① a)

	a	b	c	d	e	f	nearest-neighbour L_1
a	0	1,5	1,5	4,5	7	6	$\Rightarrow$ b,c = 1,5
b	1,5	0	3	4	6,5	5,5	$\Rightarrow$ a = 1,5
c	1,5	3	0	3	5,5	4,5	$\Rightarrow$ a = 1,5
d	4,5	4	3	0	2,5	3,5	$\Rightarrow$ e = 2,5
e	7	6,5	5,5	2,5	0	1	$\Rightarrow$ f = 1
f	6	5,5	4,5	3,5	1	0	$\Rightarrow$ e = 1

$\sum_i |u_i - v_i|$

b)

	a	b	c	d	e	f	nearest neighbour L_2
a	0	$\sqrt{1,25}$	1,5	$\sqrt{10,25}$	$\sqrt{26,5}$	$\sqrt{22,5}$	$\Rightarrow b = \sqrt{1,25}$
b	$\sqrt{1,25}$	0	$\sqrt{5}$	$\sqrt{10}$	$\sqrt{21,25}$	$\sqrt{16,25}$	$\Rightarrow a = \sqrt{1,25}$
c	1,5	$\sqrt{5}$	0	$\sqrt{5}$	$\sqrt{21,25}$	$\sqrt{20,25}$	$\Rightarrow$ a = 1,5
d	$\sqrt{10,25}$	$\sqrt{10}$	$\sqrt{5}$	0	$\sqrt{6,25}$	$\sqrt{7,25}$	$\Rightarrow c = \sqrt{5}$
e	$\sqrt{26,5}$	$\sqrt{21,25}$	$\sqrt{21,25}$	$\sqrt{6,25}$	0	1	$\Rightarrow$ f = 1
f	$\sqrt{22,5}$	$\sqrt{20,25}$	$\sqrt{20,25}$	$\sqrt{7,25}$	1	0	$\Rightarrow$ e = 1

$\sqrt{\sum_i (u_i - v_i)^2}$

c) Since we use a different measure for distance in each classification instance, and we don't guarantee the same results (closest neighbour wise). It means that each classification instance is unique and the norm to be used should vary with the problem in hand.

② 3 classes
$$\begin{cases} a \to N_A = 16 \\ b \to N_B = 32 \\ c \to N_C = 64 \end{cases}$$

a) for any new point the classifier will predict c in these condition since if k = N you will get a majority rule and c is the most frequent class

b) with the weighted version we will predict the class in which the datapoints are most similar (distance wise) as long as they are close enough to outweigh the majority of c-class points.

(3) • The units in which the dimensions are represented are not really comparable so the distance measures wouldn't have much success. We could solve this by normalizing every dimension, since we would be comparing relatively similar values

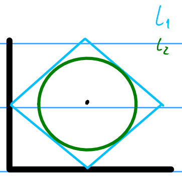
=> regarding decision trees, since we make comparisons for each dimension in isolation, we don't run into this problem

• Say we have atleast 100 data points for each class. This means that, assuming each data point is unique in every dimension. We cover roughly 500 points in a $\mathbb{R}^5$ space, with a total of 100^5 possible points. This means that the space covered equates to something as $500/100^5 = 5 \times 10^{-6}$ % of of the total sample space. This problem can't be fully solved but we can try to maximize the value of the available data by performing k-fold cross validation.

=> This is a common problem for most ML methods, including decision trees as we can't generate information if we don't have enough information to start.

(4)
$$\sum_{i=1}^{d} \sqrt{(x_i - y_i)^2} \leq \sum_{i=1}^{d} |x_i - y_i|$$

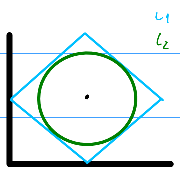
Let's consider a point p in $\mathbb{R}^2$ (generalizable to more dimensions)



if we consider an arbitrary distance d in which another point resides we can see that the L_1 norm forms a rhombus around p, as for L_2 we have a circular region around p.

visually we have that all points at distance d in L_2 of p are $\leq$ than all points of distance d of p in L_1. Proving that the original statement is correct

(5)



Should we take the point y in the center, and let's say we have a point x in the green region. Say that this point x is y's closest neighbor.

• by the proof above where $L_2 \leq L_1$, if we have a point in the green region this implies that there can't be another point z in the blue region where $d_{L_1} z \leq d_{L_2} x$. Which is equivalent to saying that if x is the point which is closest to y in L_2, the same applies when measuring distances in L_1.

(6) No since every split of continuous data can only be binary. the lines dividing the input space which are created by decision trees, can only be parallel to the vectors Ox and Oy which means that in order to create a perfect split of this data, we would have to create a sort of ladder with infinitesimal steps that approximate a line with slope 1

=> we need a DT with m (large enough) depth in order to classify this dataset with 100% accuracy. If the dataset fully occupied the space available then we would need m to be infinite.

(7) $I_H(t) = -\sum_{c \in C} \pi_c \log_2(c)$ | Note: $I(x,y) = x\log_2(x) + y\log_2(y)$

$$\Rightarrow I_H(y) = -\big(\underbrace{2/5 \log(2/5)}_{w} + \underbrace{3/5 \log(3/5)}_{l}\big)$$

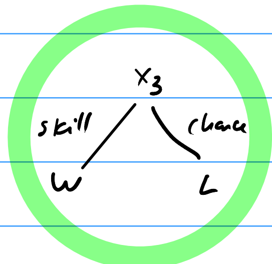
$$= (0{,}5288 + 0{,}4428)$$

$$= (0{,}971)$$

- $\Delta_H(x1) = I_H(y) - I_{team} - I_{ind} = 0$
  - $\Rightarrow I_{team} = 5/10 \; I(2/5, 3/5) \simeq 0{,}485$
  - $\Rightarrow I_{ind} = 5/10 \; I(3/5, 2/5) = 0{,}485$

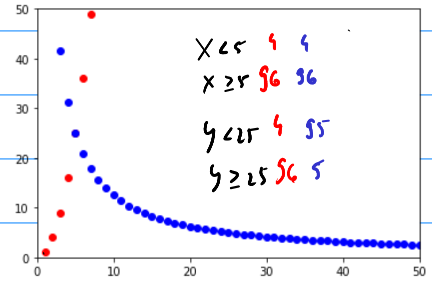
- $\Delta_H(x2) = I_H(y) - I_{metal} - I_{play} \simeq 0{,}021$
  - $\Rightarrow I_{metal} = 4/10 \; I(1/2, 1/2) = 0{,}4$
  - $\Rightarrow I_{play} = 6/10 \; I(1/3, 2/3) \simeq 0{,}55$

=> since a split over x_3 has the highest information gain, an optimal DT with depth 1 is:



- $\Delta_H(x_3) = I_H(y) - I_{skill} - I_{chance} \simeq 0{,}971 - 0{,}485 - 0{,}361 = 0{,}125$
  - $\Rightarrow I_{skill} = 5/10 \; I(3/5, 2/5) \simeq 0{,}485$
  - $\Rightarrow I_{chance} = 5/10 \; I(1/5, 4/5) \simeq 0{,}361$

8) plotting the data (limiting both coordinates to 50)



C1, C2 coincide when $x=5$ and $y=25$

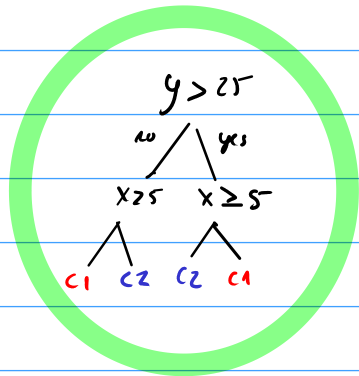
⇒ given the shape of the curve we plotted we can say that an optimum first split would have to occur over this point

⇒ choosing either splitting over $x<5$ or $y<25$

- the total entropy is $-\left(1/2 \log(1/2) + 1/2 \log(1/2)\right) = 1$
- $\Delta_H(x<5) = 1 - I_{x<5} - I_{x\geq 5} = 0$
  - $I_{x<5} = 8/200 \; I(4/8, 4/8) = 0{,}04$
  - $I_{x\geq 5} = 192/200 \; I(1/2, 1/2) = 0{,}96$
- $\Delta_H(y<25) = 1 - I_{y\geq 25} - I_{y<25} = 0{,}7356$
  - $I_{y<25} = 99/200 \; I(4/99, 95/99) \simeq 0{,}1209$
  - $I_{y\geq 25} = 101/200 \; I(96/101, 5/101) \simeq 0{,}1436$

⇒ this result was intuitive after looking at the plotted data

⇒ this means that we need to split over $x<5$ in the next leaf for both branches



⇒ after computing the optimal tree, we can see that 199/200 points are correctly classified. With the one error being the overlapping point of both classes. Since it shares the same coordinates, there is no possible decision tree that predicts two different classes for the same datapoint.

⇒ We can also see that there are equivalent trees, namely by changing the split values to the opposite region $(x\geq 5, y\geq 25)$ or, in hindsight, since we end up splitting over $x<5$ in both branches anyway it would result in a tree with the same accuracy and depth if we chose it as the first split.