Question 1

1. False
2. False
3. True
4. False (I think it is False, -> case when only 1 feature then don't need to scale)
5. False (dimensionality reduction method)
6. True (only non-negative eigenvectors required)
7. False (L1 regularization can shrink features to zero)
8. False (not linear regression as a whole but log models are invariant to scaling)
9. False
10. True
11. True

Question 2

Part (a)

$$Posterior = \frac{Likelihood\ x\ Prior}{Evidence}$$

Part (b)

Posterior probability is the probability an event will happen after all evidence or background information has been taken into account. On the other hand, the prior probability is the probability an event will happen before taking any new evidence into account. Therefore, the posterior probability can be seen as an adjustment, using the given likelihood, on the prior probability and thus holds this relationship:

$$Posterior \propto Likelihood\ x\ Prior$$

Part (c)

Since we are dealing with a poor classification model overall (poor training and testing accuracy), boosting would be a better choice to try to improve the performance since the issue at hand is not over-fitting. Using the boosting ensemble method, each classifier is trained to reduce the errors of the previous ensemble. Boosting also reduces the bias by generating an ensemble of weak classifiers and overall is quite resilient to overfitting, but there is still the possibility of overfitting when using boosting.

Part (d)

Boosting combines many weak, high bias, models in an ensemble that has a lower bias than the individual models, yet the variance is increased. On the other hand, bagging combines strong learners in a way that reduces their variance but at the same time maintains relatively the same bias.

Part (e)

The main modelling assumption in the Naïve Bayes classifier is that it assumes the word features x_i are conditionally independent given some class c.

Part (f)

The sum of squared distances of data points to their assigned cluster centres is the objective function that is minimized.

Let

- $x^{(n)} \in \mathbb{R}^D \; for \; n = 1 \; to \; N$ be the data sample (observed)
- $m_k \in \mathbb{R}^D \; for \; k = 1 \; to \; K$ be the cluster centre (not observed)
- $r^{(n)} \in \mathbb{R}^K \; for \; n = 1 \; to \; N$ be the cluster assignment for sample n (1-of-K encoding, not observed)

The formula is:

$$\min_{\{m_k\},\{r^{(n)}\}} \sum_{n=1}^{N} \sum_{k=1}^{K} r_k^{(n)} \left\| m_k - x^{(n)} \right\|^2, \qquad where \; r_k^{(n)} = \mathbb{I}\{x^{(n)} \; is \; assigned \; to \; cluster \; k\}$$

Part (g)

The Bellman optimality equation:

$$Q^*(s,a) = r(s,a) + \gamma \int_S P(s'|s,a) \max_{a' \in A} Q^*(s',a') ds'$$

Question 3

Part (a)

KNN

- K (nearest neighbor)

Part (b)

Deep Neural Networks

- Number of hidden units
- Learning rate

Part (c)

Support Vector Machines

- Kernel
- C (Regularization)
- Gamma

Part (d)

PCA

- K (top K eigenvectors)

Part (e)

K-Means Clustering

- K (number of K-means clusters)

Part (f)

Gaussian Mixture Models

- K (number of mixture clusters)

Part (g)

Q-Learning

- α (learning rate)
- ϵ (exploration parameter)

Question 4

Part (a)

Let

- T be a test that is positive
- D be a person who has the disease

P(T|D) = 0.99
P(T|~D) = 0.01
P(D) = 0.001
P(~D) = 0.999

$$\begin{aligned} P(D|T) &= \frac{P(T|D) \cdot P(D)}{P(T)} \\ &= \frac{P(T|D) \cdot P(D)}{P(T|\sim D) \cdot P(\sim D) + P(T|D) \cdot P(D)} \\ &= \frac{0.99 \cdot 0.001}{0.01 \cdot 0.999 + 0.99 \cdot 0.001} \\ &= 0.090163934 \end{aligned}$$

The probability that this person has the disease is 0.090163934, or similarly 9.0163934%.

Part (b)

Let

- T_{old} be a test that is positive using the old laboratory
- T_{new}be a test that is positive using the new laboratory
- D be a person who has the disease

P($T_{old}|D$) = 0.99
P($T_{old}|\sim D$) = 0.01
P($T_{new}|D$) = 0.99
P($T_{new}|\sim D$) = 0.01
P(D) = 0.001

P(~D) = 0.999

Assuming that both laboratory tests are conditionally independent of one another, we have

$$P(D|T_{old}, T_{new}) = \frac{P(T_{old}, T_{new}, |D) \cdot P(D)}{P(T_{old}, T_{new})}$$
$$= \frac{P(T_{old}|T_{new}, D) \cdot P(T_{new}|D) \cdot P(D)}{P(T_{old}, T_{new}|D) \cdot P(D) + P(T_{old}, T_{new}|\sim D) \cdot P(\sim D)}$$
$$= \frac{P(T_{old}|D) \cdot P(T_{new}|D) \cdot P(D)}{\big(P(T_{old}|T_{new}, D) \cdot P(T_{new}|D) \cdot P(D)\big) + \big(P(T_{old}|T_{new}, \sim D) \cdot P(T_{new}|\sim D) \cdot P(\sim D)\big)}$$
$$= \frac{P(T_{old}|D) \cdot P(T_{new}|D) \cdot P(D)}{\big(P(T_{old}|D) \cdot P(T_{new}|D) \cdot P(D)\big) + \big(P(T_{old}|\sim D) \cdot P(T_{new}|\sim D) \cdot P(\sim D)\big)}$$
$$= \frac{0.99 \cdot 0.99 \cdot 0.001}{(0.99 \cdot \ 0.99 \cdot 0.001) + (0.01 \cdot 0.01 \cdot 0.999)}$$
$$= 0.9075$$

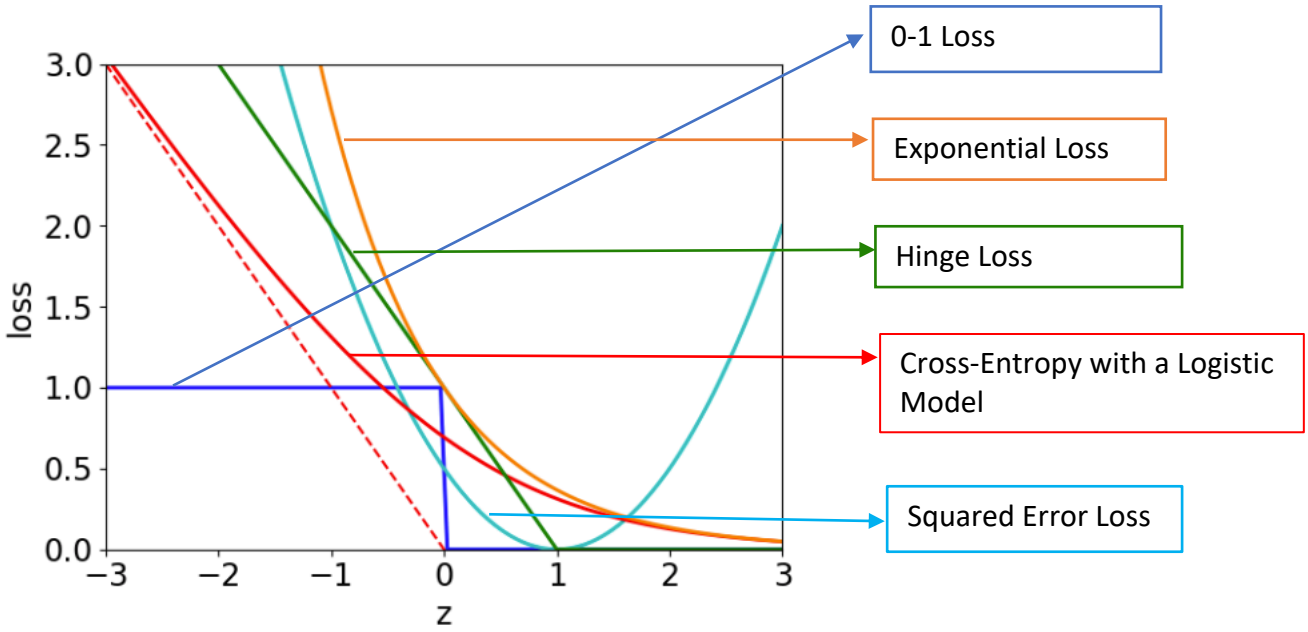
The probability that the person actually has the disease when both laboratories tests are positive is 0.9075, or similarly 90.75%.

Question 5

We can construct and combine several binary classifiers to formulate a multi-class classification problem. We can use the one-versus-all strategy such that each binary classifier would be trained the entire training dataset. Each binary classifier would be used to determine whether a data is classified as 0.0, 0.1, …, 1.0 by taking the highest confidence score of the 11 binary classifiers. A correct classification occurs when any data classifications with a resolution of up to 0.1 in relation to the classifier results. Any predictions/decisions will be made by using each binary classifier and selecting the classification with the highest confidence score.

Question 6

Part (a)



Part (b)

When we take the partial derivative of the 0-1 loss function with respect to the weights, we result in a gradient of 0 everywhere the loss function is defined. Therefore, changing the weights has no effect on the loss since almost any point has 0 gradient.

Part (c)

Support Vector Machine method uses the Hinge loss.

Part (d)

AdaBoost method can be interpreted as using the exponential loss.

Question 7

Part (a)

Define a new weight $w' = w_1 w_2$

The 1-layer NN:

$$y = w'x$$

Part (b)

Forward pass:

- $z = w_1 x$
- $y = w_2 z$
- $L = \frac{1}{2}(y - t)^2$

Backward pass:

- $\bar{L} = 1$
- $\bar{y} = \bar{L}\frac{dL}{dy} = \bar{L}(y - t)$
- $\overline{w_2} = \bar{y}\frac{dy}{dw_2} = \bar{y}(z) = \bar{y}(w_1 x)$
- $\bar{z} = \bar{y}\frac{dy}{dz} = \bar{y}(w_2)$
- $\overline{w_1} = \bar{z}\frac{dz}{dw_1} = \bar{z}(x)$

The gradient of the loss with respect to w_1:

$$\begin{aligned}\frac{dL}{dw_1} &= \bar{z}(x) \\ &= \bar{y}(w_2)(x) \\ &= \bar{L}(y - t)(w_2)(x) \\ &= (y - t)w_2 x\end{aligned}$$

The gradient of loss with respect to w_2:

$$\begin{aligned}\frac{dL}{dw_2} &= \bar{y}(z) \\ &= \bar{L}(y - t)(w_1 x) \\ &= (y - t)w_1 x\end{aligned}$$

Part (c)

If the loss function of the 2-layer NN is convex, we require the second derivative of the loss function with respect to w_1 and w_2 to be non-negative.

$$\begin{aligned}\frac{d^2 L}{dw_1^2} &= \frac{d}{dw_1}\left(\frac{dL}{dw_1}\right) \\ &= \frac{d}{dw_1}[(y - t)w_2 x] \\ &= \frac{d}{dw_1}[(w_2 w_1 x - t)w_2 x]\end{aligned}$$

$$
\begin{aligned}
&= \frac{d}{dw_1}[w_2^2 w_1 x^2 - t w_2 x] \\
&= w_2^2 x^2 \\
&\geq 0, \qquad since\ all\ terms\ are\ squared
\end{aligned}
$$

Since second derivative of the loss function with respect w_1 is non-negative, the loss function of the 2-layer NN with respect to w_1 is convex.

$$
\begin{aligned}
\frac{d^2 L}{dw_2^2} &= \frac{d}{dw_2}\left(\frac{dL}{dw_2}\right) \\
&= \frac{d}{dw_2}[(y - t) w_1 x] \\
&= \frac{d}{dw_2}[(w_2 w_1 x - t) w_1 x] \\
&= \frac{d}{dw_2}[w_2 w_1^2 x^2 - t w_1 x] \\
&= w_1^2 x^2 \\
&\geq 0, \qquad since\ all\ terms\ are\ squared
\end{aligned}
$$

Since second derivative of the loss function with respect w_2 is non-negative, the loss function of the 2-layer NN with respect to w_2 is convex.

Therefore, the loss function of the 2-layer NN is convex with respect to w_1 and w_2, as needed.

Question 8

Part (a)

$$P(x|t=0) = \frac{P(t=0|x) \cdot P(x)}{P(t=0)}$$

$$P(x|t=1) = \frac{P(t=1|x) \cdot P(x)}{P(t=1)}$$

$$P(t=0) = \frac{P(t=0|x) \cdot P(x)}{P(x|t=0)}$$

$$P(t=1) = \frac{P(t=1|x) \cdot P(x)}{P(x|t=1)}$$

Part (b)

- $P(t=0) = 0.5$

- $P(t=1)=0.5$

$$\begin{aligned}P(t=0|x) &= \frac{P(x|t=0)\cdot P(t=0)}{P(x)}\\ &= \frac{P(x|t=0)\cdot P(t=0)}{P(x|t=0)\cdot P(t=0)+P(x|t=1)\cdot P(t=1)}\\ &= \frac{P(x|t=0)\cdot 0.5}{P(x|t=0)\cdot 0.5+P(x|t=1)\cdot 0.5}\\ &= \frac{P(x|t=0)}{P(x|t=0)+P(x|t=1)}\end{aligned}$$

Question 9

Part (a)

$$\begin{aligned}&\underset{k}{\text{argmax}}\log p(t_k|x) = \underset{k}{arg\max}\log\frac{p(x|t_k)p(t_k)}{p(x)}\\ &= \underset{x}{\text{argmax}}(\log p(x|t_k)+\log p(t_k)-\log p(x))\\ &= \underset{k}{\text{argmax}}\left(-\frac{D}{2}\log(2\pi)-\frac{1}{2}\log|\Sigma_k^{-1}|-\frac{1}{2}(x-\mu_k)^T\Sigma_k^{-1}(x-\mu_k)+\log p(t_k)-\log p(x)\right)\end{aligned}$$

Part (b)

Note:

- $\log p(t_k|x) = -\frac{d}{2}\log(2\pi)-\frac{1}{2}\log|\Sigma_k^{-1}|-\frac{1}{2}(x-\mu_k)^T\Sigma_1^{-1}(x-\mu_k)+\log p(t_k)-\log p(x)$
- $C_{1,2}=\log p(t_2)-\log p(t_1)$

$$\begin{gathered}\log p(t_1|x)=\log p(t_2|x)\\ (x-\mu_1)^T\Sigma_1^{-1}(x-\mu_1)+\log p(t_1)=(x-\mu_2)^T\Sigma_2^{-1}(x-\mu_2)+\log p(t_2)\\ (x-\mu_1)^T\Sigma^{-1}(x-\mu_1)=(x-\mu_2)^T\Sigma^{-1}(x-\mu_2)+(\log p(t_2)-\log p(t_1))\\ (x-\mu_1)^T\Sigma^{-1}(x-\mu_1)=(x-\mu_2)^T\Sigma^{-1}(x-\mu_2)+C_{1,2}\\ x^T\Sigma^{-1}x-2\mu_1^T\Sigma^{-1}x=x^T\Sigma^{-1}x-2\mu_2^T\Sigma^{-1}x+C_{1,2}\\ -2\mu_1^T\Sigma^{-1}x=-2\mu_2^T\Sigma^{-1}x+C_{1,2}\end{gathered}$$

Therefore, we can see that it is a linear function in x, meaning the decision boundary between the two classes is linear.

Question 10

Part (a)

The likelihood function $L(\theta)$:

$$L(\theta) = P(D_N|\theta) = \prod_{i=1}^{N} \theta \cdot (1-\theta)^{K_i} = \theta^N \cdot (1-\theta)^{\sum_{i=1}^{N} K_i}$$

Part (b)

The log likelihood function $l(\theta)$:

$$l(\theta) = \log L(\theta) = N \log \theta + \sum_{i=1}^{N} K_i \cdot \log(1-\theta)$$

Part (c)

The maximum likelihood estimator $\hat{\theta}_{ML}$:

$$\begin{aligned}
\frac{dl}{d\theta} &= \frac{d}{d\theta}\left(N \log \theta + \sum_{i=1}^{N} K_i \cdot \log(1-\theta)\right) \\
&= \frac{d}{d\theta}(N \log \theta) + \frac{d}{d\theta}\left(\sum_{i=1}^{N} K_i \cdot \log(1-\theta)\right) \\
&= \frac{N}{\theta} - \sum_{i=1}^{N} K_i \cdot \frac{1}{1-\theta}
\end{aligned}$$

Therefore, setting the derivative equal to 0:

$$\begin{aligned}
\frac{N}{\theta} - \sum_{i=1}^{N} K_i \cdot \frac{1}{1-\theta} &= 0 \\
\frac{N}{\theta} &= \sum_{i=1}^{N} K_i \cdot \frac{1}{1-\theta} \\
\frac{1-\theta}{\theta} &= \sum_{i=1}^{N} K_i \\
\frac{1}{\theta} - 1 &= \sum_{i=1}^{N} K_i
\end{aligned}$$

$$\frac{1}{\theta} = \left(\sum_{i=1}^{N} K_i\right) + 1$$

$$\hat{\theta}_{ML} = \frac{1}{(\sum_{i=1}^{N} K_i) + 1}$$

Part (d)

..

Part (e)

The Maximum A-Posteriori (MAP) estimate $\hat{\theta}_{MAP}$:

$$\begin{aligned}\log p(\theta, D_N) &= \log p(\theta) + \log p(D_N|\theta) \\ &= constant + (a-1)\log\theta + (b-1)\log(1-\theta) + N\log\theta + \sum_{i=1}^{N} K_i \log(1-\theta) \\ &= constant + (N+a-1)\log\theta + \left(\sum_{i=1}^{N} K_i + b - 1\right)\log(1-\theta)\end{aligned}$$

Maximizing by finding a critical point:

$$\begin{aligned} 0 &= \frac{d}{d\theta}\log p(\theta, D_N) \\ 0 &= \frac{N+a-1}{\theta} - \frac{\sum_{i=1}^{N} K_i + b - 1}{1-\theta} \\ \frac{N+a-1}{\theta} &= \frac{\sum_{i=1}^{N} K_i + b - 1}{1-\theta} \\ \frac{1-\theta}{\theta} &= \frac{\sum_{i=1}^{N} K_i + b - 1}{N+a-1} \\ \frac{1}{\theta} &= \frac{\sum_{i=1}^{N} K_i + b - 1}{N+a-1} + 1 \\ \theta &= \frac{1}{\frac{\sum_{i=1}^{N} K_i + b - 1 + N + a - 1}{N+a-1}} \\ \hat{\theta}_{MAP} &= \frac{N+a-1}{N + \sum_{i=1}^{N} K_i + a + b - 2} \end{aligned}$$

Part (f)

The Posterior probability of θ:

$$p(\theta|D_N) = \frac{\Gamma(N + a + \sum_{i=1}^{N} K_i + b)}{\Gamma(N + a)\Gamma(\sum_{i=1}^{N} K_i + b)} \theta^{N+a-1}(1-\theta)^{\sum_{i=1}^{N} K_i + b - 1}$$

$$\theta|D_N \sim Beta(N + a, \sum_{i=1}^{N} K_i + b)$$

Part (g)

The expected value of θ according to the posterior distribution:

$$\mathbb{E}[\theta|D_N] = \frac{N + a}{N + a + \sum_{i=1}^{N} K_i + b}$$

Part (h)

MLE

- Advantages
    - Easy to do in practice, since we can just do gradient descent
- Disadvantages
    - Data sparsity, if there is too little data, it can overfit