Theoretical Exercises

Q: Given this hypothesis $\boldsymbol{h_\theta(x) = \theta_0 + \theta_1 x}$ and the initial state ($\boldsymbol{\theta_0 = 0,\ \theta_1 = 0}$), show how to obtain the values of $\boldsymbol{\theta_0}$ and $\boldsymbol{\theta_1}$ after a single step of SGD with learning rate $\boldsymbol{\alpha = 0.01}$ and a training sample $(\boldsymbol{x^*, y^*})$ with $\boldsymbol{x^* = 5}$ and $\boldsymbol{y^* = 2}$.

In **Stochastic Gradient Descent** (**SGD**) to update the parameters is used only **one** training sample at each step:

$$\theta_j := \theta_j - \alpha \frac{\partial J(\theta)}{\theta_j}$$

where:

$$\frac{\partial J(\theta)}{\theta_j} = \left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)x_j^{(i)}$$

Note: $x_0^{(i)} = 1.$

The hypothesis is:

$$h_\theta(x^*) = \theta_0 x_0^* + \theta_0 x_1^* = 0 * 0 + 0 * 5 = 0$$

We first compute the partial derivates w.r.t to each parameter:

$$\frac{\partial J(\theta)}{\theta_0} = (h_\theta(x^*) - y^*)x_0^* = (0-2) * 1 = -2$$

$$\frac{\partial J(\theta)}{\theta_1} = (h_\theta(x^*) - y^*)x_1^* = (0-2) * 5 = -10$$

Then we proceed with the updates:

$$\theta_0 - \alpha \frac{\partial J(\theta)}{\theta_0} = 0 - 0.01 * (-2) = 0.02$$

$$\rightarrow \quad \theta_0 := 0.02$$

$$\theta_1 - \alpha \frac{\partial J(\theta)}{\theta_1} = 0 - 0.01 * (-10) = 0.1$$

$$\theta_1 := 0.1$$

Therefore, after one step of SGD we get $\theta_0 = 0.02$ and $\theta_1 = 0.1$.

Q: Consider a linear regression model with the hypothesis:

$$\boldsymbol{h_\theta\left(x^{(i)}\right) = \theta_0 + \theta_1 x_1^{(i)} + \theta_2 x_2^{(i)} + \theta_3 x_3^{(i)}}$$

You are given the following training dataset:

$$\mathcal{D} = \begin{Bmatrix} \left(\boldsymbol{x}^{(1)} = [\mathbf{1}, \mathbf{2}, \mathbf{3}], \boldsymbol{y}^{(1)} = \mathbf{6}\right) \\ \left(\boldsymbol{x}^{(2)} = [\mathbf{2}, \mathbf{1}, \mathbf{0}], \boldsymbol{y}^{(2)} = \mathbf{3}\right) \\ \left(\boldsymbol{x}^{(3)} = [\mathbf{0}, \mathbf{3}, \mathbf{1}], \boldsymbol{y}^{(3)} = \mathbf{5}\right) \\ \left(\boldsymbol{x}^{(4)} = [\mathbf{4}, \mathbf{2}, \mathbf{1}], \boldsymbol{y}^{(4)} = \mathbf{10}\right) \\ \left(\boldsymbol{x}^{(5)} = [\mathbf{3}, \mathbf{3}, \mathbf{2}], \boldsymbol{y}^{(5)} = \mathbf{11}\right) \end{Bmatrix}$$

You need to:

1. **Update $\boldsymbol{\theta}$ using Batch Gradient Descent.**
2. **Update $\boldsymbol{\theta}$ using Stochastic Gradient Descent.**
3. **Update $\boldsymbol{\theta}$ using Mini-Batch Gradient Descent with a batch size $\boldsymbol{b} = \mathbf{3}$.**

Consider a learning rate $\boldsymbol{\alpha} = \mathbf{0.1}$ and that the initial parameters are:

$$\boldsymbol{\theta} = \begin{pmatrix} \mathbf{1} \\ \mathbf{0.5} \\ \mathbf{-0.2} \\ \mathbf{0.8} \end{pmatrix}$$

The Gradient Descent (GD) update each parameter as follows:

$$\theta_j := \theta_j - \alpha \frac{\partial J}{\partial \theta_j}$$

where $\frac{\partial J}{\partial \theta_j}$ is the partial derivate of the cost function J w.r.t. to the parameter θ_j.

1. For **Batch Gradient Descent (BGD)** the $\frac{\partial J}{\partial \theta_j}$ is given by:

$$\boldsymbol{\frac{\partial J}{\partial \theta_j} = \frac{1}{m} \sum_{i=1}^{m} \left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right) x_j^{(i)}}$$

In BGD **all** m data points are considered to perform a GD update step.

First, we compute predictions $h_\theta\left(x^{(i)}\right)$ for all data points:

$$h_\theta\left(x^{(1)}\right) = 1 + 0.5 * 1 - 0.2 * 2 + 0.8 * 3 = 3.5$$

$$h_\theta\left(x^{(2)}\right) = 1 + 0.5 * 2 - 0.2 * 1 + 0.8 * 0 = 1.8$$

$$h_\theta\left(x^{(3)}\right) = 1 + 0.5 * 0 - 0.2 * 3 + 0.8 * 1 = 1.2$$

$$h_\theta\left(x^{(4)}\right) = 1 + 0.5 * 4 - 0.2 * 2 + 0.8 * 1 = 3.4$$

$$h_\theta\left(x^{(5)}\right) = 1 + 0.5 * 3 - 0.2 * 3 + 0.8 * 2 = 3.5$$

Given it we can compute all the partial derivates:

$$\frac{\partial J}{\partial \theta_0} = \frac{1}{m} \sum_{i=1}^{m} \left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right) x_0^{(i)} = \frac{1}{5}[(3.5 - 6) + (1.8 - 3) + (1.2 - 5) + (3.4 - 10) + (3.5 - 11)] = -4.32$$

$$\frac{\partial J}{\partial \theta_1} = \frac{1}{m}\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)x_1^{(i)} =$$

$$= \frac{1}{5}[(3.5-6)*1 + (1.8-3)*2 + (1.2-5)*0 + (3.4-10)*4 + (3.5-11)*3] = -10.76$$

$$\frac{\partial J}{\partial \theta_2} = \frac{1}{m}\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)x_2^{(i)} =$$

$$= \frac{1}{5}[(3.5-6)*2 + (1.8-3)*1 + (1.2-5)*3 + (3.4-10)*2 + (3.5-11)*3] = -10.66$$

$$\frac{\partial J}{\partial \theta_3} = \frac{1}{m}\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)x_3^{(i)} =$$

$$= \frac{1}{5}[(3.5-6)*3 + (1.8-3)*0 + (1.2-5)*1 + (3.4-10)*1 + (3.5-11)*2] = -6.58$$

Once computed them, we can then update all the parameters simultaneously such as:

$$\theta_0 := \theta_0 - \alpha\frac{\partial J}{\partial \theta_0} = 1 - 0.1(-4.32) = 1.432$$

$$\theta_1 := \theta_1 - \alpha\frac{\partial J}{\partial \theta_1} = 0.5 - 0.1(-10.76) = 1.567$$

$$\theta_2 := \theta_2 - \alpha\frac{\partial J}{\partial \theta_2} = -0.2 - 0.1(-10.66) = 0.866$$

$$\theta_3 := \theta_3 - \alpha\frac{\partial J}{\partial \theta_3} = 0.8 - 0.1(-6.58) = 1.458$$

Therefore, with BGD the final parameters values are given by:

$$\boldsymbol{\theta_0 = 1.432} \qquad \boldsymbol{\theta_1 = 1.567} \qquad \boldsymbol{\theta_2 = 0.866} \qquad \boldsymbol{\theta_3 = 1.458}$$

2. For **Stochastic Gradient Descent (SGD)** the $\frac{\partial J}{\partial \theta_j}$ is given by:

$$\boldsymbol{\frac{\partial J}{\partial \theta_j} = \left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)x_j^{(i)}}$$

In SGD a **single** data point at time is considered to perform a GD update step. The update is performed for each datapoint in the training set.

For the **first datapoint** $\left(\boldsymbol{x^{(1)}, y^{(1)}}\right)$ we have:

$$h_\theta\left(x^{(1)}\right) = 1 + 0.5*1 - 0.2*2 + 0.8*3 = 3.5$$

Given it we can compute all the partial derivates:

$$\frac{\partial J}{\partial \theta_0} = \left(h_\theta\left(x^{(1)}\right) - y^{(1)}\right)x_0^{(1)} = 3.5 - 6 = -2.5$$

$$\frac{\partial J}{\partial \theta_1} = \left(h_\theta\left(x^{(1)}\right) - y^{(1)}\right)x_1^{(1)} = (3.5-6)*1 = -2.5$$

$$\frac{\partial J}{\partial \theta_2} = \left(h_\theta\left(x^{(1)}\right) - y^{(1)}\right)x_2^{(1)} = (3.5 - 6) * 2 = -5$$

$$\frac{\partial J}{\partial \theta_3} = \left(h_\theta\left(x^{(1)}\right) - y^{(i)}\right)x_3^{(1)} = (3.5 - 6) * 3 = -7.5$$

Once computed them, we can then update all the parameters simultaneously such as:

$$\theta_0 := \theta_0 - \alpha \frac{\partial J}{\partial \theta_0} = 1 - 0.1(-2.5) = 1.25$$

$$\theta_1 := \theta_1 - \alpha \frac{\partial J}{\partial \theta_1} = 0.5 - 0.1(-2.5) = 0.75$$

$$\theta_2 := \theta_2 - \alpha \frac{\partial J}{\partial \theta_2} = -0.2 - 0.1(-5) = 0.3$$

$$\theta_3 := \theta_3 - \alpha \frac{\partial J}{\partial \theta_3} = 0.8 - 0.1(-7.5) = 1.55$$

For the **second datapoint** $\left(\boldsymbol{x}^{(2)}, \boldsymbol{y}^{(2)}\right)$ we have:

$$h_\theta\left(x^{(2)}\right) = 1.25 * 1 + 0.75 * 2 + 0.3 * 1 + 1.55 * 0 = 3.05$$

Given it we can compute all the partial derivates:

$$\frac{\partial J}{\partial \theta_0} = \left(h_\theta\left(x^{(2)}\right) - y^{(2)}\right)x_0^{(2)} = (3.05 - 3) = 0.05$$

$$\frac{\partial J}{\partial \theta_1} = \left(h_\theta\left(x^{(2)}\right) - y^{(2)}\right)x_1^{(2)} = (3.05 - 3) * 2 = 0.1$$

$$\frac{\partial J}{\partial \theta_2} = \left(h_\theta\left(x^{(2)}\right) - y^{(2)}\right)x_2^{(2)} = (3.05 - 3) * 1 = 0.05$$

$$\frac{\partial J}{\partial \theta_3} = \left(h_\theta\left(x^{(2)}\right) - y^{(2)}\right)x_3^{(2)} = (3.05 - 3) * 0 = 0$$

Once computed them, we can then update all the parameters simultaneously such as:

$$\theta_0 := \theta_0 - \alpha \frac{\partial J}{\partial \theta_0} = 1.25 - 0.1 * 0.05 = 1.245$$

$$\theta_1 := \theta_1 - \alpha \frac{\partial J}{\partial \theta_1} = 0.75 - 0.1 * 0.1 = 0.74$$

$$\theta_2 := \theta_2 - \alpha \frac{\partial J}{\partial \theta_2} = 0.3 - 0.1 * 0.05 = 0.295$$

$$\theta_3 := \theta_3 - \alpha \frac{\partial J}{\partial \theta_3} = 1.55 - 0.1 * 0 = 1.55$$

For the **third datapoint** $\left(\boldsymbol{x}^{(3)}, \boldsymbol{y}^{(3)}\right)$ we have:

$$h_\theta\left(x^{(3)}\right) = 1.245 * 1 + 0.74 * 0 + 0.295 * 3 + 1.55 * 1 = 3.68$$

Given it we can compute all the partial derivates:

$$\frac{\partial J}{\partial \theta_0} = \left(h_\theta\left(x^{(3)}\right) - y^{(3)}\right)x_0^{(3)} = (3.68 - 5) = -1.32$$

$$\frac{\partial J}{\partial \theta_1} = \left(h_\theta\left(x^{(3)}\right) - y^{(3)}\right)x_1^{(3)} = (3.68 - 5) * 0 = 0$$

$$\frac{\partial J}{\partial \theta_2} = \left(h_\theta\left(x^{(3)}\right) - y^{(3)}\right)x_2^{(3)} = (3.68 - 5) * 3 = -3.96$$

$$\frac{\partial J}{\partial \theta_3} = \left(h_\theta\left(x^{(3)}\right) - y^{(3)}\right)x_3^{(3)} = (3.68 - 5) * 1 = -1.32$$

Once computed them, we can then update all the parameters simultaneously such as:

$$\theta_0 := \theta_0 - \alpha \frac{\partial J}{\partial \theta_0} = 1.245 - 0.1 * (-1.32) = 1.377$$

$$\theta_1 := \theta_1 - \alpha \frac{\partial J}{\partial \theta_1} = 0.74 - 0.1 * 0 = 0.74$$

$$\theta_2 := \theta_2 - \alpha \frac{\partial J}{\partial \theta_2} = 0.295 - 0.1 * (-3.95) = 0.69$$

$$\theta_3 := \theta_3 - \alpha \frac{\partial J}{\partial \theta_3} = 1.55 - 0.1 * (-1.32) = 1.682$$

For the **fourth datapoint** $\left(\boldsymbol{x^{(4)}}, \boldsymbol{y^{(4)}}\right)$ we have:

$$h_\theta\left(x^{(4)}\right) = 1.377 * 1 + 0.74 * 4 + 0.69 * 2 + 1.682 * 1 = 7.399$$

Given it we can compute all the partial derivates:

$$\frac{\partial J}{\partial \theta_0} = \left(h_\theta\left(x^{(4)}\right) - y^{(4)}\right)x_0^{(4)} = (7.399 - 10) = -2.601$$

$$\frac{\partial J}{\partial \theta_1} = \left(h_\theta\left(x^{(4)}\right) - y^{(4)}\right)x_1^{(4)} = (7.399 - 10) * 4 = -10.404$$

$$\frac{\partial J}{\partial \theta_2} = \left(h_\theta\left(x^{(4)}\right) - y^{(4)}\right)x_2^{(4)} = (7.399 - 10) * 2 = -5.202$$

$$\frac{\partial J}{\partial \theta_3} = \left(h_\theta\left(x^{(4)}\right) - y^{(4)}\right)x_3^{(4)} = (7.399 - 10) * 1 = -2.601$$

Once computed them, we can then update all the parameters simultaneously such as:

$$\theta_0 := \theta_0 - \alpha \frac{\partial J}{\partial \theta_0} = 1.377 - 0.1 * (-2.601) = 1.637$$

$$\theta_1 := \theta_1 - \alpha \frac{\partial J}{\partial \theta_1} = 0.74 - 0.1 * (-10.404) = 1.78$$

$$\theta_2 := \theta_2 - \alpha \frac{\partial J}{\partial \theta_2} = 0.69 - 0.1 * (-5.202) = 1.21$$

$$\theta_3 := \theta_3 - \alpha \frac{\partial J}{\partial \theta_3} = 1.682 - 0.1 * (-2.601) = 1.94$$

For the **fifth datapoint** $(\boldsymbol{x}^{(5)}, \boldsymbol{y}^{(5)})$ we have:

$$h_\theta(x^{(5)}) = 1.637 * 1 + 1.78 * 3 + 1.21 * 3 + 1.94 * 2 = 14.487$$

Given it we can compute all the partial derivates:

$$\frac{\partial J}{\partial \theta_0} = (h_\theta(x^{(5)}) - y^{(5)})x_0^{(5)} = (14.487 - 11) = 3.487$$

$$\frac{\partial J}{\partial \theta_1} = (h_\theta(x^{(5)}) - y^{(5)})x_1^{(5)} = (14.487 - 11) * 3 = 10.461$$

$$\frac{\partial J}{\partial \theta_2} = (h_\theta(x^{(5)}) - y^{(5)})x_2^{(5)} = (14.487 - 11) * 3 = 10.461$$

$$\frac{\partial J}{\partial \theta_3} = (h_\theta(x^{(5)}) - y^{(5)})x_3^{(5)} = (14.487 - 11) * 2 = 6.974$$

Once computed them, we can then update all the parameters simultaneously such as:

$$\theta_0 := \theta_0 - \alpha \frac{\partial J}{\partial \theta_0} = 1.637 - 0.1 * 3.487 = 1.288$$

$$\theta_1 := \theta_1 - \alpha \frac{\partial J}{\partial \theta_1} = 1.78 - 0.1 * 10.461 = 0.734$$

$$\theta_2 := \theta_2 - \alpha \frac{\partial J}{\partial \theta_2} = 1.21 - 0.1 * 10.461 = 0.164$$

$$\theta_3 := \theta_3 - \alpha \frac{\partial J}{\partial \theta_3} = 1.94 - 0.1 * 6.974 = 1.243$$

Therefore, with SGD the final parameters values are given by:

$$\boldsymbol{\theta_0 = 1.288} \qquad \boldsymbol{\theta_1 = 0.734} \qquad \boldsymbol{\theta_2 = 0.164} \qquad \boldsymbol{\theta_3 = 1.243}$$

3. For **Mini Batch Gradient Descent (MBGD)** the $\frac{\partial J}{\partial \theta_J}$ is given by:

$$\boldsymbol{\frac{\partial J}{\partial \theta_j} = \frac{1}{b}\sum_{i=1}^{b}(h_\theta(x^{(i)}) - y^{(i)})x_j^{(i)}}$$

In MBGD a **batch** of **b** data point at time is considered to perform a GD update step. The training set is divided into batches, each of a given size, and the update is performed for each batch in the training set.

Given a batch size $b = 3$, we will have the following batches:

$$b_1 = [(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), (x^{(3)}, y^{(3)})] \qquad b_2 = [(x^{(4)}, y^{(4)}), (x^{(5)}, y^{(5)})]$$

Note: In case in the last batch there are less samples than batch size (e.g. $b = 3$) you must simply include the remaining ones.

Considering the **first batch of datapoints** b_1 we have:

$$h_\theta(x^{(1)}) = 1 * 1 + 0.5 * 1 - 0.2 * 2 + 0.8 * 3 = 3.5$$

$$h_\theta(x^{(2)}) = 1 * 1 + 0.5 * 2 - 0.2 * 1 + 0.8 * 0 = 1.8$$

$$h_\theta(x^{(3)}) = 1 * 1 + 0.5 * 0 - 0.2 * 3 + 0.8 * 1 = 1.2$$

Given them we can compute all the partial derivates:

$$\frac{\partial J}{\partial \theta_0} = \frac{1}{b}\sum_{i=1}^{b}\left(h_\theta(x^{(i)}) - y^{(i)}\right)x_0^{(i)} = \frac{1}{3}[(3.5-6) + (1.8-3) + (1.2-5)] = -2.5$$

$$\frac{\partial J}{\partial \theta_1} = \frac{1}{b}\sum_{i=1}^{b}\left(h_\theta(x^{(i)}) - y^{(i)}\right)x_1^{(i)} = \frac{1}{3}[(3.5-6) * 1 + (1.8-3) * 2 + (1.2-5) * 0] = -1.633$$

$$\frac{\partial J}{\partial \theta_2} = \frac{1}{b}\sum_{i=1}^{b}\left(h_\theta(x^{(i)}) - y^{(i)}\right)x_2^{(i)} = \frac{1}{3}[(3.5-6) * 2 + (1.8-3) * 1 + (1.2-5) * 3] = -5.867$$

$$\frac{\partial J}{\partial \theta_3} = \frac{1}{b}\sum_{i=1}^{b}\left(h_\theta(x^{(i)}) - y^{(i)}\right)x_3^{(i)} = \frac{1}{3}[(3.5-6) * 3 + (1.8-3) * 0 + (1.2-5) * 1] = -3.767$$

Once computed them, we can then update all the parameters simultaneously such as:

$$\theta_0 := \theta_0 - \alpha\frac{\partial J}{\partial \theta_0} = 1 - 0.1 * (-2.5) = 1.25$$

$$\theta_1 := \theta_1 - \alpha\frac{\partial J}{\partial \theta_1} = 0.5 - 0.1 * (-1.633) = 0.663$$

$$\theta_2 := \theta_2 - \alpha\frac{\partial J}{\partial \theta_2} = -0.2 - 0.1 * (-5.867) = 0.387$$

$$\theta_3 := \theta_3 - \alpha\frac{\partial J}{\partial \theta_3} = 0.8 - 0.1(-3.767) = 1.177$$

Considering the **second batch of datapoints** b_2 we have:

$$h_\theta(x^{(4)}) = 1.25 * 1 + 0.663 * 4 + 0.387 * 2 + 1.177 * 1 = 5.853$$

$$h_\theta(x^{(5)}) = 1.25 * 1 + 0.663 * 3 + 0.387 * 3 + 1.177 * 2 = 6.664$$

Given them we can compute all the partial derivates:

$$\frac{\partial J}{\partial \theta_0} = \frac{1}{b}\sum_{i=1}^{b}\left(h_\theta(x^{(i)}) - y^{(i)}\right)x_0^{(i)} = \frac{1}{2}[(5.853-10) + (6.664-11)] = -4.242$$

$$\frac{\partial J}{\partial \theta_1} = \frac{1}{b}\sum_{i=1}^{b}\left(h_\theta(x^{(i)}) - y^{(i)}\right)x_1^{(i)} = \frac{1}{2}[(5.853-10) * 4 + (6.664-11) * 3] = -14.798$$

$$\frac{\partial J}{\partial \theta_2} = \frac{1}{b}\sum_{i=1}^{b}\left(h_\theta(x^{(i)}) - y^{(i)}\right)x_2^{(i)} = \frac{1}{2}[(5.853-10) * 2 + (6.664-11) * 3] = -10.651$$

$$\frac{\partial J}{\partial \theta_3} = \frac{1}{b}\sum_{i=1}^{b}\left(h_\theta(x^{(i)}) - y^{(i)}\right)x_3^{(i)} = \frac{1}{2}[(5.853-10) * 1 + (6.664-11) * 2] = -6.41$$

Once computed them, we can then update all the parameters simultaneously such as:

$$\theta_0 := \theta_0 - \alpha \frac{\partial J}{\partial \theta_0} = 1.25 - 0.1 * (-4.242) = 1.674$$

$$\theta_1 := \theta_1 - \alpha \frac{\partial J}{\partial \theta_1} = 0.663 - 0.1 * (-14.798) = 2.143$$

$$\theta_2 := \theta_2 - \alpha \frac{\partial J}{\partial \theta_2} = 0.387 - 0.1 * (-10.651) = 1.452$$

$$\theta_3 := \theta_3 - \alpha \frac{\partial J}{\partial \theta_3} = 1.177 - 0.1 * (-6.41) = 1.818$$

Therefore, with MBGD the final parameters values are given by:

$$\boldsymbol{\theta_0 = 1.674} \qquad \boldsymbol{\theta_1 = 2.143} \qquad \boldsymbol{\theta_2 = 1.452} \qquad \boldsymbol{\theta_3 = 1.818}$$

Q: Given this hypothesis $h_\theta(x^{(i)}) = \theta_0 + \theta_1 x_1^{(i)} + \theta_1 x_2^{(i)}$ and the initial state ($\theta_0 = 1, \theta_1 = 0.2, \theta_2 = 0.4$), show how to obtain the values of θ_0, θ_1 and θ_2 after a single step of SGD with learning rate $\alpha = 0.1$ in case <u>l2 regularization is applied</u> with $\lambda = 0.8$ and given the following training samples:

$$\left(\boldsymbol{x}^{(1)} = [4, 2]^T, \boldsymbol{y}^{(1)} = 6\right)$$

$$\left(\boldsymbol{x}^{(2)} = [1, 9]^T, \boldsymbol{y}^{(2)} = 3\right)$$

The **Stochastic Gradient Descent** (**SGD**) takes in consideration only **one** training sample at time to update the parameters. The regularization is applied **only** to weights (θ_1 and θ_2) and **NOT** to the bias parameter $\boldsymbol{\theta_0}$.

The update rule for SGD given l2 regularization applied is given by:

$$\theta_j := \theta_j - \alpha \frac{\partial J(\theta)}{\theta_j}$$

where:

$$\begin{cases} \dfrac{\partial J(\theta)}{\theta_0} = \left(h_\theta(x^{(i)}) - y^{(i)}\right) x_0^{(i)} \\ \dfrac{\partial J(\theta)}{\theta_j} = \left(h_\theta(x^{(i)}) - y^{(i)}\right) x_j^{(i)} + \lambda\theta_j \qquad for\ j = \mathbf{1}, \dots, n \end{cases}$$

Said that considering the first training sample $\left(\boldsymbol{x}^{(1)}, \boldsymbol{y}^{(1)}\right)$ we have:

$$\frac{\partial J(\theta)}{\theta_0} = \left(h_\theta(x^{(1)}) - y^{(1)}\right) x_0^{(1)} = \left((1 + 0.2 * 4 + 0.4 * 2) - 6\right) * 1 = -3.4$$

$$\frac{\partial J(\theta)}{\theta_1} = \left(h_\theta(x^{(1)}) - y^{(1)}\right) x_1^{(1)} + \lambda\theta_1 = \left((1 + 0.2 * 4 + 0.4 * 2) - 6\right) * 4 + 0.8 * 0.2 = -13.44$$

$$\frac{\partial J(\theta)}{\theta_2} = \left(h_\theta\left(x^{(1)}\right) - y^{(1)}\right)x_2^{(1)} + \lambda\theta_2 = \left((1 + 0.2 * 4 + 0.4 * 2) - 6\right) * 2 + 0.8 * 0.4 = -6.48$$

And then:

$$\theta_0 := \theta_0 - \alpha\frac{\partial J}{\partial\theta_0} = 1 - 0.1(-3.4) = 1.34$$

$$\theta_1 := \theta_1 - \alpha\frac{\partial J}{\partial\theta_1} = 0.2 - 0.1(-13.44) = 1.544$$

$$\theta_2 := \theta_2 - \alpha\frac{\partial J}{\partial\theta_2} = 0.4 - 0.1(-6.48) = 1.048$$

Now considering the second training sample $\left(\boldsymbol{x^{(2)}}, \boldsymbol{y^{(2)}}\right)$ we have:

$$\frac{\partial J(\theta)}{\theta_0} = \left(h_\theta\left(x^{(2)}\right) - y^{(2)}\right)x_0^{(2)} = \left((1.34 * 1 + 1.544 * 1 + 1.048 * 9) - 3\right) = 9.316$$

$$\frac{\partial J(\theta)}{\theta_1} = \left(h_\theta\left(x^{(2)}\right) - y^{(2)}\right)x_1^{(2)} + \lambda\theta_1 = \left((1.34 * 1 + 1.544 * 1 + 1.048 * 9) - 3\right) * 1 + 0.8 * 1.544 = 10.551$$

$$\frac{\partial J(\theta)}{\theta_2} = \left(h_\theta\left(x^{(2)}\right) - y^{(2)}\right)x_2^{(2)} + \lambda\theta_2 = \left((1.34 * 1 + 1.544 * 1 + 1.048 * 9) - 3\right) * 9 + 0.8 * 1.048 = 84.682$$

And then:

$$\theta_0 := \theta_0 - \alpha\frac{\partial J}{\partial\theta_0} = 1.34 - 0.1 * 9.316 = 0.408$$

$$\theta_1 := \theta_1 - \alpha\frac{\partial J}{\partial\theta_1} = 1.544 - 0.1 * 10.551 = 0.489$$

$$\theta_2 := \theta_2 - \alpha\frac{\partial J}{\partial\theta_2} = 1.048 - 0.1 * 84.682 = -7.42$$

Therefore, the final parameters values are given by:

$$\boldsymbol{\theta_0 = 0.408} \qquad \boldsymbol{\theta_1 = 0.489} \qquad \boldsymbol{\theta_2 = -7.42}$$

Q: Consider the following dataset and verify if an outlier is present using IQR.

$$\{\mathbf{53, 50, 52, 56, 57, 95, 45, 58, 59, 60, 51, 61}\}$$

First, we sort it in ascending order:

$$\{45,50,51,52,53,56,57,58,59,60,61,95\}$$

Then we calculate the quartiles:

$$Q_2 = \frac{56 + 57}{2} = 56.5$$

$$Q_1 = \frac{51 + 52}{2} = 51.5$$

$$Q_3 = \frac{60 + 61}{2} = 61.5$$

Next, we compute the IQR:

$$IQR = Q_3 - Q_1 = 61.5 - 51,5 = 10$$

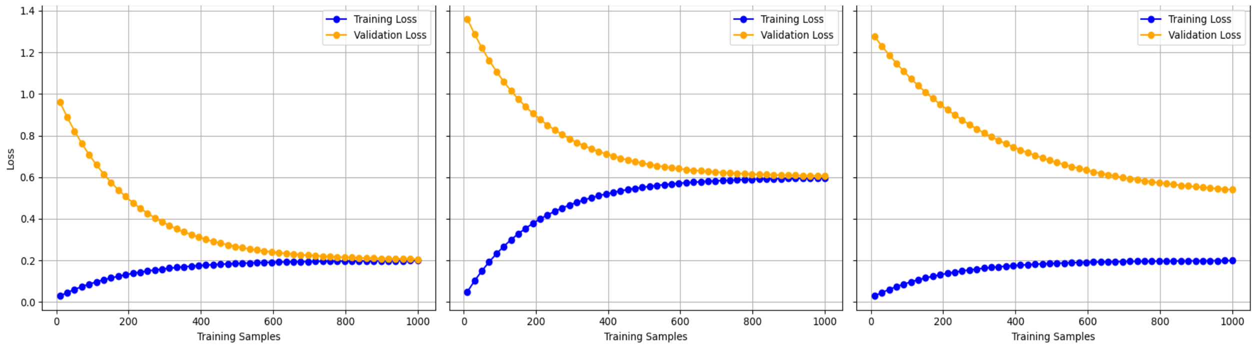
At the end we compute the lower and upper bounds (outlier bounds):

$$Q_1 - 1.5 * IQR = 51.5 - 1.5 * 10 = 36.5$$

$$Q_3 + 1.5 * IQR = 61.5 + 1.5 * 10 = 76.5$$

Therefore, in this case we can say that the data value 95 (> 76.5) is an **outlier**.

Q: Below are three plots depicting learning curves from a machine learning training process. First, explain what learning curves are, and then analyze the three plots.



Learning curves are a graphical representation that shows how a machine learning model's performance on a training set and a validation set changes with the amount of training data. Typically, the training set size is plotted on the x-axis, and the value of the loss function on the y-axis.

Learning curves can be used to diagnose problems such determining if the model suffers from a variance error (i.e. it overfits) or a bias error (i.e. it underfits).

Specifically in the three plots we can see:

- In the first plot, both the training loss and validation loss converge to similar low values as the number of training examples increases. This scenario demonstrates a well-trained model with a good **bias-variance tradeoff** (**good fit**). The model is effectively learning the patterns in the data and it's also able to generalize well to unseen data. No significant improvements are needed here.

- In the second plot, both the training loss and validation loss remain high as the number of training examples increases. This scenario reflects **high bias** (**underfitting**), where the model is too simple to capture the underlying patterns in the data. As consequence, the model performs poorly on both the training and validation data.

  To address underfitting:

1. Increase the model complexity.

2. Increase the set of features.

3. If using L1 or L2 regularization, decrease the regularization parameter λ.

- In the third plot, the training loss converges to a very low value, while the validation loss converges at a significantly higher value. A large gap exists between the two losses. This scenario highlights **high variance** (**overfitting**), where the model memorizes the training data (low training loss), but struggles to generalize to unseen validation data (high validation loss).
  To address overfitting:

  1. Use regularization techniques like L1 or L2, or if you are already using one of these two you can consider increasing the regularization parameter λ.

  2. Reduce the model complexity.

  3. Try smaller sets of features.

  4. Increase the amount of training examples.

Q: Consider a binary classifier that has output the following probability scores for a set of 10 test samples, along with their actual ground truth labels:

Sample	**Actual Class (0 = Negative, 1 = Positive)**	**Predicted Probability (Positive Class)**
1	0	0.1
2	0	0.4
3	1	0.35
4	1	0.8
5	1	0.7
6	0	0.6
7	1	0.9
8	0	0.3
9	1	0.75
10	0	0.2

Compute the FPR and TPR for the following thresholds 0.1, 0.3, 0.4, 0.5, 0.7 in order to plot the ROC Curve. After that, compute the AUC to evaluate the binary classifier overall performance.

Let's first compute the FPR and TPR for each threshold.

Threshold 0.1: For threshold $= 0.1$, we classify samples with predicted probability ≥ 0.1 as positive, and the rest as negative.

Given that we can build the confusion matrix:

$TP = 5$	$FN = 0$
$FP = 5$	$TN = 0$

Confusion Matrix:

- **TP = 5** (samples 3, 4, 5, 7, 9)
- **FP = 5** (samples 1, 2, 6, 8, 10)
- **TN = 0**
- **FN = 0**

And we can calculate FPR and TPR:

$$FPR = \frac{FP}{FP + TN} = \frac{5}{5 + 0} = 1.0$$

$$TPR = \frac{TP}{TP + FN} = \frac{5}{5 + 0} = 1.0$$

Threshold 0.3: For threshold $= 0.3$, we classify samples with predicted probability ≥ 0.3 as positive, and the rest as negative.

Given that we can build the confusion matrix:

$TP = 5$	$FN = 0$
$FP = 3$	$TN = 2$

Confusion Matrix:

- **TP = 5** (samples 3, 4, 5, 7, 9)
- **FP = 3** (samples 2, 6, 8)
- **TN = 2** (samples 1,10)
- **FN = 0**

And we can calculate FPR and TPR:

$$FPR = \frac{FP}{FP + TN} = \frac{3}{3 + 2} = 0.6$$

$$TPR = \frac{TP}{TP + FN} = \frac{5}{5 + 0} = 1.0$$

Threshold 0.4: For threshold $= 0.4$, we classify samples with predicted probability ≥ 0.4 as positive, and the rest as negative.

Given that we can build the confusion matrix:

$TP = 4$	$FN = 3$
$FP = 2$	$TN = 1$

Confusion Matrix:

- **TP = 4** (samples 4, 5, 7, 9)
- **FP = 2** (samples 2, 6)
- **TN = 3** (samples 1, 8, 10)
- **FN = 1** (sample 3)

And we can calculate FPR and TPR:

$$FPR = \frac{FP}{FP + TN} = \frac{2}{2 + 3} = 0.4$$

$$TPR = \frac{TP}{TP + FN} = \frac{4}{4 + 1} = 0.8$$

Threshold 0.5: For threshold $= 0.5$, we classify samples with predicted probability ≥ 0.5 as positive, and the rest as negative.

Given that we can build the confusion matrix:

$TP = 4$	$FN = 1$
$FP = 1$	$TN = 4$

Confusion Matrix:

- **TP = 4** (samples 4, 5, 7, 9)
- **FP = 1** (sample 6)
- **TN = 4** (samples 1, 2, 8, 10)
- **FN = 1** (sample 3)

And we can calculate FPR and TPR:

$$FPR = \frac{FP}{FP + TN} = \frac{1}{1 + 4} = 0.2$$

$$TPR = \frac{TP}{TP + FN} = \frac{4}{4 + 1} = 0.8$$

Threshold 0.7: For threshold $= 0.7$, we classify samples with predicted probability ≥ 0.7 as positive, and the rest as negative.

Given that we can build the confusion matrix:

$TP = 4$	$FN = 1$
$FP = 0$	$TN = 5$

Confusion Matrix:

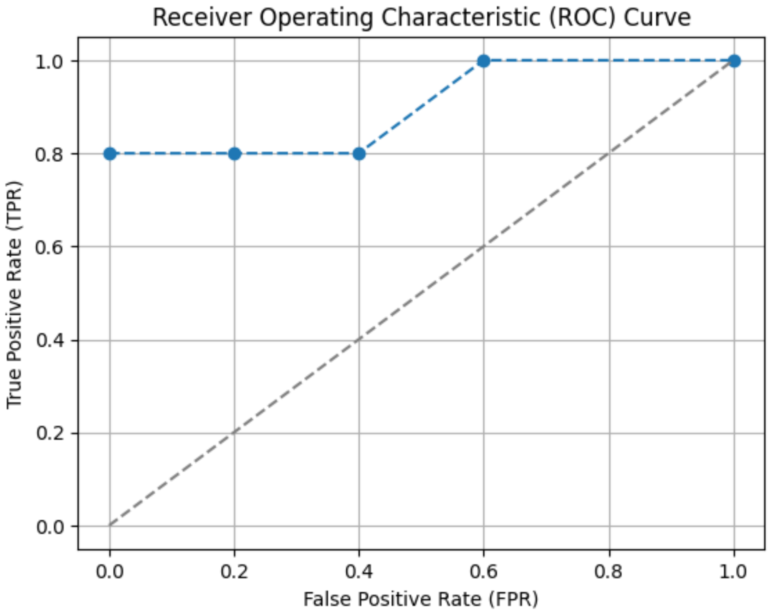
- **TP = 4** (samples 4,5,7,9)
- **FP = 0**
- **TN = 5** (samples 1,2,6,8,10)
- **FN = 1** (sample 3)

And we can calculate FPR and TPR:

$$FPR = \frac{FP}{FP + TN} = \frac{0}{0 + 5} = 0.0$$

$$TPR = \frac{TP}{TP + FN} = \frac{4}{4 + 1} = 0.8$$

Once calculated the TPR and FPR for all thresholds, we can plot the ROC curve with FPR on the x-axis and TPR on the y-axis.



Compute the AUC

After plotting the ROC curve, we are now ready to calculate the **AUC (Area Under the Curve)**. If you have the TPR and FPR values at each threshold (or at least the right number of thresholds such as our case), you can approximate the AUC using the trapezoidal rule.

The formula for AUC using the trapezoidal rule is:

$$AUC = \sum_{i=1}^{n-1} \left[\frac{(FPR_i - FPR_{i-1})(TPR_i + TPR_{i-1})}{2} \right]$$

This formula computes the area of the trapezoids formed between consecutive points on the ROC curve.

Note: The **FPR values** must be sorted in **increasing order** before applying this formula.

Sorted FPR, TPR values (sorted by FPR):

Threshold	FPR	TPR
0.7	0.0	0.8
0.5	0.2	0.8
0.4	0.4	0.8
0.3	0.6	1.0
0.1	1.0	1.0

We calculate the area of each trapezoid formed between consecutive points:

1. **Between threshold 0.1 and 0.3**:

$$Area_{1-2} = \frac{(FPR_2 - FPR_1)(TPR_2 + TPR_1)}{2} = \frac{(0.2 - 0.0)(0.8 + 0.8)}{2} = 0.16$$

1. **Between threshold 0.3 and 0.4**:

$$Area_{2-3} = \frac{(FPR_3 - FPR_2)(TPR_3 + TPR_2)}{2} = \frac{(0.4 - 0.2)(0.8 + 0.8)}{2} = 0.16$$

1. **Between threshold 0.4 and 0.5**:

$$Area_{3-4} = \frac{(FPR_4 - FPR_3)(TPR_4 + TPR_3)}{2} = \frac{(0.6 - 0.4)(1.0 + 0.8)}{2} = 0.18$$

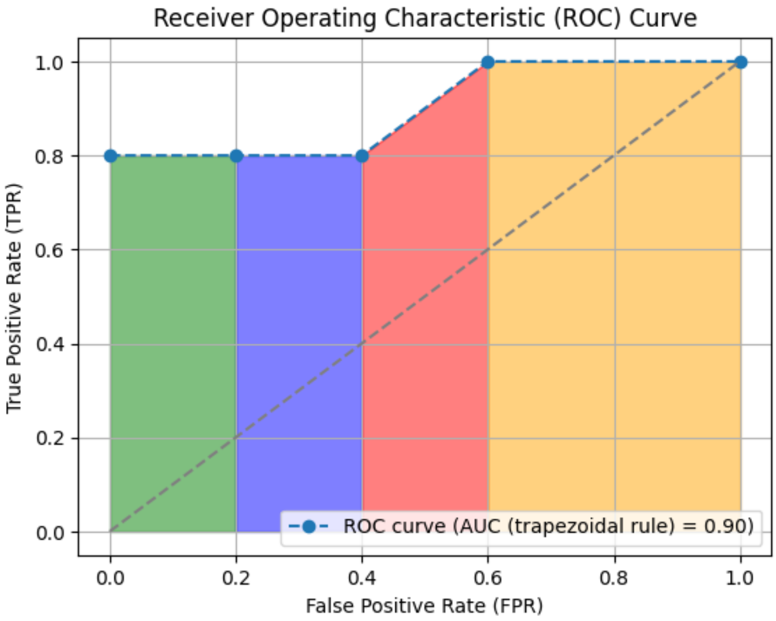
1. **Between threshold 0.5 and 0.7**:

$$Area_{4-5} = \frac{(FPR_5 - FPR_4)(TPR_5 + TPR_4)}{2} = \frac{(1.0 - 0.6)(1.0 + 1.0)}{2} = 0.4$$

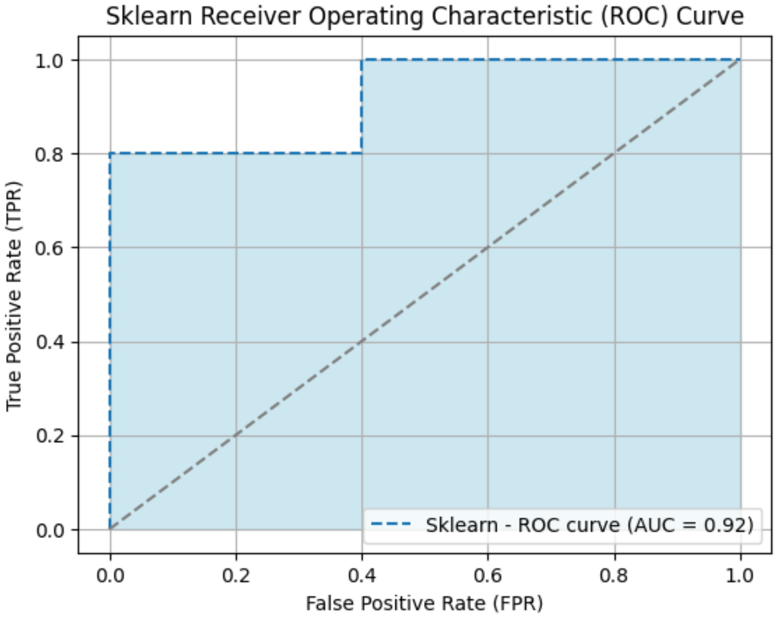
The total AUC is obtained by summing the areas of all trapezoids:

$$\boldsymbol{AUC} = 0.16 + 0.16 + 0.18 + 0.4 = \mathbf{0.90}$$

In Figure is shown the AUC obtained using the trapezoidal rule (where the areas of the trapezoids are highlighted in different colors):



We can compare it with the ROC curve and the AUC score that we would obtain using Sklearn:



Using Sklearn, the AUC is calculated with finer granularity, considering all unique thresholds.

We can see that we obtained a pretty good approximation since with trapezoidal rule we got $AUC = 0.90$, while the actual AUC was $AUC = 0.92$ (sklearn).

Note: The difference between the AUC with trapezoidal rule and the Sklearn AUC arises because Sklearn accounts for all available thresholds, including intermediate points that refine the curve, while here we accounted just some thresholds. Despite this, the threshold we considered let us compute an AUC that is a very close approximation of the real one.

Q: Consider a neural network adopted for a regression task consisting of a single hidden layer with 2 neurons and an output layer with 1 neuron. The network receives as input a single training example $(\boldsymbol{x} = [\mathbf{0.5}, \mathbf{0.2}, \mathbf{0}], \boldsymbol{y} = \mathbf{1.2})$

The parameters for the layers are:

$$\boldsymbol{\Theta}^{[1]} = \begin{bmatrix} \mathbf{0.6} & \mathbf{0.4} & \mathbf{0.3} & \mathbf{0.2} \\ \mathbf{0.1} & \mathbf{0.1} & \mathbf{0.7} & \mathbf{0.5} \end{bmatrix}$$

$$\boldsymbol{\Theta}^{[2]} = [\mathbf{0.5} \quad \mathbf{0.4} \quad \mathbf{0.2}]$$

Suppose just sigmoid activation functions are applied. Moreover, suppose to adopt a learning rate of $\boldsymbol{\alpha} = \mathbf{0.1}$ and a regularization factor of $\boldsymbol{\lambda} = \mathbf{0.7}$.

Perform a single step of forward and backward propagation and use the gradients so obtained to then update the parameters.

First for convenience let's separate weight and bias parameters:

$$W^{[1]} = \begin{bmatrix} 0.4 & 0.3 & 0.2 \\ 0.1 & 0.7 & 0.5 \end{bmatrix} \quad b^{[1]} = \begin{bmatrix} 0.6 \\ 0.1 \end{bmatrix}$$

$$W^{[2]} = [0.4 \quad 0.2] \qquad b^{[2]} = [0.5]$$

Forward propagation step (calculate the final output):

First, we compute the weighted sum for the hidden layer:

$$z^{[1]} = W^{[1]}x + b^{[1]} =$$

$$= \begin{bmatrix} 0.4 & 0.3 & 0.2 \\ 0.1 & 0.7 & 0.5 \end{bmatrix} \begin{bmatrix} 0.5 \\ 0.2 \\ 0 \end{bmatrix} + \begin{bmatrix} 0.6 \\ 0.1 \end{bmatrix} =$$

$$= \begin{bmatrix} (0.4)(0.5) + (0.3)(0.2) + (0.2)(0) \\ (0.1)(0.5) + (0.7)(0.2) + (0.5)(0) \end{bmatrix} + \begin{bmatrix} 0.6 \\ 0.1 \end{bmatrix} =$$

$$= \begin{bmatrix} 0.26 \\ 0.19 \end{bmatrix} + \begin{bmatrix} 0.6 \\ 0.1 \end{bmatrix} = \begin{bmatrix} 0.86 \\ 0.29 \end{bmatrix}$$

Then we compute the activations in the hidden layer:

$$a^{[1]} = \sigma(z^{[1]}) =$$

$$= \begin{bmatrix} \dfrac{1}{1 + e^{-0.86}} \\ \dfrac{1}{1 + e^{-0.29}} \end{bmatrix} \approx \begin{bmatrix} 0.702 \\ 0.572 \end{bmatrix}$$

Next, we compute the weighted sum for the output layer:

$$z^{[2]} = W^{[2]}a^{[1]} + b^{[2]} =$$

$$= [0.4 \quad 0.2] \begin{bmatrix} 0.702 \\ 0.572 \end{bmatrix} + [0.5] =$$

$$= [(0.4)(0.702) + (0.2)(0.572)] + [0.5] =$$

$$= [0.3952] + [0.5] = [0.895]$$

And at the end, recalling that for regression the sigmoid is not applied to the output layer, the final output is:

$$a^{[2]} = z^{[2]} = [0.895]$$

Backward propagation step (obtain the gradients):

First, we compute the error for the output layer which for a regression task considering using the MSE is simply given by the output minus the true target value:

$$\delta^{[2]} = a^{[2]} - y = [0.895] - [1.2] = [-0.305]$$

Given this error we can easily compute $\frac{\partial}{\partial w^{[2]}}$ and $\frac{\partial}{\partial b^{[2]}}$ as:

$$\frac{\partial}{\partial W^{[2]}} = \delta^{[2]}\left(a^{[1]}\right)^T + \lambda W^{[2]} = [-0.305]\,[0.702 \quad 0.572] + 0.7 * [0.4 \quad 0.2] =$$

$$= [-0.214 \quad -0.174] + [0.28 \quad 0.14] = [\mathbf{0.066} \quad \mathbf{-0.034}]$$

$$\frac{\partial}{\partial b^{[2]}} = \delta^{[2]} = [\mathbf{-0.305}\ ]$$

Note: We recall that we do not apply regularization to bias.

Next, we compute the error for layer one using also the error coming from the next layer as follows:

$$\delta^{[1]} = \left(\left(W^{[2]}\right)^T \delta^{[2]}\right) \odot \left(a^{[1]}\left(1 - a^{[1]}\right)\right) = \left(\begin{bmatrix}0.4\\0.2\end{bmatrix}[-0.3048]\right) \odot \left(\begin{bmatrix}0.702\\0.572\end{bmatrix} \odot \left(1 - \begin{bmatrix}0.702\\0.572\end{bmatrix}\right)\right)$$

$$= \begin{bmatrix}-0.122\\-0.061\end{bmatrix} \odot \left(\begin{bmatrix}0.209\\0.245\end{bmatrix}\right) = \begin{bmatrix}-0.025\\-0.015\end{bmatrix}$$

Given this error, we can easily compute $\frac{\partial}{\partial w^{[1]}}$ and $\frac{\partial}{\partial b^{[1]}}$ as:

$$\frac{\partial}{\partial W^{[1]}} = \delta^{[1]} x^T + \lambda W^{[1]} = \begin{bmatrix}-0.025\\-0.015\end{bmatrix}[0.5 \quad 0.2 \quad 0] + 0.7 * \begin{bmatrix}0.4 & 0.3 & 0.2\\0.1 & 0.7 & 0.5\end{bmatrix} =$$

$$= \begin{bmatrix}-0.013 & -0.005 & 0\\-0.075 & 0.003 & 0\end{bmatrix} + \begin{bmatrix}0.28 & 0.21 & 0.14\\0.07 & 0.49 & 0.35\end{bmatrix} = \begin{bmatrix}\mathbf{0.267} & \mathbf{0.205} & \mathbf{0.14}\\\mathbf{-0.005} & \mathbf{0.493} & \mathbf{0.35}\end{bmatrix}$$

$$\frac{\partial}{\partial b^{[1]}} = \delta^{[1]} = \begin{bmatrix}\mathbf{-0.025}\\\mathbf{-0.015}\end{bmatrix}$$

GD Parameters Update: Having computed the gradients of each layer, we can proceed in computing the new update parameters through Gradient Descent (GD):

$$W^{[2]} := W^{[2]} - \alpha\frac{\partial}{\partial W^{[2]}} = [0.4 \quad 0.2] - 0.1 * [0.066 \quad -0.034] = [0.393 \quad 0.203]$$

$$b^{[2]} := b^{[2]} - \alpha\frac{\partial}{\partial b^{[2]}} = [0.5] - 0.1 * [-0.305\ ] = [0.531]$$

$$W^{[1]} := W^{[1]} - \alpha\frac{\partial}{\partial W^{[1]}} = \begin{bmatrix}0.4 & 0.3 & 0.2\\0.1 & 0.7 & 0.5\end{bmatrix} - 0.1 * \begin{bmatrix}0.267 & 0.205 & 0.14\\-0.005 & 0.493 & 0.35\end{bmatrix} = \begin{bmatrix}0.373 & 0.280 & 0.186\\0.101 & 0.651 & 0.465\end{bmatrix}$$

$$b^{[1]} := b^{[1]} - \alpha\frac{\partial}{\partial b^{[1]}} = \begin{bmatrix}0.6\\0.1\end{bmatrix} - 0.1 * \begin{bmatrix}-0.025\\-0.015\end{bmatrix} = \begin{bmatrix}0.603\\0.102\end{bmatrix}$$

Therefore, at the end we got as new parameters:

$$\boldsymbol{W}^{[1]} = \begin{bmatrix}\mathbf{0.373} & \mathbf{0.280} & \mathbf{0.186}\\\mathbf{0.101} & \mathbf{0.651} & \mathbf{0.465}\end{bmatrix}, \quad \boldsymbol{b}^{[1]} = \begin{bmatrix}\mathbf{0.603}\\\mathbf{0.102}\end{bmatrix}$$

$$\boldsymbol{W}^{[2]} = [\mathbf{0.393} \quad \mathbf{0.203}], \quad \boldsymbol{b}^{[2]} = [\mathbf{0.531}]$$

Q: Given the following dataset build a Classification Tree able to predict diabetes risk.

ID	Age Group	BMI Category	Family History of Diabates	Activity Level	Diabetes Risk
1	0-29	Normal	Yes	High	No
2	30-45	Overweight	No	Low	Yes
3	46-59	Obese	Yes	Medium	Yes
4	0-29	Normal	No	Medium	No
5	60-79	Obese	Yes	Low	Yes
6	30-45	Overweight	No	High	No
7	0-29	Normal	No	Low	No
8	46-59	Overweight	Yes	Medium	Yes
9	60-79	Obese	No	Low	Yes
10	30-45	Normal	Yes	Low	Yes

We start by computing the **initial entropy $\boldsymbol{H_0}$** of the dataset before considering any split:

$$\boldsymbol{H_0} = \boldsymbol{H}\left(\frac{\boldsymbol{t}}{\boldsymbol{t+f}}, \frac{\boldsymbol{f}}{\boldsymbol{t+f}}\right) = H\left(\frac{6}{6+4}, \frac{4}{6+4}\right) = H\left(\frac{3}{5}, \frac{2}{5}\right) = -\frac{3}{5}log_2\left(\frac{3}{5}\right) - \frac{2}{5}log_2\left(\frac{2}{5}\right) = 0.97$$

Next, we evaluate the discriminatory power of each feature E by computing the entropy **after splitting** based on the considered feature. The **final entropy after splitting based on the feature $\boldsymbol{E}$**, $\boldsymbol{H(E)}$, is computed as the weighted sum of the entropies of its k subsets:

$$\boldsymbol{H(E) = \sum_{i=1}^{k} \frac{t_i + f_i}{t+f} H(E_i)}$$

where each subset entropy $H(E_i)$ is computed as:

$$\boldsymbol{H(E_i) = H\left(\frac{t_i}{t_i+f_i}, \frac{f_i}{t_i+f_i}\right) = -\frac{t_i}{t_i+f_i}\log_2\frac{t_i}{t_i+f_i} - \frac{f_i}{t_i+f_i}\log_2\frac{f_i}{t_i+f_i}}$$

From this, for each feature we calculate the **Information Gain (IG)**, which measures the reduction in entropy of a dataset after splitting it based on a specific feature (in this case E):

$$\boldsymbol{IG(E) = H_0 - H(E)}$$

The feature with the **highest Information Gain** is chosen as decision node to perform the split.

Therefore, let’s calculate the IGs for all the features:

Age Group

This attribute can have the following values: “0-29” (T:0,F:3), “30-45” (T:2 F:1), “46-59” (T:2 F:0), “60-79” (T:2 F:0).

$$IG(Age\ Group) = H_0 - H(Age\ Group) =$$

$$= H_0 - \left[\frac{t_1+f_1}{t+f}H(\text{“}0-29\text{”}) + \frac{t_2+f_2}{t+f}H(\text{“}30-45\text{”}) + \frac{t_3+f_3}{t+f}H(\text{“}46-59\text{”}) + \frac{t_4+f_4}{t+f}H(\text{“}60-79\text{”})\right] =$$

$$= 0.97 - \left[\frac{3}{10}H(0,1) + \frac{3}{10}H\left(\frac{2}{3}, \frac{1}{3}\right) + \frac{2}{10}H(1,0) + \frac{2}{10}H(1,0)\right] =$$

$$= 0.97 - \frac{3}{10} H\left(\frac{2}{3}, \frac{1}{3}\right) =$$

$$= 0.97 - \frac{3}{10}\left(-\frac{2}{3} log_2\left(\frac{2}{3}\right) - \frac{1}{3} log_2\left(\frac{1}{3}\right)\right) =$$

$$= 0.97 - \frac{3}{10} 0.918 = 0.695$$

BMI Category

This attribute can have the following values: “Normal” (T:1,F:3), “Overweight” (T:2 F:1), “Obese” (T:3 F:0).

$$IG(BMI\ Category) = H_0 - H(BMI\ Category) =$$

$$= H_0 - \left[\frac{t_1 + f_1}{t + f} H(\text{Normal}) + \frac{t_2 + f_2}{t + f} H(\text{Overweight}) + \frac{t_3 + f_3}{t + f} H(\text{Obese})\right] =$$

$$= 0.97 - \left[\frac{4}{10} H\left(\frac{1}{4}, \frac{3}{4}\right) + \frac{3}{10} H\left(\frac{2}{3}, \frac{1}{3}\right) + \frac{3}{10} H(1,0)\right] =$$

$$= 0.97 - \left[\frac{4}{10} H\left(\frac{1}{4}, \frac{3}{4}\right) + \frac{3}{10} H\left(\frac{2}{3}, \frac{1}{3}\right)\right] =$$

$$= 0.97 - \left[\frac{4}{10}\left(-\frac{1}{4} log_2\left(\frac{1}{4}\right) - \frac{3}{4} log_2\left(\frac{3}{4}\right)\right) + \frac{3}{10}\left(-\frac{2}{3} log_2\left(\frac{2}{3}\right) - \frac{1}{3} log_2\left(\frac{1}{3}\right)\right)\right] =$$

$$= 0.97 - \left[\frac{4}{10} 0.811 + \frac{3}{10} 0.918\right] = 0.370$$

Family History of Diabates

This attribute can have the following values: “Yes” (T:4,F:1), “No” (T:2,F:3).

$$IG(Family\ History\ of\ Diabates) = H_0 - H(Family\ History\ of\ Diabates) =$$

$$= H_0 - \left[\frac{t_1 + f_1}{t + f} H(\text{Yes}) + \frac{t_2 + f_2}{t + f} H(\text{No})\right] =$$

$$= 0.97 - \left[\frac{5}{10} H\left(\frac{4}{5}, \frac{1}{5}\right) + \frac{5}{10} H\left(\frac{2}{5}, \frac{3}{5}\right)\right] =$$

$$= 0.97 - \left[\frac{5}{10}\left(-\frac{4}{5} log_2\left(\frac{4}{5}\right) - \frac{1}{5} log_2\left(\frac{1}{5}\right)\right) + \frac{5}{10}\left(-\frac{2}{5} log_2\left(\frac{2}{5}\right) - \frac{3}{5} log_2\left(\frac{3}{5}\right)\right)\right] =$$

$$= 0.97 - \left[\frac{5}{10} 0.722 + \frac{5}{10} 0.971\right] = 0.124$$

Activity Level

This attribute can have the following values: “Low” (T:4,F:1), “Medium” (T:2,F:1), “High” (T:0,F:2).

$$IG(Activity\ Level) = H_0 - H(Activity\ Level) =$$

$$= H_0 - \left[\frac{t_1 + f_1}{t + f} H(\text{Low}) + \frac{t_2 + f_2}{t + f} H(\text{Medium}) + \frac{t_3 + f_3}{t + f} H(\text{High})\right] =$$

$$= 0.97 - \left[\frac{5}{10} H\left(\frac{4}{5}, \frac{1}{5}\right) + \frac{3}{10} H\left(\frac{2}{3}, \frac{1}{3}\right) + \frac{2}{10} H(0,1)\right] =$$

$$= 0.97 - \left[\frac{5}{10} H\left(\frac{4}{5}, \frac{1}{5}\right) + \frac{3}{10} H\left(\frac{2}{3}, \frac{1}{3}\right)\right] =$$

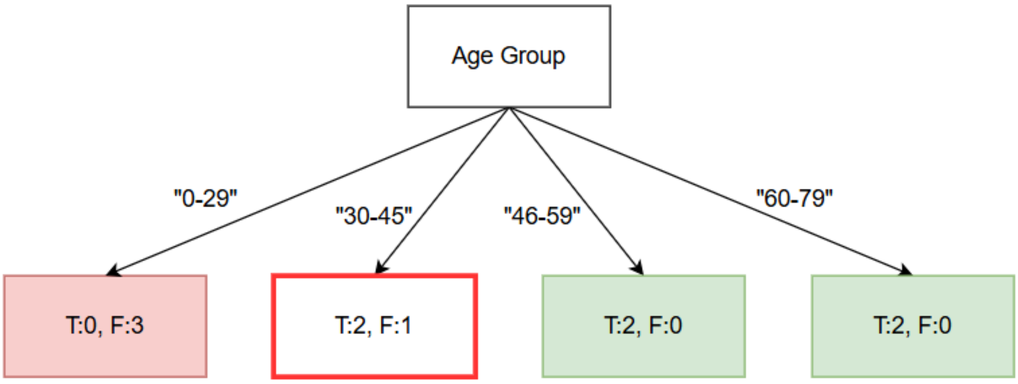
$$= 0.97 - \left[\frac{5}{10}\left(-\frac{4}{5} log_2\left(\frac{4}{5}\right) - \frac{1}{5} log_2\left(\frac{1}{5}\right)\right) + \frac{3}{10}\left(-\frac{2}{3} log_2\left(\frac{2}{3}\right) - \frac{1}{3} log_2\left(\frac{1}{3}\right)\right)\right] =$$

$$= 0.97 - \left[\frac{5}{10} 0.722 + \frac{3}{10} 0.918\right] = 0.334$$

We choose the feature with the highest information gain:

$$\max(IG) = IG(Age\ Group) = 0.695$$

We make a tree using this feature and check if there are impure leaves.



The tree has an impure leaf. This needs to be split using another feature. We reduce the dataset keeping only the tuples in which ***Age Group = 30-45*** and calculate the information gain of the remaining features.

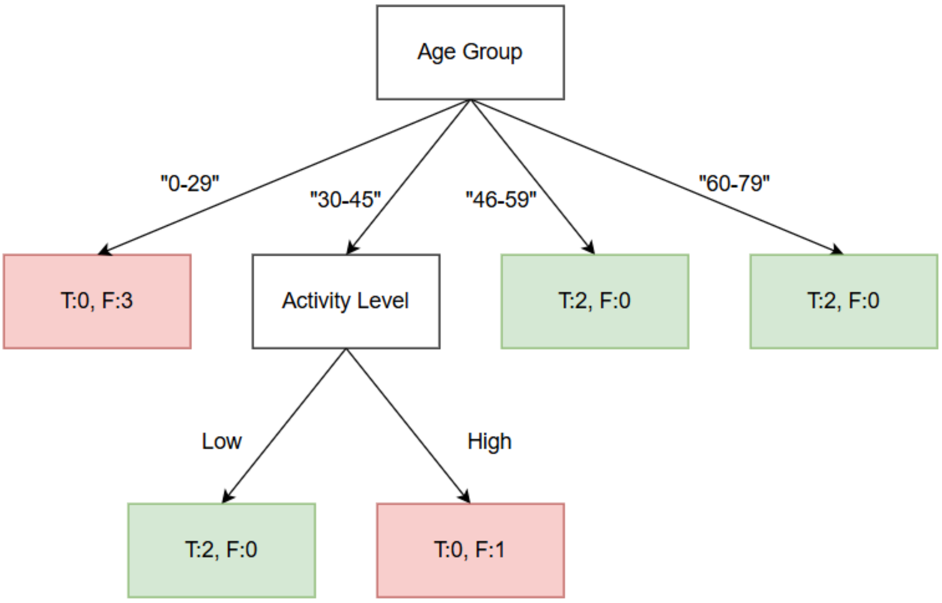
ID	Age Group	BMI Category	Family History of Diabates	Activity Level	Diabetes Risk
~~1~~	~~0-29~~	~~Normal~~	~~Yes~~	~~High~~	~~No~~
2	30-45	Overweight	No	Low	Yes
~~3~~	~~46-59~~	~~Obese~~	~~Yes~~	~~Medium~~	~~Yes~~
~~4~~	~~0-29~~	~~Normal~~	~~No~~	~~Medium~~	~~No~~
~~5~~	~~60-79~~	~~Obese~~	~~Yes~~	~~Low~~	~~Yes~~
6	30-45	Overweight	No	High	No
~~7~~	~~0-29~~	~~Normal~~	~~No~~	~~Low~~	~~No~~
~~8~~	~~46-59~~	~~Overweight~~	~~Yes~~	~~Medium~~	~~Yes~~
~~9~~	~~60-79~~	~~Obese~~	~~No~~	~~Low~~	~~Yes~~
10	30-45	Normal	Yes	Low	Yes

By inspection we observe that the target variable depends on the feature "Activity Level". The final entropy choosing that feature is zero, so the information gain is maximum. The information gains of the other features are smaller.

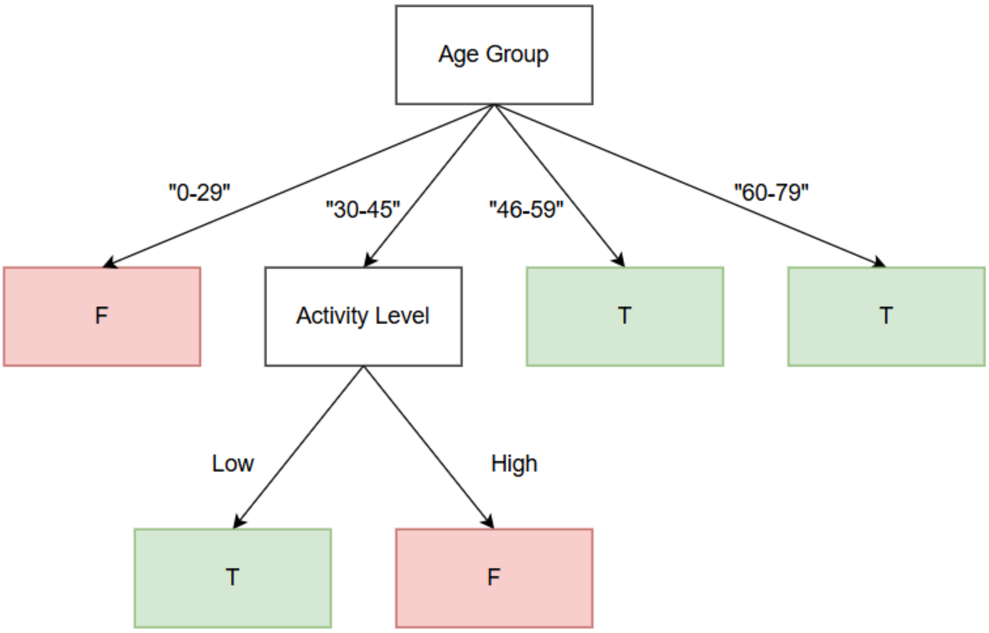
We can say that:

$$Activity\ Level = Low \rightarrow Diabetes\ Risk = Yes$$

$$Activity\ Level = High \rightarrow Diabetes\ Risk = No$$



There are no impure leaves in the tree, so the building of the tree is complete. The final classification tree is therefore as follows:



Note: I have created a Jupyter Notebook that you can use to verify your solution or experiment with building a new classification tree. Simply provide the dataset, and the notebook will generate a classification tree following the logic covered during the course.

Q: Build a regression tree to predict drug effectiveness using the provided dataset. Follow these guidelines:

- **For continuous variables (e.g., Dosage, Age), consider a number of thresholds equal to 3 such that to divide their range into four evenly spaced intervals.**
- **For discrete variables (e.g., Sex), consider all possible unique values as thresholds for splitting.**
- **Continue splitting a node only if it contains at least 7 observations, otherwise treat it as a leaf.**

Dosage	Age	Sex	Drug Effectiveness
26	59	0	5
41	19	0	50
45	45	0	50
36	46	1	100
24	55	1	8
35	51	1	9
36	27	1	100
30	46	0	50
44	24	1	100
11	59	0	4
49	20	0	50
38	35	1	100
23	58	0	8
39	23	0	50
34	37	0	50
47	52	1	9
23	39	1	15
13	48	0	15
29	21	0	50
18	30	0	24
10	33	0	22
26	47	1	100
28	36	0	50
12	37	0	31
17	30	1	23
17	47	1	16

Step 1: Let's calculate the **Residual Sum of Squares (RSS)** for different thresholds on each feature. At the end we choose the pair $(feature, threshold)$ that gives us the lowest RSS.

Dosage:

Range: $min = 10,\ \ max = 49$ → Thresholds: $T_1 = 19.75,\ \ T_2 = 29.5,\ \ T_3 = 39.25$

- **Split on Dosage with Threshold = 19.75:**

  Right Group: Dosage > 19.75

  - Observations y_{right}: $[5, 50, 50, 100, 8, 9, 100, 50, 100, 50, 100, 8, 50, 50, 9, 15, 50, 100, 50]$
  - $\bar{y}_{right} = \frac{5+50+50+100+8+9+100+50+100+50+100+8+50+50+9+15+50+100+50}{19} = 50.21$

  Left Group: Dosage ≤ 19.75

  - Observations y_{left}: $[4, 15, 24, 22, 31, 23, 16]$

- $\bar{y}_{left} = \frac{4+15+24+22+31+23+16}{7} = 19.29$

➔ $h(x^{(i)}) = \begin{cases} \bar{y}_{right}, & \text{Dosage} > 19.75 \\ \bar{y}_{left}, & \text{Dosage} \leq 19.75 \end{cases}$

$$RSS = \sum\left(h(x^{(i)}) - y^{(i)}\right)^2 = \sum\left(\bar{y}_{right} - y_{right}^{(i)}\right)^2 + \sum\left(\bar{y}_{left} - y_{left}^{(i)}\right)^2 = 22639.16 + 443.43 = 23082.59$$

- **Split on Dosage with Threshold = 29.5:**

  Right Group: Dosage > 29.5

  - Observations y_{right}: $[50, 50, 100, 9, 100, 50, 100, 50, 100, 50, 50, 9]$
  - $\bar{y}_{right} = \frac{50+50+100+9+100+50+100+50+100+50+50+9}{12} = 59.83$

  Left Group: Dosage ≤ 29.5

  - Observations y_{left}: $[5, 8, 4, 8, 15, 15, 50, 24, 22, 100, 50, 31, 23, 16]$
  - $\bar{y}_{left} = \frac{5+8+4+8+15+15+50+24+22+100+50+31+23+16}{14} = 26.50$

➔ $h(x^{(i)}) = \begin{cases} \bar{y}_{right}, & \text{Dosage} > 29.5 \\ \bar{y}_{left}, & \text{Dosage} \leq 29.5 \end{cases}$

$$RSS = \sum\left(h(x^{(i)}) - y^{(i)}\right)^2 = \sum\left(\bar{y}_{right} - y_{right}^{(i)}\right)^2 + \sum\left(\bar{y}_{left} - y_{left}^{(i)}\right)^2 = 12201.67 + 8593.50 = 20795.17$$

- **Split on Dosage with Threshold = 39.25:**

  Right Group: Dosage > 39.25

  - Observations y: $[50,50,100,50,9]$
  - $\bar{y}_{right} = \frac{50+50+100+50+9}{5} = 51.80$

  Left Group: Dosage ≤ 39.25

  - Observations y: $[5, 100, 8, 9, 100, 50, 4, 100, 8, 50, 50, 15, 15, 50, 24, 22, 100, 50, 31, 23, 16]$
  - $\bar{y}_{left} = \frac{5+100+8+9+100+50+4+100+8+50+50+15+15+50+24+22+100+50+31+23+16}{21} = 39.52$

➔ $h(x^{(i)}) = \begin{cases} \bar{y}_{right}, & \text{Dosage} > 39.25 \\ \bar{y}_{left}, & \text{Dosage} \leq 39.25 \end{cases}$

$$RSS = \sum\left(h(x^{(i)}) - y^{(i)}\right)^2 = \sum\left(\bar{y}_{right} - y_{right}^{(i)}\right)^2 + \sum\left(\bar{y}_{left} - y_{left}^{(i)}\right)^2 = 4164.80 + 23201.24 = 27366.04$$

Age

Range: $min = 19,\ \ max = 59$ → Thresholds: $T_1 = 29,\ \ T_2 = 39,\ \ T_3 = 49$

- **Split on Age with Threshold = 29:**

  Right Group: Age > 29

- Observations y_{right}: $[5, 50, 100, 8, 9, 50, 4, 100, 8, 50, 9, 15, 15, 24, 22, 100, 50, 31, 23, 16]$
- $\bar{y}_{right} = \frac{5+50+100+8+9+50+4+100+8+50+9+15+15+24+22+100+50+31+23+16}{20} = 34.45$

Left Group: Age ≤ 29

- Observations y_{left}: $[50, 100, 100, 50, 50, 50]$
- $\bar{y}_{left} = \frac{50+100+100+50+50+50}{6} = 66.67$

➔ $h(x^{(i)}) = \begin{cases} \bar{y}_{right}, & \text{Age} > 29 \\ \bar{y}_{left}, & \text{Age} \leq 29 \end{cases}$

$$RSS = \sum \left(h(x^{(i)}) - y^{(i)}\right)^2 = \sum \left(\bar{y}_{right} - y^{(i)}_{right}\right)^2 + \sum \left(\bar{y}_{left} - y^{(i)}_{left}\right)^2 = 19850.95 + 3333.33 = 23184.28$$

- **Split on Age with Threshold = 39:**

Right Group: Age > 39

- Observations y_{right}: $[5,50,100,8,9,50,4,8,9,15,100,16]$
- $\bar{y}_{right} = \frac{5+50+100+8+9+50+4+8+9+15+100+16}{12} = 31.67$

Left Group: Age ≤ 39

- Observations y_{left}: $[50,100,100,50,100,50,50,15,50,24,22,50,31,23]$
- $\bar{y}_{left} = \frac{50+100+100+50+100+50+50+15+50+24+22+50+31+23}{14} = 55.14$

➔ $h(x^{(i)}) = \begin{cases} \bar{y}_{right}, & \text{Age} > 39 \\ \bar{y}_{left}, & \text{Age} \leq 39 \end{cases}$

$$RSS = \sum \left(h(x^{(i)}) - y^{(i)}\right)^2 = \sum \left(\bar{y}_{right} - y^{(i)}_{right}\right)^2 + \sum \left(\bar{y}_{left} - y^{(i)}_{left}\right)^2 = 9174.67 + 13761.14 = 22935.81$$

- **Split on Age with Threshold = 49:**

Right Group: Age > 49

- Observations y_{right}: $[5,8,9,4,8,9]$
- $\bar{y}_{right} = \frac{5+8+9+4+8+9}{6} = 7.17$

Left Group: Age ≤ 49

- Observations y_{left}: $[50,50,100,100,50,100,50,100,50,50,15,15,50,24,22,100,50,31,23,16]$
- $\bar{y}_{left} = \frac{50+50+100+100+50+100+50+100+50+50+15+15+50+24+22+100+50+31+23+16}{20} = 54.00$

➔ $h(x^{(i)}) = \begin{cases} \bar{y}_{right}, & \text{Age} > 49 \\ \bar{y}_{left}, & \text{Age} \leq 49 \end{cases}$

$$RSS = \sum \left(h(x^{(i)}) - y^{(i)}\right)^2 = \sum \left(\bar{y}_{right} - y^{(i)}_{right}\right)^2 + \sum \left(\bar{y}_{left} - y^{(i)}_{left}\right)^2 = 24.83 + 17045.00 = 17069.83$$

Sex:

Discrete values are $\{0,1\}$

- **Split on Sex with Threshold = 0:**

  Group 0: Sex == 0

  - Observations y_0: $[5,50,50,50,4,50,8,50,50,15,50,24,22,50,31]$
  - $\bar{y}_0 = \frac{5+50+50+50+4+50+8+50+50+15+50+24+22+50+31}{15} = 36.80$

  Group 1: Sex != 0 (which in this case means Sex == 1)

  - Observations y_1: $[100,8,9,100,100,100,9,15,100,23,16]$
  - $\bar{y}_1 = \frac{100+8+9+100+100+100+9+15+100+23+16}{11} = 61.91$

➔ $h\left(x^{(i)}\right) = \begin{cases} \bar{y}_0, & \text{Sex} == \ 0 \\ \bar{y}_1, & \text{Sex} \, ! = 0 \end{cases}$

$$RSS = \sum \left(h\left(x^{(i)}\right) - y^{(i)}\right)^2 = \sum \left(\bar{y}_0 - y_0^{(i)}\right)^2 + \sum \left(\bar{y}_1 - y_1^{(i)}\right)^2 = 10952.40 + 11516.91 = 22469.31$$

Note: We didn't test the threshold value of 1 because it is symmetric to the threshold value of 0. Both would result in the same Residual Sum of Squares (RSS). Therefore, in this case, we chose to use only the threshold value of 0.

Let's see what we got:

- Best RSS for Dosage is 20,795.17 corresponding to Threshold = 29.5.
- Best RSS for Age is 17,069.83 corresponding to Threshold = 49.
- Best RSS for Sex is 22,469.31 corresponding to Threshold = 0.

→ Therefore, the **Best Overall feature to choose is Age>49**

Now, for Age > 49 we have just 6 samples, so we conclude it with a leaf. The prediction value for that leaf is **7.17**.

For Age ≤ 49, instead, since we have 20 sample we need to continue the splitting. Specifically, we will continue to split just this subset of entries, considering only the remaining features, i.e. Dosage and Sex (I colored with red, the rows and column that should NOT be considered).

Dosage	Age	Sex	Drug Effectiveness
26	59	0	5
41	19	0	50
45	45	0	50
36	46	1	100
24	55	1	8
35	51	1	9
36	27	1	100
30	46	0	50
44	24	1	100
11	59	0	4
49	20	0	50
38	35	1	100

23	58	0	8
39	23	0	50
34	37	0	50
47	52	1	9
23	39	1	15
13	48	0	15
29	21	0	50
18	30	0	24
10	33	0	22
26	47	1	100
28	36	0	50
12	37	0	31
17	30	1	23
17	47	1	16

Step 2: Let's repeat the process we did above again, but this time just considering this subset of entries and the remaining features.

Dosage:

Range: $min = 10, \;\; max = 49$ ➔ Thresholds: $T_1 = 19.75, \;\; T_2 = 29.5, \;\; T_3 = 39.25$ (min and max are the same, so also the thresholds remained the same).

- **Split on Dosage with Threshold = 19.75:**

  Right Group: Dosage > 19.75

  - Observations y_{right}: $[50{,}50{,}100{,}100{,}50{,}100{,}50{,}100{,}50{,}50{,}15{,}50{,}100{,}50]$
  - $\bar{y}_{right} = \frac{50+50+100+100+50+100+50+100+50+50+15+50+100+50}{14} = 65.36$

  Left Group: Dosage ≤ 19.75

  - Observations y_{left}: $[15{,}24{,}22{,}31{,}23{,}16]$
  - $\bar{y}_{left} = \frac{15+24+22+31+23+16}{6} = 21.83$

➔ $h(x^{(i)}) = \begin{cases} \bar{y}_{right}, & \text{Dosage } > \; 19.75 \\ \bar{y}_{left}, & \text{Dosage } \leq \; 19.75 \end{cases}$

$$RSS = \sum\left(h(x^{(i)}) - y^{(i)}\right)^2 = \sum\left(\bar{y}_{right} - y_{right}^{(i)}\right)^2 + \sum\left(\bar{y}_{left} - y_{left}^{(i)}\right)^2 = 10423.21 + 170.83 = 10594.04$$

- **Split on Dosage with Threshold = 29.5:**

  Right Group: Dosage > 29.5

  - Observations y_{right}: $[50{,}50{,}100{,}100{,}50{,}100{,}50{,}100{,}50{,}50]$
  - $\bar{y}_{right} = \frac{50+50+100+100+50+100+50+100+50+50}{10} = 70.00$

  Left Group: Dosage ≤ 29.5

  - Observations y_{left}: $[15{,}15{,}50{,}24{,}22{,}100{,}50{,}31{,}23{,}16]$

- $\bar{y}_{left} = \frac{15+15+50+24+22+100+50+31+23+16}{10} = 34.60$

➔ $h(x^{(i)}) = \begin{cases} \bar{y}_{right}, & \text{Dosage} > 29.5 \\ \bar{y}_{left}, & \text{Dosage} \leq 29.5 \end{cases}$

$$RSS = \sum \left(h(x^{(i)}) - y^{(i)}\right)^2 = \sum \left(\bar{y}_{right} - y^{(i)}_{right}\right)^2 + \sum \left(\bar{y}_{left} - y^{(i)}_{left}\right)^2 = 6000.00 + 6284.40 = 12284.40$$

- **Split on Dosage with Threshold = 39.25:**

  Right Group: Dosage > 39.25

  - Observations y_{right}: [50,50,100,50]
  - $\bar{y}_{right} = \frac{50+50+100+50}{4} = 62.50$

  Left Group: Dosage ≤ 39.25

  - Observations y_{left}: [100,100,50,100,50,50,15,15,50,24,22,100,50,31,23,16]
  - $\bar{y}_{left} = \frac{100+100+50+100+50+50+15+15+50+24+22+100+50+31+23+16}{16} = 49.75$

➔ $h(x^{(i)}) = \begin{cases} \bar{y}_{right}, & \text{Dosage} > 39.25 \\ \bar{y}_{left}, & \text{Dosage} \leq 39.25 \end{cases}$

$$RSS = \sum \left(h(x^{(i)}) - y^{(i)}\right)^2 = \sum \left(\bar{y}_{right} - y^{(i)}_{right}\right)^2 + \sum \left(\bar{y}_{left} - y^{(i)}_{left}\right)^2 = 1875.00 + 16155.00 = 18030.00$$

Sex:

Discrete values are $\{0,1\}$

- **Split on Sex with Threshold = 0:**

  Group 0: Sex == 0

  - Observations y_0: [50,50,50,50,50,50,15,50,24,22,50,31]
  - $\bar{y}_0 = \frac{50+50+50+50+50+50+15+50+24+22+50+31}{12} = 41.00$

  Group 1: Sex != 0

  - Observations y_1: [100,100,100,100,15,100,23,16]
  - $\bar{y}_1 = \frac{100+100+100+100+15+100+23+16}{8} = 69.25$

➔ $h(x^{(i)}) = \begin{cases} \bar{y}_0, & \text{Sex} == 0 \\ \bar{y}_1, & \text{Sex} != 0 \end{cases}$

$$RSS = \sum \left(h(x^{(i)}) - y^{(i)}\right)^2 = \sum \left(\bar{y}_0 - y^{(i)}_0\right)^2 + \sum \left(\bar{y}_1 - y^{(i)}_1\right)^2 = 2074.00 + 12645.50 = 14719.50$$

Let's see what we got:

- Best RSS for Dosage is $\mathbf{10594.04}$ corresponding to Threshold = 19.75.
- Best RSS for Sex is 14719.50 corresponding to Threshold = 0.

→ Therefore, the **Best Overall feature to choose is Dosage>19.75**.

Now, for Dosage <= 19.75 we have just 6 samples, so we conclude it with a leaf. The prediction value for that leaf is **21.83**.

For Dosage > 19.75, instead, since we have 14 samples we continue the splitting. Specifically, we will continue to split just this subset of entries, considering only the remaining feature, i.e. Sex.

Dosage	Age	Sex	Drug Effectiveness
26	59	0	5
41	19	0	50
45	45	0	50
36	46	1	100
24	55	1	8
35	51	1	9
36	27	1	100
30	46	0	50
44	24	1	100
11	59	0	4
49	20	0	50
38	35	1	100
23	58	0	8
39	23	0	50
34	37	0	50
47	52	1	9
23	39	1	15
13	48	0	15
29	21	0	50
18	30	0	24
10	33	0	22
26	47	1	100
28	36	0	50
12	37	0	31
17	30	1	23
17	47	1	16

Step 3: Examining the remaining entries, we observe that:

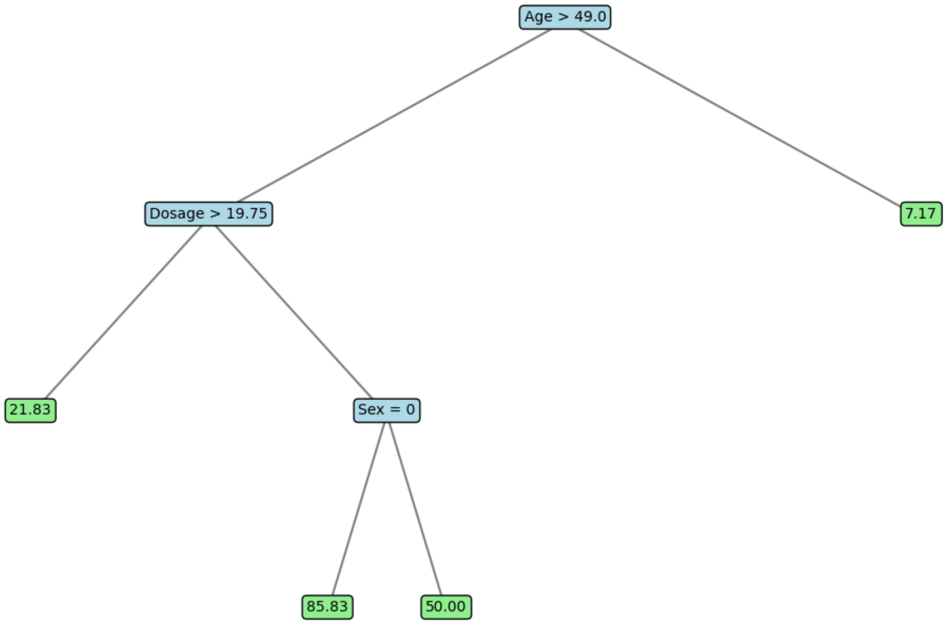
- for Sex=0 we have:
  - Observations y: $[50,50,50,50,50,50]$
  - $\bar{y}_0 = \frac{50+50+50+50+50+50}{6} = 50$
- while for Sex=1 we have:
  - Observations y: $[100,100,100,100,15,100]$
  - $\bar{y}_1 = \frac{100+100+100+100+15+100}{6} = 85.83$

Since each category (Sex = 0 and Sex = 1) contains exactly 6 entries, we can split based on **Sex = 0**. This will result in:

1. A leaf node containing the 6 entries for which **Sex = 0** having a prediction value of **50**.

2. Another leaf node with the remaining 6 entries (for **Sex = 1**) having a prediction value of approximately **85.83**.

This concludes the construction of the regression tree that we would look like:



Note: I have also created another Jupyter Notebook that you can use to verify your solution or experiment with building a new regression tree. While it is highly unlikely that this will appear on the exam, it's a good exercise to reinforce your understanding.

Q: Apply a single iteration of K-Means clustering to group six data points

$$\boldsymbol{x^{(1)} = 3, x^{(2)} = 8, x^{(3)} = 5, x^{(4)} = 1, x^{(5)} = 2, x^{(6)} = 4}$$

into two clusters ($K = 2$) based on the following given distances between pairs of data points:

$$\boldsymbol{d} = \left[\begin{array}{ccccccc} \backslash & \boldsymbol{x^{(1)}} & \boldsymbol{x^{(2)}} & \boldsymbol{x^{(3)}} & \boldsymbol{x^{(4)}} & \boldsymbol{x^{(5)}} & \boldsymbol{x^{(6)}} \\ \boldsymbol{x^{(1)}} & \mathbf{0} & \mathbf{2} & \mathbf{6} & \mathbf{1} & \mathbf{2} & \mathbf{8} \\ \boldsymbol{x^{(2)}} & \mathbf{2} & \mathbf{0} & \mathbf{5} & \mathbf{9} & \mathbf{8} & \mathbf{7} \\ \boldsymbol{x^{(3)}} & \mathbf{6} & \mathbf{5} & \mathbf{0} & \mathbf{4} & \mathbf{3} & \mathbf{3} \\ \boldsymbol{x^{(4)}} & \mathbf{1} & \mathbf{9} & \mathbf{4} & \mathbf{0} & \mathbf{1} & \mathbf{2} \\ \boldsymbol{x^{(5)}} & \mathbf{2} & \mathbf{8} & \mathbf{3} & \mathbf{1} & \mathbf{0} & \mathbf{1} \\ \boldsymbol{x^{(6)}} & \mathbf{8} & \mathbf{7} & \mathbf{3} & \mathbf{2} & \mathbf{1} & \mathbf{0} \end{array}\right]$$

Initial centroids are chosen randomly as:

$$\boldsymbol{\mu_1 = x^{(1)}}, \qquad \boldsymbol{\mu_2 = x^{(3)}}$$

+ Then do the same by applying a single iteration of the K-medoid clustering algorithm.

.. K-Means ....

Since the centroids have already been initialized in the track, we proceed by assigning each data point to the closest cluster based on their distance from each cluster centroid.

Note: In this case the distances are already given so we need just to perform the assignment.

$d(x^{(2)}, \mu_1) = 2, \quad d(x^{(2)}, \mu_2) = 5 \rightarrow$ $x^{(2)}$ is assigned to C_1 (cluster 1)

$d(x^{(4)}, \mu_1) = 1, \quad d(x^{(4)}, \mu_2) = 4 \rightarrow$ $x^{(4)}$ is assigned to C_1 (cluster 2)

$d(x^{(5)}, \mu_1) = 2, \quad d(x^{(5)}, \mu_2) = 3 \rightarrow$ $x^{(5)}$ is assigned to C_1

$d(x^{(6)}, \mu_1) = 8, \quad d(x^{(6)}, \mu_2) = 3 \rightarrow$ $x^{(6)}$ is assigned to C_2

→ **Clusters after assignment:**

$$C_1: \{x^{(1)}, x^{(2)}, x^{(4)}, x^{(5)}\} \qquad C_2: \{x^{(3)}, x^{(6)}\}$$

For each cluster, we recalculate the centroid as the mean of the points in the cluster.

for $k = 1$ *to* K

$$\mu_k := \frac{1}{|C_k|} \sum_{x^{(i)} \in C_k} x^{(i)}$$

- For C_1 we have:

$$\mu_1 = \frac{1}{4}\left(x^{(1)} + x^{(2)} + x^{(4)} + x^{(5)}\right) = \frac{3+8+1+2}{4} = 3.5$$

- For C_2 we have:

$$\mu_2 = \frac{1}{2}\left(x^{(3)} + x^{(6)}\right) = \frac{5+4}{2} = 4.5$$

→ **Updated Centroids:**

$$\mu_1 = 3.5 \qquad \mu_2 = 4.5$$

.. K-Medoids ....

Since the centroids have already been initialized in the track, we proceed by assigning each data point to the closest cluster based on their distance from each cluster medoid:

$d(x^{(2)}, \mu_1) = 2, \quad d(x^{(2)}, \mu_2) = 5 \rightarrow$ $x^{(2)}$ is assigned to C_1

$d(x^{(4)}, \mu_1) = 1, \quad d(x^{(4)}, \mu_2) = 4 \rightarrow$ $x^{(4)}$ is assigned to C_1

$d(x^{(5)}, \mu_1) = 2, \quad d(x^{(5)}, \mu_2) = 3 \rightarrow$ $x^{(5)}$ is assigned to C_1

$d(x^{(6)}, \mu_1) = 8, \quad d(x^{(6)}, \mu_2) = 3 \rightarrow$ $x^{(6)}$ is assigned to C_2

→ Current Clusters and Medoids (after the assignment):

$$C_1: \{x^{(1)}, x^{(2)}, x^{(4)}, x^{(5)}\}, \quad \mu_1 = x^{(1)}$$

$$C_2: \{x^{(3)}, x^{(6)}\}, \qquad \mu_2 = x^{(3)}$$

Compute the current $Cost$:

- $C_1: d(x^{(2)}, \mu_1) + d(x^{(4)}, \mu_1) + d(x^{(5)}, \mu_1) = 2 + 1 + 2 = 5$
- $C_2: d(x^{(6)}, \mu_2) = 3$

→ $\boldsymbol{Cost = 5 + 3 = 8}$

At this point, for each cluster and for each data point that is not a medoid, exchange $x^{(i)}$ with μ_k:

Swap $x^{(2)}$ with μ_1:

- New medoids: $\boldsymbol{\mu_1 = x^{(2)} = 8}$, $\mu_2 = x^{(3)} = 5$.
- Re-associate each data point to closest medoid:

$$\begin{aligned}
d(x^{(1)}, \mu_1) = 2, \quad & d(x^{(1)}, \mu_2) = 6 \rightarrow x^{(1)} \text{ is assigned to } C_1 \\
d(x^{(4)}, \mu_1) = 9, \quad & d(x^{(4)}, \mu_2) = 4 \rightarrow x^{(4)} \text{ is assigned to } C_2 \\
d(x^{(5)}, \mu_1) = 8, \quad & d(x^{(5)}, \mu_2) = 3 \rightarrow x^{(5)} \text{ is assigned to } C_2 \\
d(x^{(6)}, \mu_1) = 7, \quad & d(x^{(6)}, \mu_2) = 3 \rightarrow x^{(6)} \text{ is assigned to } C_2
\end{aligned}$$

$$C_1: \{x^{(1)}, x^{(2)}\} \qquad C_2: \{x^{(3)}, x^{(4)}, x^{(5)}, x^{(6)}\}$$

- Re-compute $NewCost$:
  - $C_1: d(x^{(1)}, \mu_1) = 2$
  - $C_2: d(x^{(4)}, \mu_2) + d(x^{(5)}, \mu_2) + d(x^{(6)}, \mu_2) = 4 + 3 + 3 = 10$
- $\boldsymbol{NewCost = 2 + 10 = 12}$

Since $\boldsymbol{NewCost} > \boldsymbol{Cost}$, given that $12 > 8$, we **undo the swap**.

Swap $x^{(4)}$ with μ_1:

- New medoids: $\boldsymbol{\mu_1 = x^{(4)} = 1}$, $\mu_2 = x^{(3)} = 5$.
- Re-associate each data point to closest medoid:

$$\begin{aligned}
d(x^{(1)}, \mu_1) = 1, \quad & d(x^{(1)}, \mu_2) = 6 \rightarrow x^{(1)} \text{ is assigned to } C_1 \\
d(x^{(2)}, \mu_1) = 9, \quad & d(x^{(2)}, \mu_2) = 5 \rightarrow x^{(2)} \text{ is assigned to } C_2 \\
d(x^{(5)}, \mu_1) = 1, \quad & d(x^{(5)}, \mu_2) = 3 \rightarrow x^{(5)} \text{ is assigned to } C_1 \\
d(x^{(6)}, \mu_1) = 2, \quad & d(x^{(6)}, \mu_2) = 3 \rightarrow x^{(6)} \text{ is assigned to } C_1
\end{aligned}$$

$$C_1: \{x^{(1)}, x^{(4)}, x^{(5)}, x^{(6)}\} \qquad C_2: \{x^{(2)}, x^{(3)}\}$$

- Re-compute $NewCost$:
  - $C_1: d(x^{(1)}, \mu_1) + d(x^{(5)}, \mu_1) + d(x^{(6)}, \mu_1) = 1 + 1 + 2 = 4$
  - $C_2: d(x^{(2)}, \mu_2) = 5$
- $\boldsymbol{NewCost = 4 + 5 = 9}$

Since $\boldsymbol{NewCost} > \boldsymbol{Cost}$, given that $9 > 8$, we **undo the swap**.

Swap $\boldsymbol{x^{(5)}}$ with $\boldsymbol{\mu_1}$:

- New medoids: $\boldsymbol{\mu_1 = x^{(5)} = 2}$, $\mu_2 = x^{(3)} = 5$.
- Re-associate each data point to closest medoid:

$$\begin{aligned} d(x^{(1)}, \mu_1) = 2, \quad & d(x^{(1)}, \mu_2) = 6 \rightarrow x^{(1)} \text{ is assigned to } C_1 \\ d(x^{(2)}, \mu_1) = 8, \quad & d(x^{(2)}, \mu_2) = 5 \rightarrow x^{(2)} \text{ is assigned to } C_2 \\ d(x^{(4)}, \mu_1) = 1, \quad & d(x^{(4)}, \mu_2) = 4 \rightarrow x^{(5)} \text{ is assigned to } C_1 \\ d(x^{(6)}, \mu_1) = 1, \quad & d(x^{(6)}, \mu_2) = 3 \rightarrow x^{(6)} \text{ is assigned to } C_1 \end{aligned}$$

$$C_1: \{x^{(1)}, x^{(4)}, x^{(5)}, x^{(6)}\} \qquad C_2: \{x^{(2)}, x^{(3)}\}$$

- Re-compute $NewCost$:
  - $C_1: d(x^{(1)}, \mu_1) + d(x^{(4)}, \mu_1) + d(x^{(6)}, \mu_1) = 2 + 1 + 1 = 4$
  - $C_2: d(x^{(2)}, \mu_2) = 5$
- $NewCost = 4 + 5 = 9$

Since $\boldsymbol{NewCost > Cost}$, given that $9 > 8$, we **undo the swap**.

Swap $\boldsymbol{x^{(6)}}$ with $\boldsymbol{\mu_2}$:

- New medoids: $\mu_1 = x^{(1)} = 3$, $\boldsymbol{\mu_2 = x^{(6)} = 4}$.
- New Clusters:

$$\begin{aligned} d(x^{(2)}, \mu_1) = 2, \quad & d(x^{(2)}, \mu_2) = 7 \rightarrow x^{(2)} \text{ is assigned to } C_1 \\ d(x^{(3)}, \mu_1) = 6, \quad & d(x^{(3)}, \mu_2) = 3 \rightarrow x^{(3)} \text{ is assigned to } C_2 \\ d(x^{(4)}, \mu_1) = 1, \quad & d(x^{(4)}, \mu_2) = 2 \rightarrow x^{(4)} \text{ is assigned to } C_1 \\ d(x^{(5)}, \mu_1) = 2, \quad & d(x^{(5)}, \mu_2) = 1 \rightarrow x^{(5)} \text{ is assigned to } C_2 \end{aligned}$$

$$C_1: \{x^{(1)}, x^{(2)}, x^{(4)}\} \qquad C_2: \{x^{(3)}, x^{(5)}, x^{(6)}\}$$

- Cluster costs:
  - $C_1: d(x^{(2)}, \mu_1) + d(x^{(4)}, \mu_1) = 2 + 1 = 3$
  - $C_2: d(x^{(3)}, \mu_2) + d(x^{(5)}, \mu_2) = 3 + 1 = 4$
- $NewCost = 3 + 4 = 7$

Since $\boldsymbol{NewCost < Cost}$, given that $7 < 8$, we **confirm the swap**:

$$Cost = NewCost = 7$$

New medoids: $\boldsymbol{\mu_1 = x^{(1)} = 3}$, $\boldsymbol{\mu_2 = x^{(6)} = 4}$

Q: Given the dataset $\boldsymbol{X}$, with covariance matrix $\boldsymbol{\Sigma}$, we obtained the eigenvector matrix $\boldsymbol{U}$ and the eigenvalue vector $\boldsymbol{s} = (\mathbf{11.2912878},\ \mathbf{6.70871215},\ \mathbf{3.9171372510^{-16}})$. Use the PCA technique to reduce the dimensionality of $\boldsymbol{X}$ retaining at least the 60% of variance of the original dataset. Finally, reconstruct the original data.

$$\boldsymbol{X} = \begin{pmatrix} \mathbf{1} & \mathbf{5} & \mathbf{3} \\ \mathbf{3} & \mathbf{4} & \mathbf{8} \\ \mathbf{5} & \mathbf{1} & \mathbf{6} \end{pmatrix}$$

$$\boldsymbol{\Sigma} = \begin{pmatrix} \mathbf{4} & \mathbf{1} & \mathbf{-4} \\ \mathbf{1} & \mathbf{7} & \mathbf{3.5} \\ \mathbf{-4} & \mathbf{3.5} & \mathbf{7} \end{pmatrix}$$

$$\boldsymbol{U} = \begin{pmatrix} \mathbf{-0.34517975} & \mathbf{0.54357965} & \mathbf{0.76509614} \\ \mathbf{-0.62904883} & \mathbf{-0.73898939} & \mathbf{0.2412307} \\ \mathbf{-0.69652603} & \mathbf{0.39801488} & \mathbf{-0.59702231} \end{pmatrix}$$

Let's suppose that X is matrix $n \times n$ where rows are features and columns are training examples (this to be aligned with the formulas found during course). Otherwise, if X is matrix $n \times n$ where rows are training examples and columns are features, then we can simply report in the first scenario by considering X^T.

Moreover, let's suppose X is already centered (since in track tells that from X they obtained U and s it seems more likely that it was already centered; otherwise, you should simply subtract the mean of each feature).

Compute the **cumulative explained variance ratio**:

$$\frac{\sum_{j=1}^{k} s_j}{\sum_{j=1}^{n} s_j} \geq 0.60$$

It allows us to track the proportion of variance explained as we include more components. It is computed as the cumulative sum of the explained variance of a given principal component over total variance.

For $k = 1$:

$$\frac{11.2912878}{11.2912878 + 6.70871215 + 3.9171372510^{-16}} \approx 0.6273$$

Therefore, to retain at least 60%, we only need **Component 1**.

We first need to project the data onto the reduced space. Since we need to consider just the first principal component we have:

$$U_{reduced} = \begin{pmatrix} -0.34517975 \\ -0.62904883 \\ -0.69652603 \end{pmatrix}$$

Therefore, the projection Z is:

$$Z = U_{reduced}^T \cdot X = \begin{pmatrix} -0.34517975 & -0.62904883 & -0.69652603 \end{pmatrix} \begin{pmatrix} 1 & 5 & 3 \\ 3 & 4 & 8 \\ 5 & 1 & 6 \end{pmatrix} = \begin{pmatrix} -5.71 & -4.94 & -10.25 \end{pmatrix}$$

Finally, to reconstruct the original data from the reduced representation we can:

$$\tilde{X} = U_{reduced} \cdot Z = \begin{pmatrix} -0.34517975 \\ -0.62904883 \\ -0.69652603 \end{pmatrix} \begin{pmatrix} -5.71 & -4.94 & -10.25 \end{pmatrix} = \begin{pmatrix} 1.97 & 1.70 & 3.54 \\ 3.59 & 3.11 & 6.45 \\ 3.98 & 3.44 & 7.14 \end{pmatrix}$$