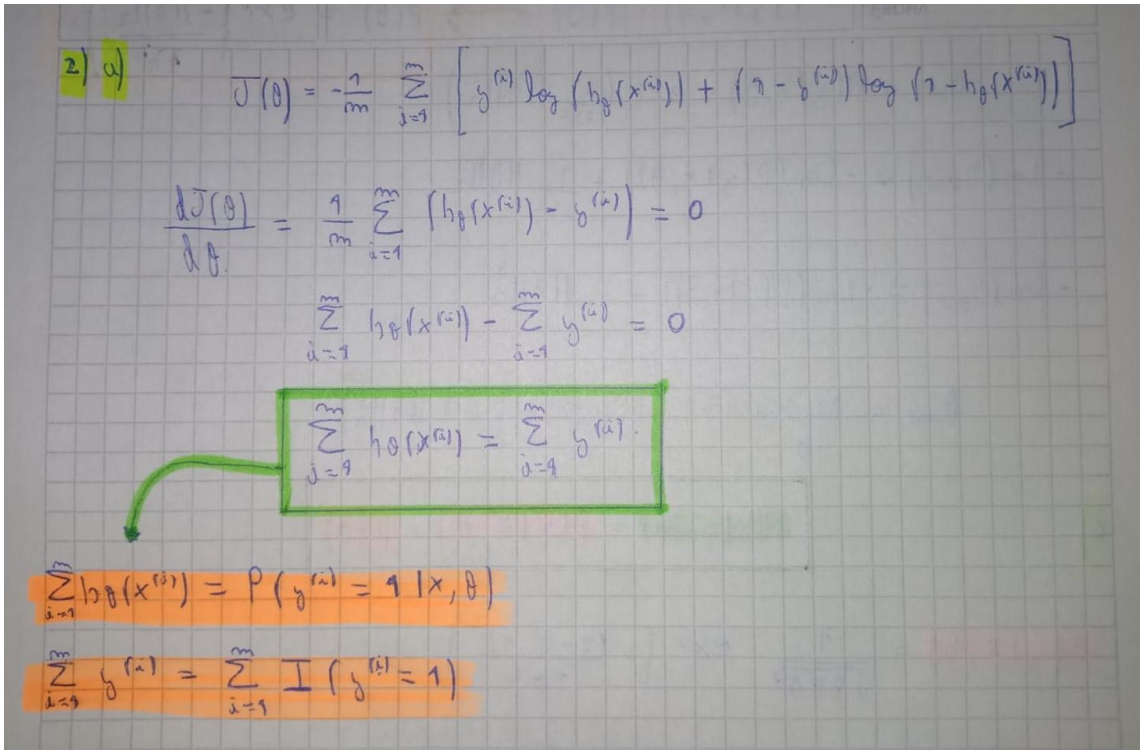
2)

a) El desarrollo matemático está plasmado en la hoja. Partimos de la función de costo que ya conocemos de antes de haberla utilizado en Análisis multivariado y a lo largo de la cursada actual. Lo que hacemos con ella es buscar el gradiente que como ya sabemos es derivar a la función de costo con respecto a θ. Una vez obtenido el gradiente pasamos a realizar la distributiva de la sumatoria y nos quedan dos términos, por un lado tenemos a la sumatoria de la sigmoide y por otro a la sumatoria negativa de las probabilidades estimadas. Por ultimo pasamos hacia el otro lado de la igualdad el segundo término para tenerlo positivo. Entonces tenemos la siguiente igualdad:

$$\sum_{i=1}^{m} h_\theta\left(x^{(i)}\right) = \sum_{i=1}^{m} y^{(i)}$$

Si tenemos en cuenta lo detallado en el enunciado y teoría previamente vista podemos decir por un lado que $h_\theta\left(x^{(i)}\right) = P(y^{(i)} = 1|x, \theta)$ y por

otro que $\sum_{i=1}^{m} y^{(i)} = \sum_{i=1}^{m} I(y^{(i)} = 1)$ y con esto damos por demostrada la propiedad.



b) Esto no es necesariamente así ya que podemos plantear y destacar la diferencia entre que un modelo perfectamente calibrado y un modelo perfectamente preciso. Por el lado de la calibración según dice el ejercicio tenemos que cuando las probabilidades generadas por un modelo coinciden con la observación empírica estamos frente a un modelo bien calibrado. Es decir si el modelo predice una probabilidad del 70% alrededor del 70% de las muestras con esa probabilidad deberían clasificarse como positivas. Y por el lado de la precisión lo que sabemos es que tan bien el modelo clasifica cada muestra, por ejemplo si estamos ante una precisión perfecta el modelo clasifica correctamente todas las muestras sin cometer error alguno.

La relación que establecemos es que un modelo puede estar perfectamente calibrado sin necesariamente tener una precisión perfecta, es decir el modelo aún puede cometer errores de clasificación según el umbral de decisión que se elija, más allá de que las probabilidades predichas son precisas en términos de la probabilidad real. Y por otro lado, un modelo puede ser perfectamente preciso pero no estar correctamente calibrado, en

este caso el modelo clasificará todas las muestras correctamente pero sus probabilidades predichas no reflejan correctamente las probabilidades reales.

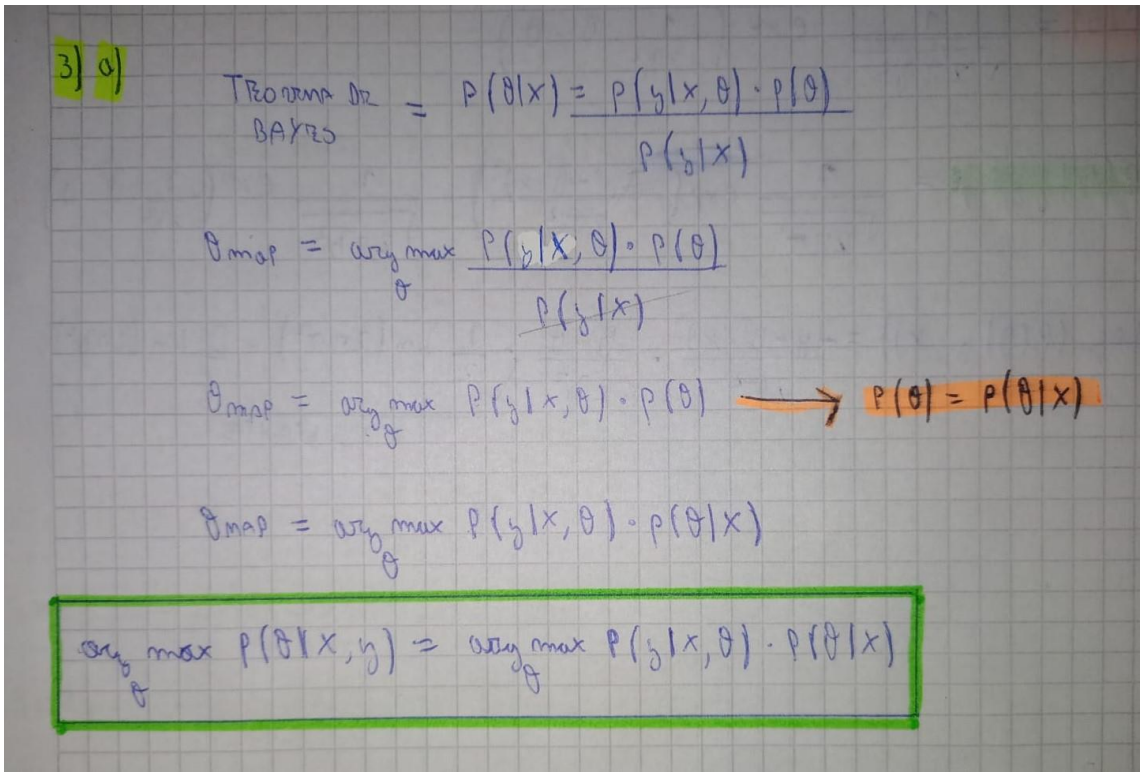
c) Según lo investigado la regularización L2 en Reg. Log. Es un término adicional que se le agrega a la función de costo. En este caso la función de costo se compone por dos partes, el término log-loss, que es la perdida logarítmica, y el término de regularización L2.

$$J(\theta) = -\frac{1}{m}\sum_{i=1}^{m}\left[y^{(i)}\log\left(h_\theta\left(x^{(i)}\right)\right) + \left(1 - y^{(i)}\right)\log\left(1 - h_\theta\left(x^{(i)}\right)\right)\right] + \frac{\lambda}{2\mathrm{m}}\sum_{j=1}^{n}\theta_j^2$$

Al incluir esta modificación, la regularización L2 puede afectar de varias maneras a la calibración. Puede aportar mayor estabilidad ya que L2 tiende a reducir la varianza del modelo al no tener en cuenta a los coeficientes grandes. También mejora cuando estamos frente a casos de overfitting, es decir cuando el modelo tiende a ajustarse demasiado a los datos de entrenamiento L2 ayuda a que el modelo se confíe en sus predicciones. Cabe destacar que estas mejoras de L2 están atadas a la elección del parámetro lambda ya que si este es grande el modelo puede volverse más conservador y perder algo de precisión y en el caso contrario donde es pequeño el modelo puede tender a sobre ajustar (hacer overfitting) y perder calibración.

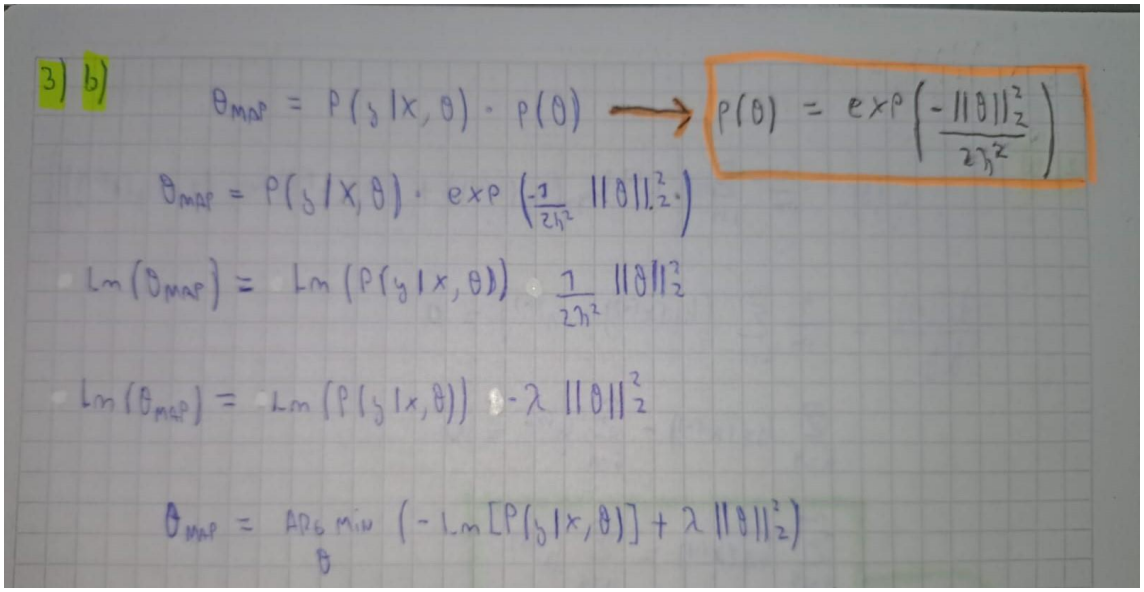
3)

a)



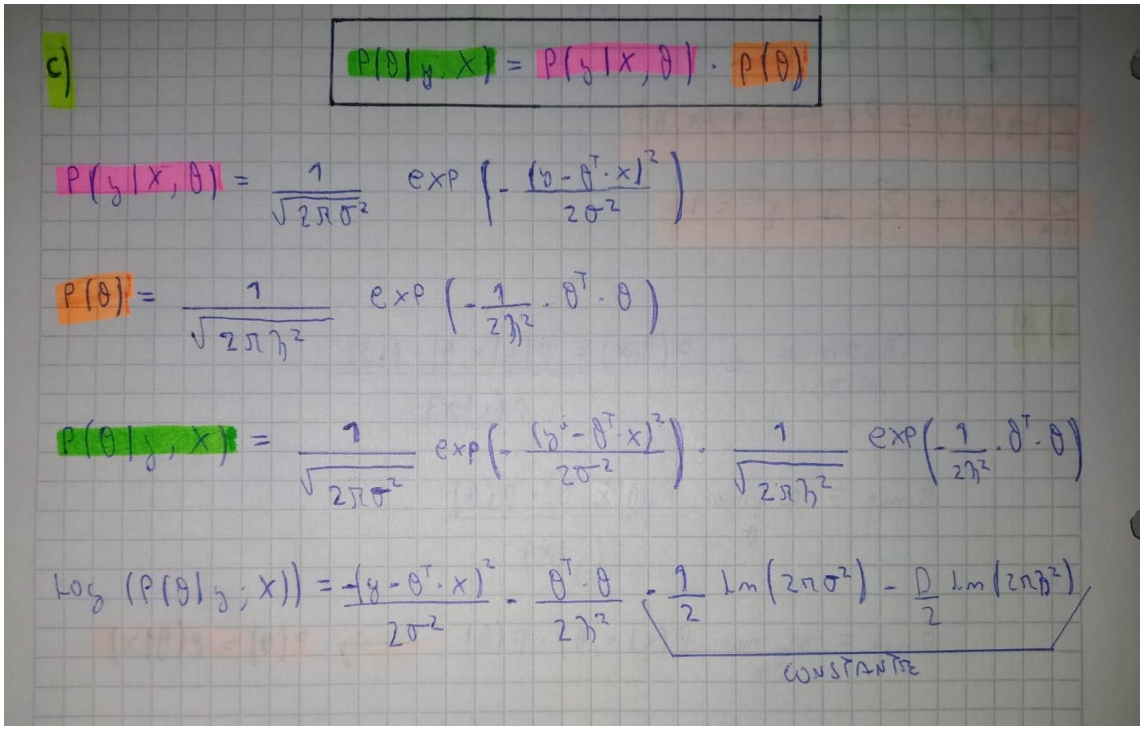
Lo que hacemos aquí es partir del teorema de Bayes, luego ignoramos o apartamos el denominador ya que no contiene ni depende de theta. Luego aprovechando lo detallado en la consigna, remplazamos al prior por la probabilidad de theta dado X.

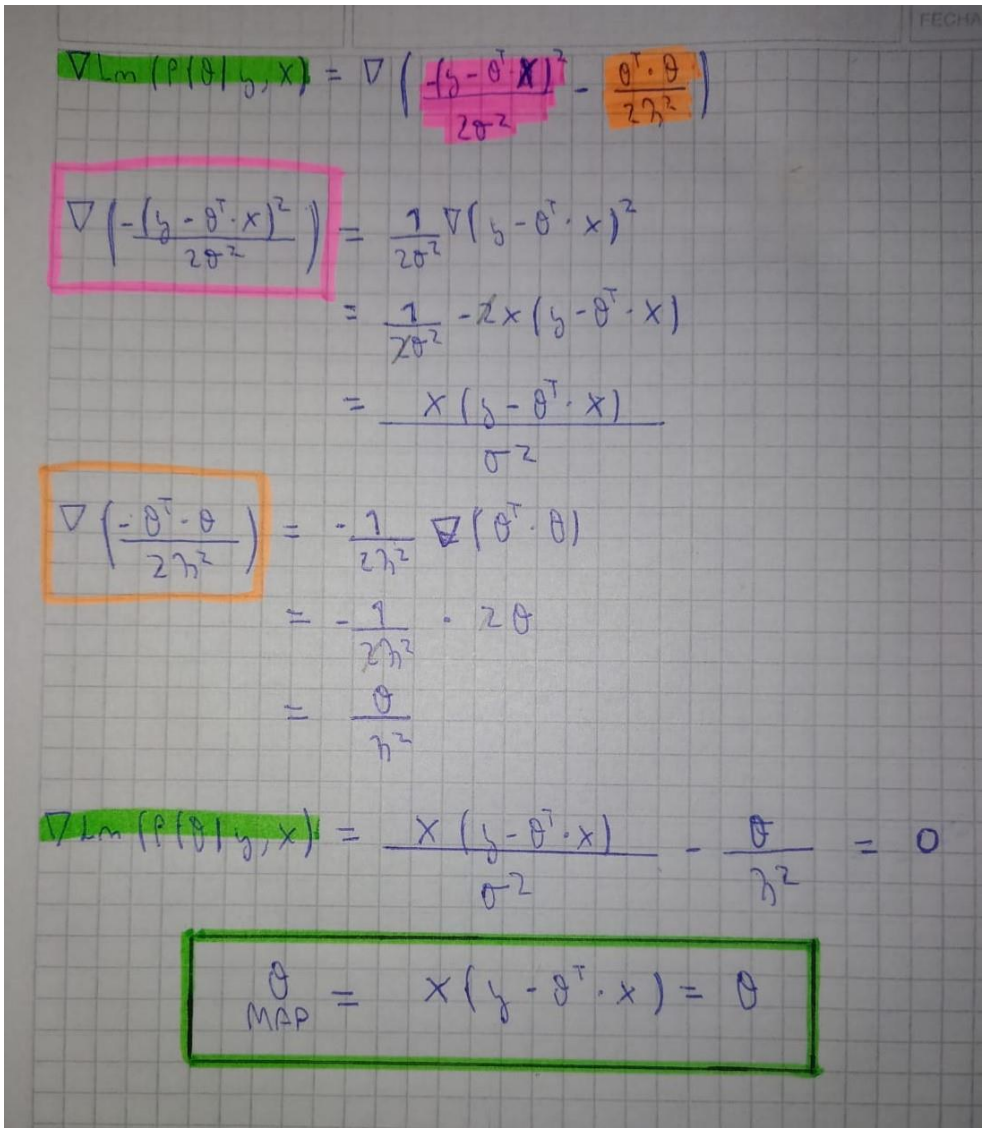
b)



Partiendo de lo hecho en el inciso (a), continuamos tomando al prior como: $\exp(-\frac{\|\theta\|^{2}_{2}}{2*\eta^{2}})$ y remplazamos en la formula. Lo que hacemos a continuación es aplicar LN y luego separamos en suma aplicando la propiedad logarítmica. Ahora por último si tomamos $-\frac{1}{2*\eta^{2}} = -\lambda$ obtendremos la expresión que queremos buscar que la misma puede ser escrita o reformulada como se ve en la hoja.

c)





Investigando en internet se encontró que podemos comenzar por remplazar los valores del prior y de la probabilidad de “y” dado X y theta. Al tener esos valores lo que hacemos es remplazar en la fórmula de $P(\theta|x,y$. Ahora para encontrar la función cerrada nos conviene utilizar el logaritmo natural ya que estamos en una distribución gaussiana. Aplicamos y luego de obtener el desarrollo ignoramos todos los términos constantes. Lo último que queda por hacer es derivar lo obtenido. Separamos en términos y derivamos. Luego de obtener la derivada despejamos Theta y obtenemos lo resaltado en la hoja.

d) ----

4)

b) Para calcular el factor de compresión se utilizaron los bits de cada imagen. Para calcular los bits, se obtuvo la resolución de la imagen y se lo multiplica por la cantidad de bits por pixel (y por canal de color), es decir, 24(8x3). Luego, los bits de la imagen comprimida es la resolución por el logaritmo en base 2 de 16 (lo que calcula los bits por pixel teniendo en cuenta esa cantidad de colores), es decir, 4 bits por pixel. Al obtener esta cuenta, se divide la cantidad de bits de la imagen original por la cantidad de bits de la imagen comprimida, y se llega a que el factor de compresión al disminuir la cantidad de colores a 16 en total es de 6.