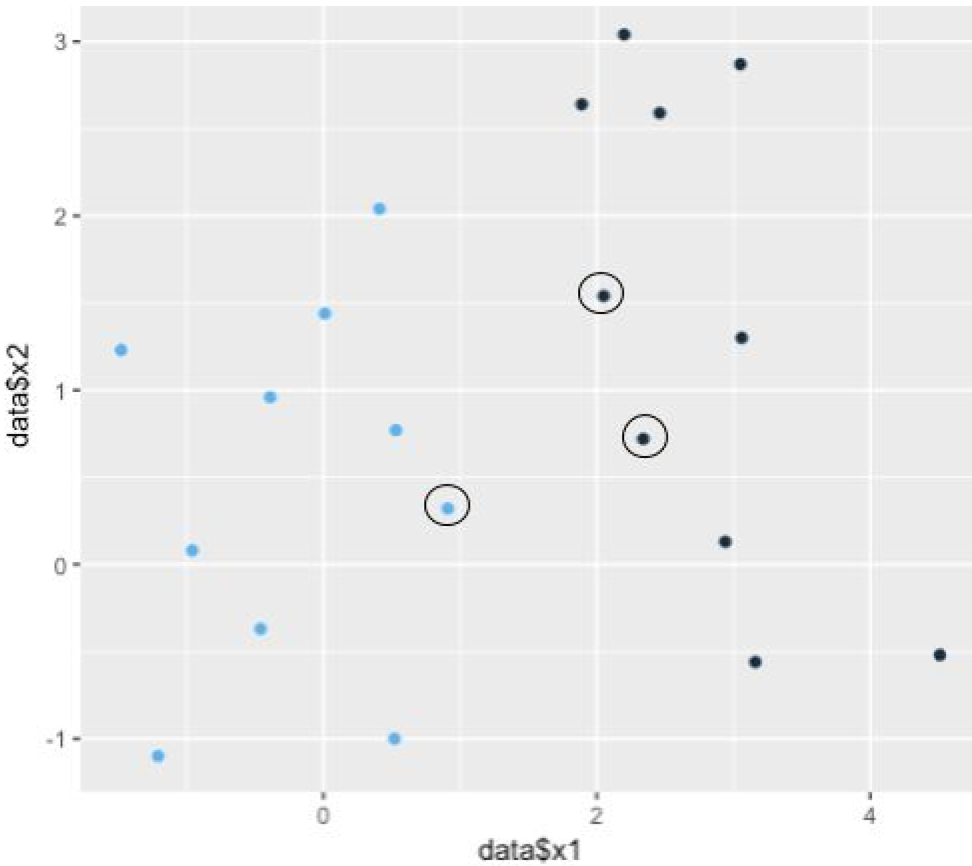
1.

a. Point 2, 18, 19



b. $w = \sum ai \cdot yi \cdot xi$

=
$0.9492 \cdot 1 \cdot [0.91, 0.32]^T + 0.3030 \cdot (-1) \cdot [2.05, 1.54]^T + 0.9053 \cdot (-1) \cdot [2.34, 0.72]^T$
= $[-1.8755, -0.8147]^T$

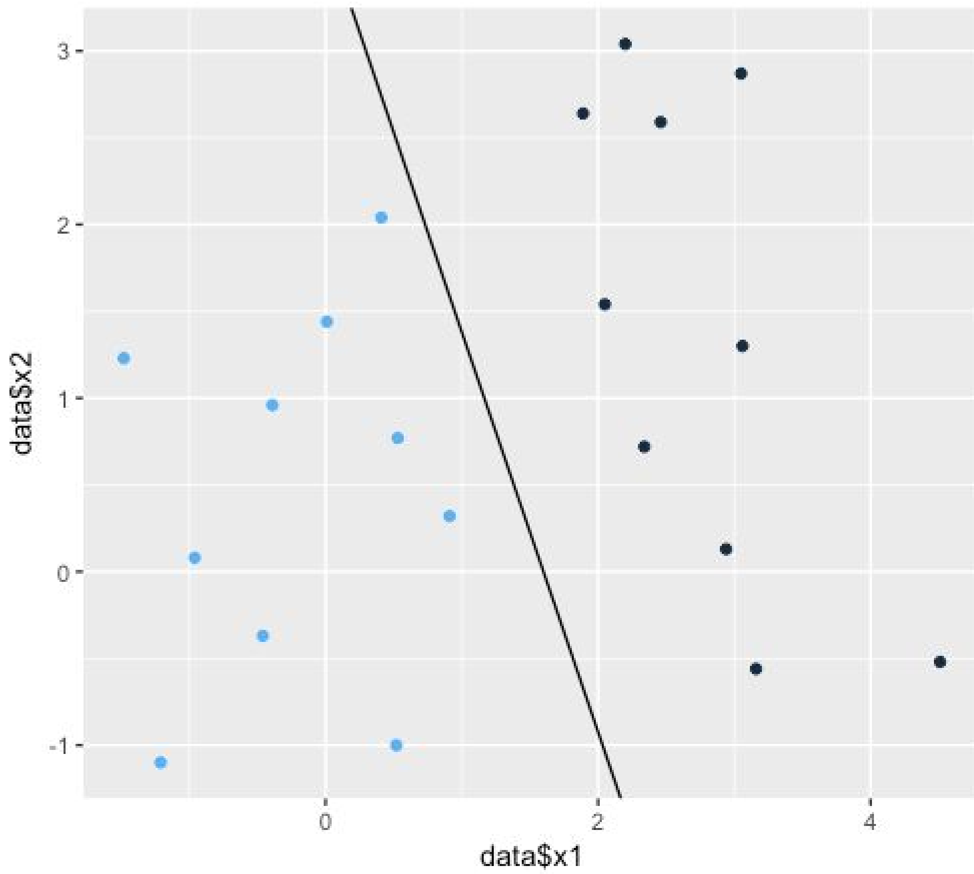
c. $b = \sum(yk - wTxk)/Nk$

= ⅓ * (1 - [-1.8755, -0.8147] * $[0.91, 0.32]^T$ + (-1) - [-1.8755, -0.8147] * $[2.05, 1.54]^T$ + (-1) - [-1.8755, -0.8147] * $[2.34, 0.72]^T$)
= (-1 + 1.9674 + 5.0994 + 4.9753) / 3
= 3.6807

d. f(x) = -1.8755 * x1 - 0.8147 * x2 + 3.6807

e. f(x) = -1.8755 * (-1) - 0.8147 * 2 + 3.6807 = 3.9268
So, the class is 1

f.



2.

a. (p+1) * 3 + (3 + 1) * 4 + (4 + 1) * k = 3p + 5k +19

b.

i. Initialize weights to small random numbers, associated with biases

ii. Repeat until terminating condition meets

1. For each training example
2. Propagate the inputs forward
   a. For a hidden or output layer unit j
      i. Calculate net input: $I_j = \sum_i w_{ij} o_i + \theta_j$
      ii. Calculate output of unit j: $O_j = \sigma(I_j) = \frac{1}{1+e^{-I_j}}$
3. Backpropagate the error
   a. For unit j in output layer: $Err_j = O_j(1 - O_j)(T_j - O_j)$
   b. For unit j in a hidden layer: Errj = Oj(1 - Oj) $\sum_k Err_k w_{jk}$
   c. Update weights: $w_{ij} = w_{ij} + \eta Err_j O_i$
   d. Update bias: $\theta_j = \theta_j + \eta Err_j$

iii. Terminating condition

c. Net input and output calculations

Unit, j	Net Input, I_j	Output, O_j
3	0.4	$1/(1+e^{-0.4}) = 0.599$
4	-0.1	$1/(1+e^{0.1}) = 0.475$
5	-0.2*0.599 - 0.3*0.475 = -0.2623	$1/(1+e^{-0.2623}) = 0.5652$

Calculation of the error at each node

Unit, j	Err_j
5	0.5652*(1-0.5652)*(1-0.5652) = 0.107
4	0.475*(1-0.475)*(0.107)*(-0.3) = -0.008
3	0.599*(1-0.599)*(0.107)*(-0.2) = -0.005

Calculations for weight and bias updating

Weight or Bias	New Value
w_{35}	-0.2 + 0.8 * 0.107 * 0.599 = -0.149
w_{45}	-0.3 + 0.8 * 0.107 * 0.475 = -0.259
w_{13}	-0.3 + 0.8 * (-0.005) * 0 = -0.3
w_{14}	0.2 + 0.8 * (-0.008) * 0 = 0.2
w_{23}	0.4 + 0.8 * (-0.005) * 1 = 0.396
w_{24}	-0.1 + 0.8 * (-0.008) * 1 = -0.1064
θ_5	0.1 + 0.8 * 0.107 = 0.1856
θ_4	-0.4 + 0.8 * (-0.008) = -0.4064
θ_3	0.1 + 0.8 * (-0.005) = 0.096

3. Base on the given training datasets, 1-NN is ok, but it would be overfitting. And the large extreme values are underfitting.