AII + ML Summer 2022

Q1 - Out of spec

Q2./a./ k-NN Pros

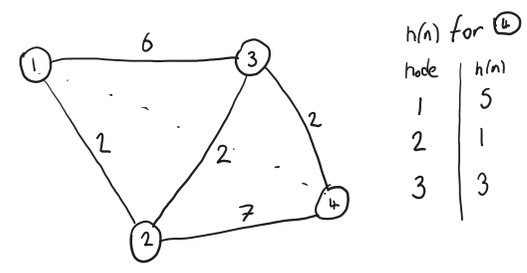
1./ K-NN is doesn't require the computation of training a model like LR. we only perform computation in inference. calculating the distance from each value to the new value and calculating an average - so training is faster than LR.

2./ K-NN may be less sensitive to outliers as only the closest k values are considered, whereas LR considers all values when training, so the weights may be skewed, affecting performance.

LR Pros

1./ LR doesn't require the entire dataset to be stored in memory, which can be prohibitive for k-NN on large datasets.

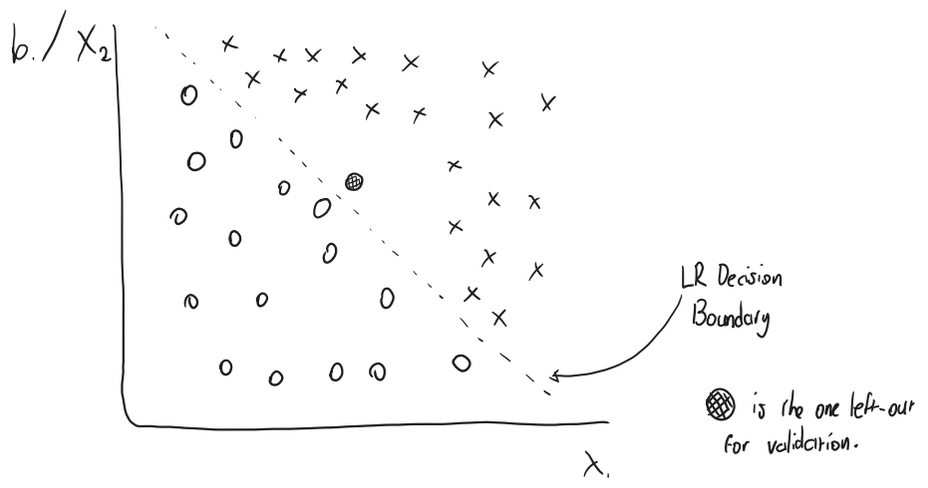
2./ Inference of the alogorithm is far quicker with LR, compared to K-NN. as most of the computation is done in training, whereas K-NN may take a long time for prediction if the dataset is large and high-dimensional.



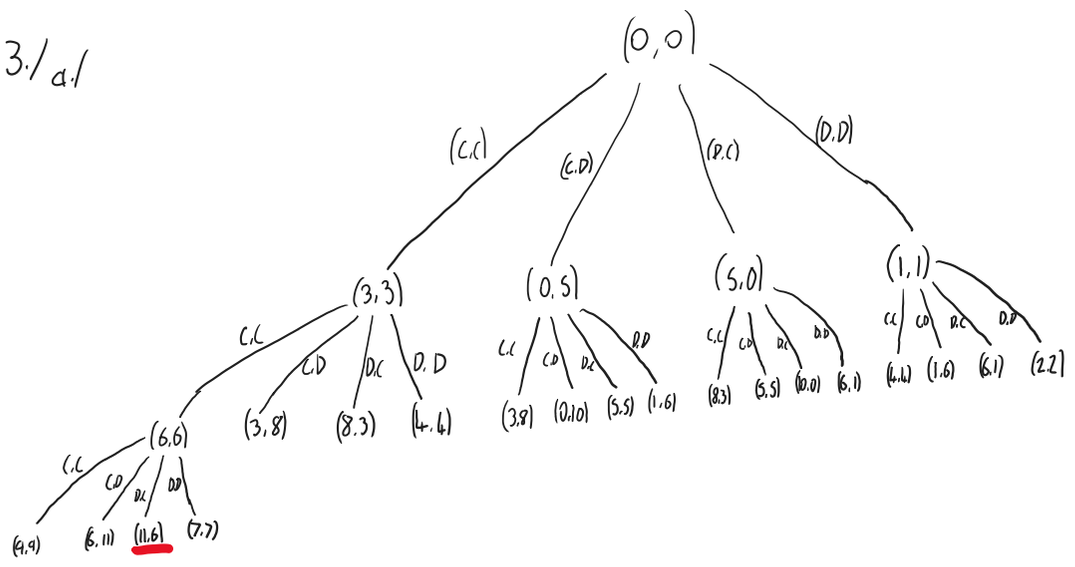
h(n) for (1)

Node	h(n)
1	5
2	1
3	3

b./



Q3./a./



Step	Node Visited/Expanded
1	0,0
2	3,3
3	0,5
4	5,0
5	1,1
6	6,6

b./ Solution: (C,C), (C,C), (D,C)

Cost is 3.

Q4./

Contractors (1, 2, ..., n) | Tasks (1, 2, ..., m)

i has C_i | j has h_j

X is a n-sized vector

X_i contains an integer from $0 \leq j \leq m$ (tasks)

i./ X is the design variable, it is n-sized vector, where n is number of contractors - so each position X_i refers to a specific contractor. At X_i, there is stored the task that the contractor is assigned to (0 if no task). This design variable is adequate to represent all the possible candidate solutions.

ii./ f(X) is the objective function. h_{x_i} represents the hours required for the task X_i, and C_i represents the hourly cost of the contractor i. $h_{x_i} \times C_i$ is summed across all of the i's from 1 up to n, which is the list of contractors. This is an adequate objective function, as it properly calculates the total cost of the candidate solution.

iii./ $g_i(x)$ is the constraints. The function $1(X_i=j)$ returns 1 if the contractor X_i is assigned to the task j for all j in $0 \leq j \leq m$. This is summed across all contractors, and made sure that the total is equal to 1. This enforces that each task is allocated to at most 1 contractor. The design variable enforces that each contractor can have at most one task by virtue of the fact we only store 1 integer value corresponding to a task. I would say that the notation is slightly ambiguous and should maybe be re-written as: $\sum_{j=0}^{m}\left(\left(\sum_{i=1}^{n} 1(X_i=j)\right)=1\right)$. But aside from this ambiguity, the constraints are adequate.