ASSOCIATION RULES

Domanda 5

Risposta non ancora data

Punteggio max.: 6,00

Contrassegna domanda

Given the list of transactions reported below compute list the closed itemset using a support threshold of 0.4 (that is 40%). ⟹ min sup = 2

- a1,a2,a4,a5
- a2,a3,a4,a5
- a3,a5
- a2
- a2,a3,a4

List all the closed itemisets using the template included in the answer box. Each closed itemset must be reported on a separate line with its support.

Closed Itemset #1: (write the closed itemset as a1, a2, a3, ...)
Support #1:

Closed Itemset #2: (write the closed itemset as a1, a2, a3, ...)
Support #2:

Closed Itemset #3: (write the closed itemset as a1, a2, a3, ...)
Support #3:

Closed Itemset #3: (write the closed itemset as a1, a2, a3, ...)
Support #3:

Closed Itemset #4: (write the closed itemset as a1, a2, a3, ...)
Support #4:

Closed Itemset #5: (write the closed itemset as a1, a2, a3, ...)
Support #5:

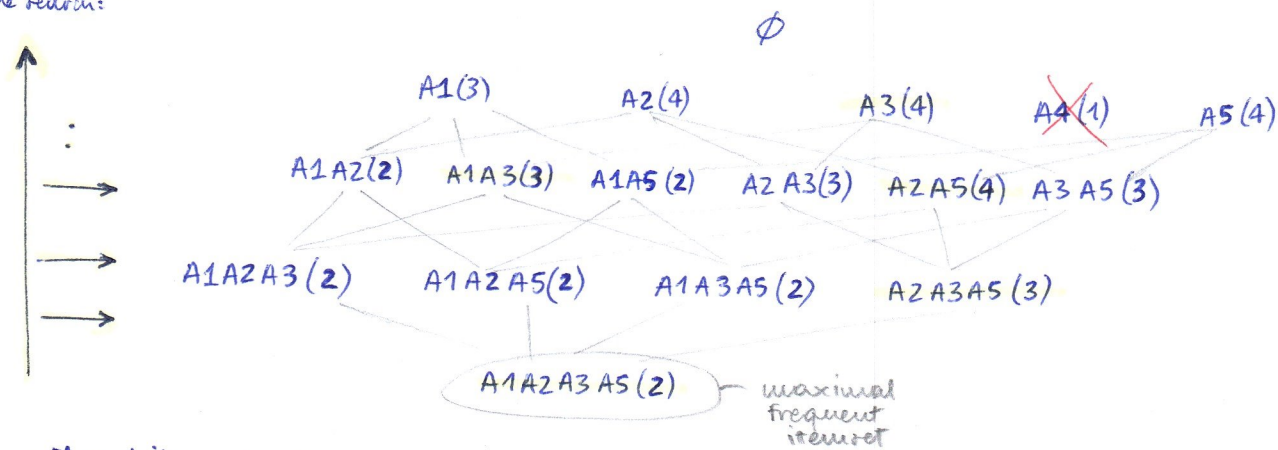
(insert additional entry if needed)

Version 2 :

1	A1 A3 A4
2	A2 A3 A5
3	A1 A2 A3 A5
4	A2 A5
5	A1 A2 A3 A5

min sup = 2

How we procede the search:

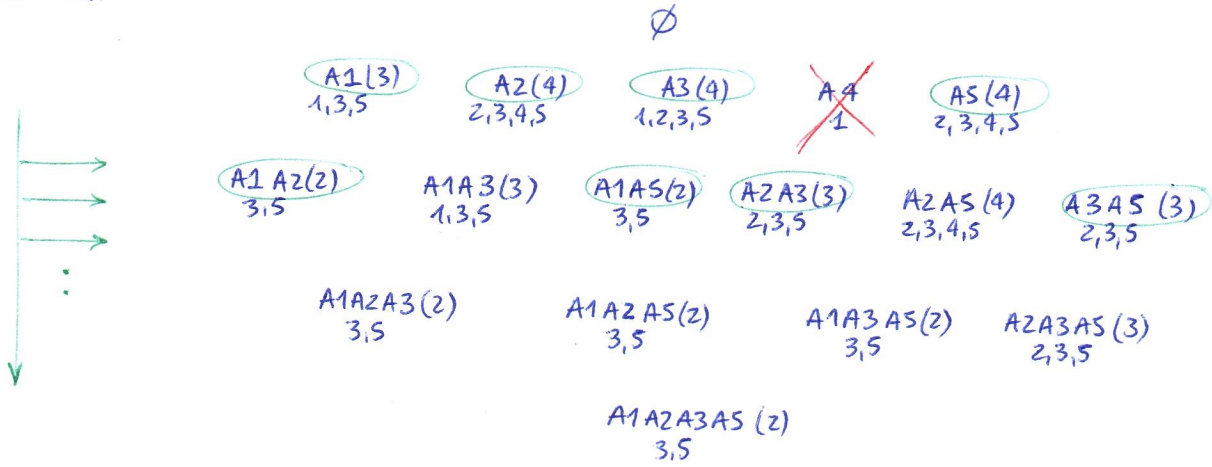


Closed itemset: itemset s.t. all its supersets have less support
We start from the bottom and we consider one level at the time.
⟹ Closed itemsets in yellow

What if we want to apply **Eclat**?
Translation of the database:

A1	A2	A3	A4	A5
1	2	1	1	2
3	3	2		3
5	4	3		4
	5	5		5

And then:

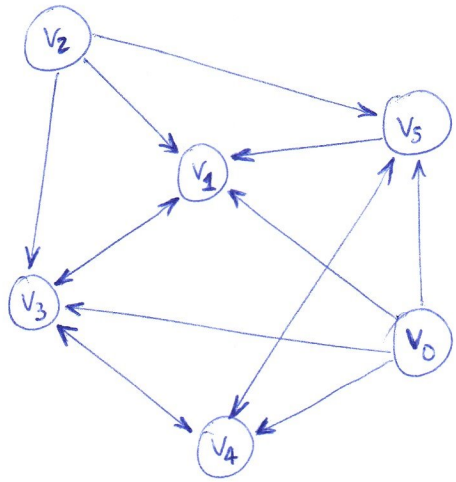


What if we look for **minimal generators**?
(itemsets that don't have subsets with the same support)

It's convenient to start from the top and consider one level at the time.

TRAWLING

Consider the graph below, identify the communities using a support treshold of 3.



$V_0 = \{V_1, V_3, V_4, V_5\}$
$V_1 = \{V_3\}$
$V_2 = \{V_1, V_3, V_5\}$
$V_3 = \{V_1, V_4\}$
$V_4 = \{V_3, V_5\}$
$V_5 = \{V_1, V_4\}$

Frequent itemsets (a priori):

$\emptyset$

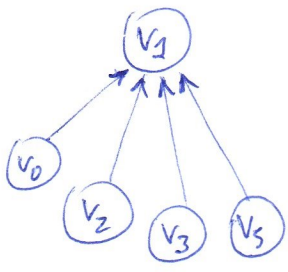
$V_2(4)$ ~~$V_2(0)$~~ $V_3(4)$ $V_4(3)$ $V_5(3)$

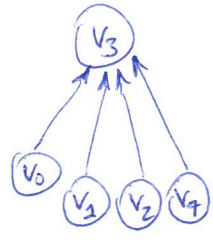
~~$V_2V_3(2)$~~ $V_1V_4(3)$ ~~$V_2V_5(2)$~~ ~~$V_3V_4(1)$~~ $V_3V_5(3)$ ~~$V_4V_5(1)$~~

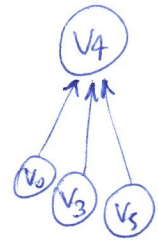
$\Longrightarrow$

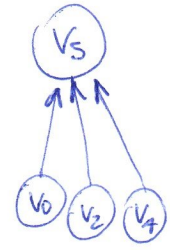
V_1	$s=4$	V_1V_4	$s=3$
V_3	$s=4$	V_3V_5	$s=3$
V_4	$s=3$		
V_5	$s=3$		

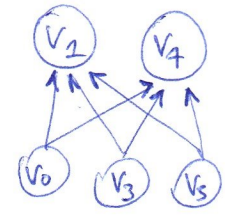
Communities:

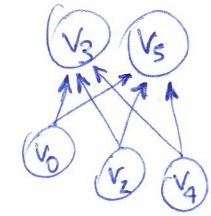












24/06/2020 – CLUSTERING

Consider the 5 datapoints p0, .., p4 and the following distance matrix:

	p0	p1	p2	p3	p4
p0	0.0	4	10	4	1
p1	4	0.0	6	2	5
p2	10	6	0.0	6	11
p3	4	2	6	0.0	5
p4	1	5	11	5	0.0

Apply the first step of hierarchical clustering using single link (MIN) approach to measure distance between clusters.

1. Report the points involved in the first merge and the corresponding distances
2. Report the resulting matrix showing one row after another

Let's do it completely!

	p0	p1	p2	p3	p4
p0	–				
p1	4	–			
p2	10	6	–		
p3	4	2	6	–	
p4	1	5	11	5	–

Remember: the matrix distance decides which elements are going to merge, the linkage decides how we'll update the matrix distance

Note: to determine the first merge (1 merge) we consider the points with smaller distance (1 linkage)

first merge: p0 and p4 ($d=1$)

we use: single linkage

	p0,p4	p1	p2	p3
p0,p4	–			
p1	4	–		
p2	10	6	–	
p3	4	2	6	–

$d(\{p0,p4\}, p1) = \min\{d(p0,p1), d(p4,p1)\}$
$= \min\{4,5\} = 4$

$d(\{p0,p4\}, p2) = \min\{10,11\} = 10$

$d(\{p0,p4\}, p3) = \min\{4,5\} = 4$

Second merge: p1 and p3 ($d=2$)

	p0,p4	p1,p3	p2
p0,p4	–		
p1,p3	4	–	
p2	10	6	–

$d(\{p1,p3\}, p2) = \min\{6,6\} = 6$

$d(\{p0,p4\}, \{p1,p3\}) =$

$= \min\{d(\{p0,p4\},p1), d(\{p0,p4\},p3)\}$

$= \min\{4,4\} = 4$

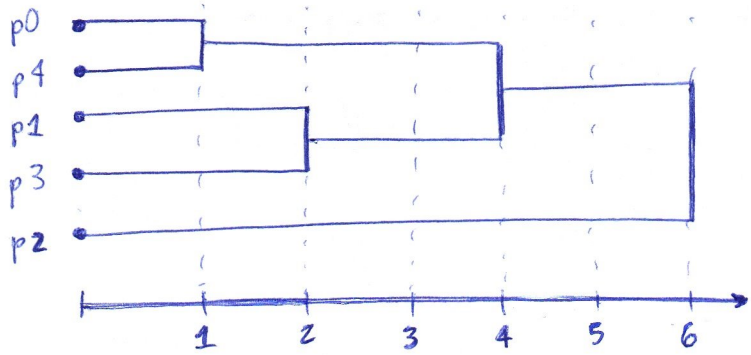
Third merge: $\{p0,p4\}$ and $\{p1,p3\}$ ($d=4$)

	p0,p1,p3,p4	p2
p0,p1,p3,p4	–	
p2	6	–

$d(\{..\}, p2) = \min\{10,6\} = 6$

The final merge is at distance 6.

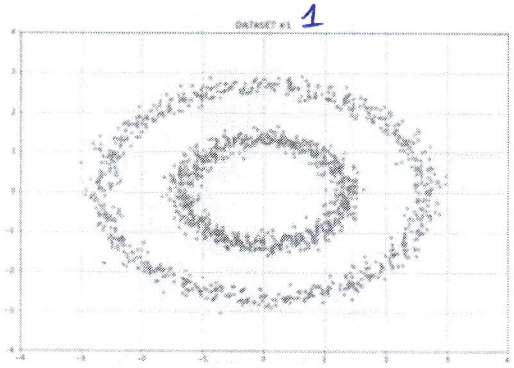
Dendrogram



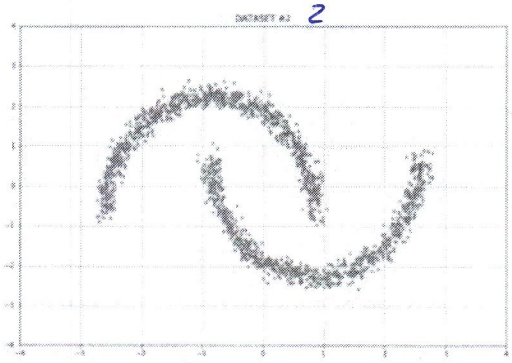
CLUSTERING

14/07/2020 (pt. 1) – Single linkage

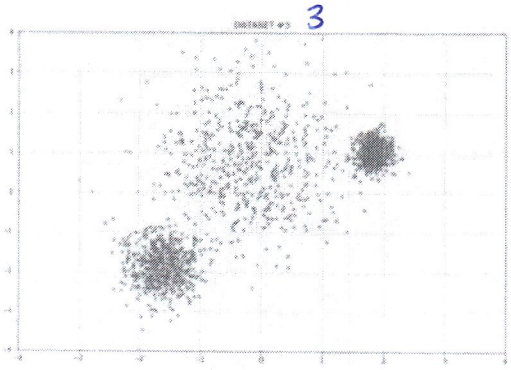
1



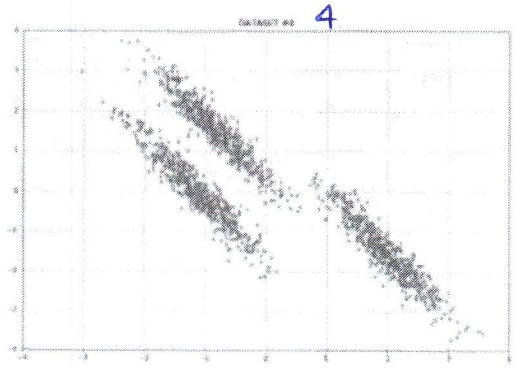
2



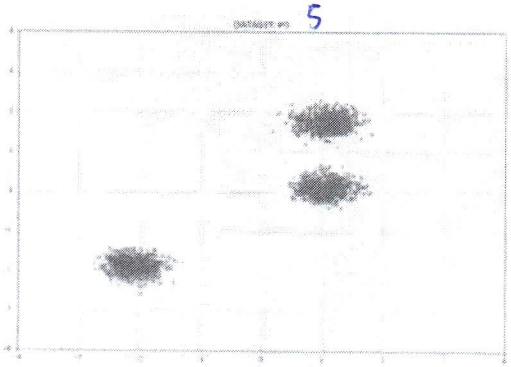
3



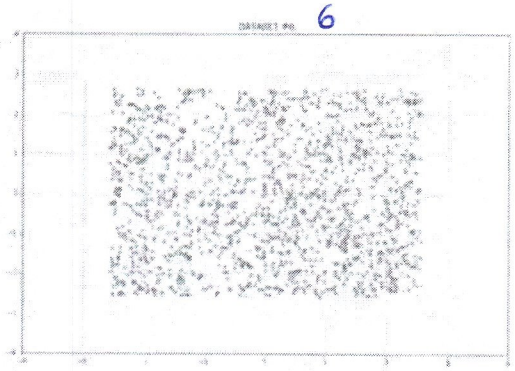
4



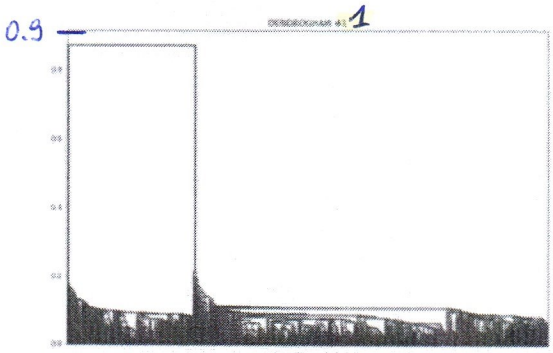
5

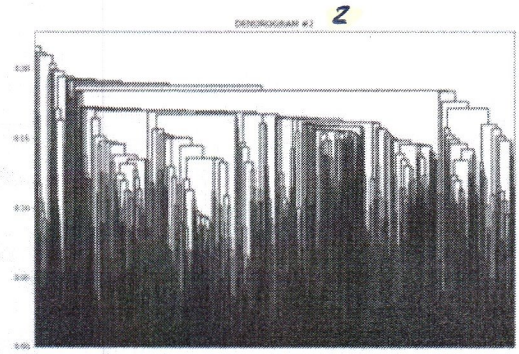


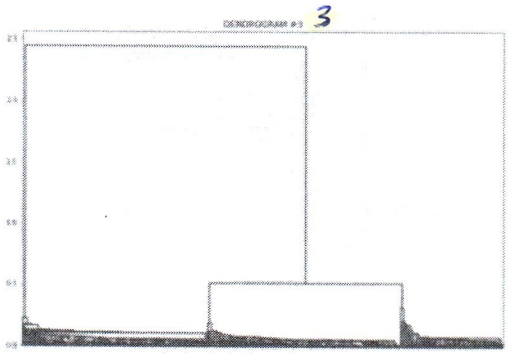
6

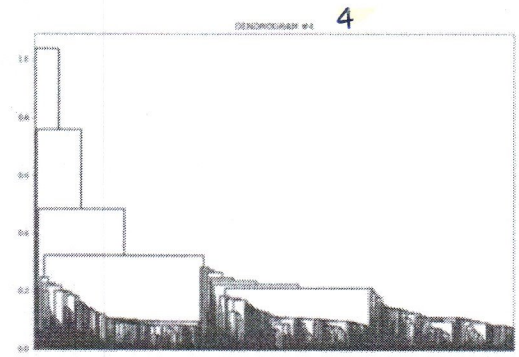


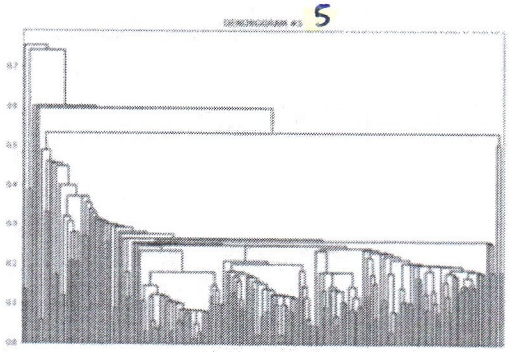
CLUSTERING
14/07/2021 (pt. 2)

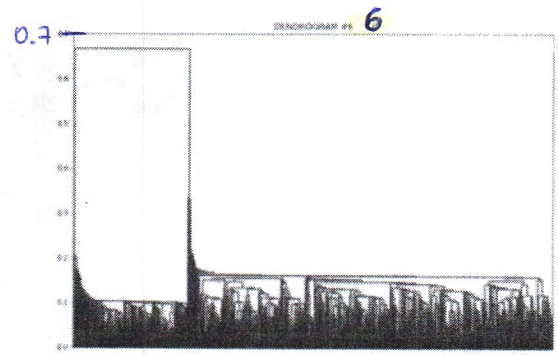








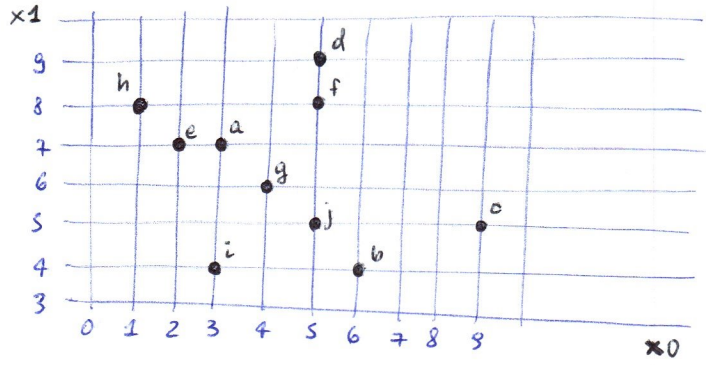




Dendrogram 3 comes from dataset 5
2 6
5 3
4 4
! 6 1
! 1 2 — because data are more dense (compact) (the points are more closer among eachother than in the dataset 1)

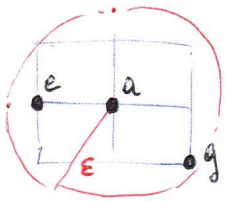
REPRESENTATIVE-BASED CLUSTERING

Given the dataset reported below, apply DBSCAN with $\varepsilon = 1.5$ and minpts = 3 and list all the core, border and noise points.



Core points = points that have at least minpoints in the way of ε
The point itself IS PART OF THE COUNT!

a)



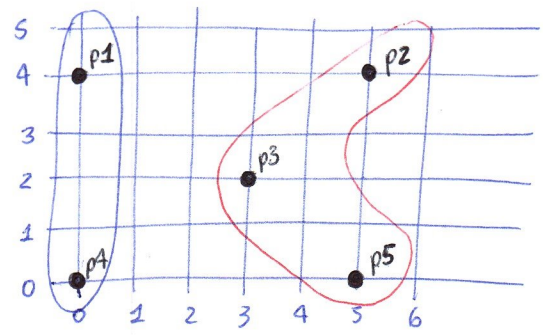
⟹ 3 points ⟹ CORE

b) 2 points ⟹ BORDER (j) CORE)
c) 1 point ⟹ NOISE
d) 2 points ⟹ NOISE
e) 3 points ⟹ CORE
f) 2 points ⟹ NOISE
j) 3 points ⟹ CORE
h) 2 points ⟹ BORDER
g) 3 points ⟹ CORE
i) 1 point ⟹ NOISE

(Taken from 5/02/2020) - REPRESENTATIVE BASED CLUSTERING

Consider the following dataset containing 5 points $(p1, \dots, p5)$ in which columns **x** and **y** represent the input variables while the column "label" contains the cluster label that has been assigned to each example by applying **k-means** with **k=2** and Euclidean distance. Compute for each example the **silhouette coefficient** using Euclidean distance.

	x	y	label
p1	0	4	0
p2	5	4	1
p3	3	2	1
p4	0	0	0
p5	5	0	1



For every point we need a, b, silhouette:

a = average of the distances with points inside the cluster

b = average of the distances with points of the closest cluster

$$s_i = \frac{b-a}{\max\{b,a\}}$$

	a	b	s_i
p1	4	5	0.2
p2	3.4	5.7	0.4
p3	2.83	3.6	0.21
p4	4	5	0.2
p5	3.4	5.7	0.4

$S_{TOT} = \frac{1}{n}\sum_{i=1}^{n} s_i$

$= 0,282$

p1) $a = 4 = d(p1, p4)$

p2) $a = (d(p2,p3) + d(p2,p5))/2 = (2\sqrt{2} + 4)/2 = \sqrt{2} + 2 = 3.4142 = 3.4$

p3) $a = (d(p3,p2) + d(p3,p5))/2 = (2\sqrt{2} + 2\sqrt{2})/2 = 2\sqrt{2} = 2.8284 = 2.83$

p4) = p1)

p5) = p2)

p1) $b = (d(p1,p2) + d(p1,p3) + d(p1,p5))/3 = (5 + \sqrt{13} + \sqrt{41})/3 = (5 + 3.6 + 6.4)/3 = 5$

p2) $b = (d(p1,p2) + d(p2,p4))/2 = (5 + \sqrt{41})/2 = (5 + 6.4)/2 = 5.7$

p3) $b = (d(p1,p3) + d(p3,p4))/2 = \sqrt{13} = 3.6$

p4) = p1)

p5) = p2)

p1) $(5-4)/5 = s_1 = 1/5 = 0.2$

p2) $s_2 = (5.7 - 3.4)/5.7 = 0.4$

p3) $s_3 = (3.6 - 2.83)/3.6 = 0.21$

p4) = p1)

p5) = p2)

July 2020 — NAIVE BAYES & BAYESIAN NETWORKS

Consider the dataset below where attribute "class" identify the class attribute. Compute the probability for the class and for the attribute "meat" that are required to compute a Naive Bayes classifier using the Laplace Estimator and report the results in the corresponding boxes below. "cooking" and "side" have been marked using "-" since they're useless for answering the questions below.

meat	cooking	side	class
chicken	fried	fried	dislike
lamb	roasted	veggie	like
veal	-	-	dislike
chicken	-	baked	like
chicken	-	-	like
veal	-	-	dislike
lamb	-	-	dislike
veal	-	-	dislike
veal	-	-	like
pork	-	-	like
veal	-	-	dislike
lamb	-	-	dislike

Part 1.

1. IP(class=dislike) = 7/12 = 0.58
2. IP(class=like) = 5/12 = 0,42
3. IP(meat=chicken | class=dislike) = 0.18
4. IP(lamb | dislike) = 0,27
5. IP(pork | dislike) = 0,09
6. IP(veal | dislike) = 0,45
7. IP(chicken | like)
8. IP(lamb | like)
9. IP(pork | like)
10. IP(veal | like)

Meat	like	dislike
chicken	2 +1	1 +1
lamb	1 +1	2 +1
veal	1 +1	4 +1
pork	1 +1	0 +1
	9	11

Laplace Estimator

Probabilities		
Meat	like	dislike
chicken	3/9	2/11
lamb	2/9	3/11
veal	2/9	5/11
pork	2/9	1/11

Part 2.

Consider the values of the other probabilities required by the Naive Bayes classifier listed below:

- IP(cooking = fried | dislike) = 0.56
- IP(roasted | dislike) = 0.44
- IP(fried | like) = 0.43
- IP(roasted | like) = 0.57
- IP(side = baked | dislike) = 0.50
- IP(fried | dislike) = 0.30
- IP(veggie | dislike) = 0.20
- IP(baked | like) = 0.38
- IP(fried | like) = 0.25
- IP(veggie | like) = 0.38

Compute the probability for Naive Bayes classifier to classify the following 3 examples in the dataset as dislike or like:

1. IP(dislike | pork, roasted, fried)
2. IP(like | pork, roasted, fried)
3. IP (dislike | veal, fried, baked)
4. IP (like | veal, fried, baked)
5. IP (dislike | lamb, fried, baked)
6. IP (like | lamb, fried, baked)

1. ℓ(dislike | pork, roasted, fried) = IP(pork | dislike) IP(roasted | dislike) IP(fried | dislike) IP(dislike)
   $\propto$ (1/11) · (0.44) · (0.30) · (0.58) = 0,01
   this still need to be normalized to be a TRUE probability !

2. ℓ(like | pork, roasted, fried) = .. = "val"

$\Longrightarrow$ the results of the questions must be normalized! "val" + 0,01 $\neq$ 1, they have to!
$\Longrightarrow$ normalization!

June 2020 — DECISION TREE

Given the following dataset in which "**class**" represents the class attribute, compute the information gain for attribute "**meat**".

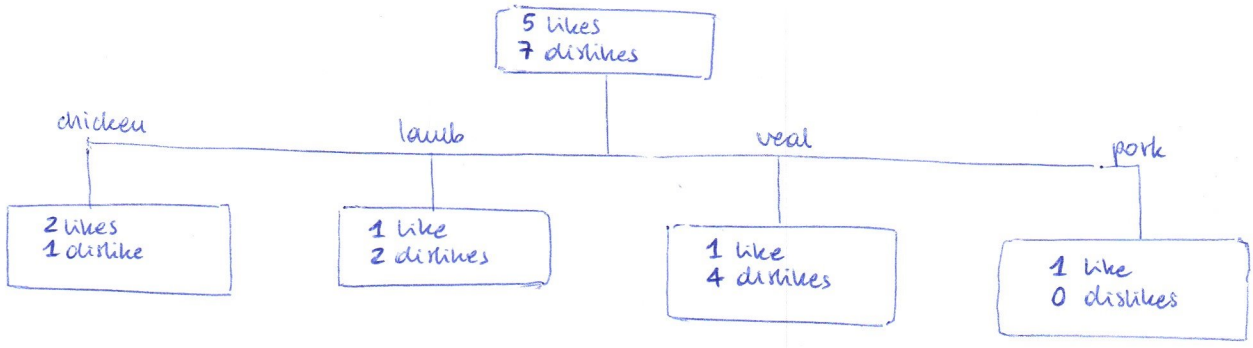
NOTE: the logarithm computation must be performed using base 2. The result must be reported using at least a two digit precision.

	meat	cooking	side	spicyness	class
0	chicken	fried	fried	1.0	dislike
1	lamb	roasted	veggie	1.0	like
2	veal	roasted	veggie	1.0	dislike
3	chicken	fried	baked	1.0	like
4	chicken	fried	baked	2.0	like
5	veal	roasted	fried	2.0	dislike
6	lamb	fried	baked	2.0	dislike
7	veal	roasted	baked	2.0	dislike
8	veal	roasted	veggie	3.0	like
9	pork	roasted	fried	3.0	like
10	veal	fried	baked	4.0	dislike
11	lamb	fried	baked	4.0	dislike

Entropy Before Splitting = 0.98

Entropy After Splitting = 0.76

Information Gain = 0.22



Root:

$info([5,7]) = entropy([5/12, 7/12]) = -5/12 \log_2(5/12) - 7/12 \log_2(7/12)$

$= -0.417 \cdot (-1.263) - 0.583 \cdot (-0.778) = 0.98$

chicken:

$info([2,1]) = entropy([2/3, 1/3]) = -2/3 \log_2(2/3) - 1/3 \log_2(1/3) = 0.918$

lamb:

$info([1,2]) = info([2,1]) = 0.918$

Pork:

$info([1,0]) = 0$ (0 confusion (entropy) when we have 100% of one class)

Veal:

$info([1,4]) = entropy([1/5, 4/5]) = -\frac{1}{5} \log_2(\frac{1}{5}) - \frac{4}{5} \log_2(\frac{4}{5}) = 0.722$

The overall entropy after the split:

$$= \frac{3}{12} \cdot 0.918 + \frac{3}{12} \cdot 0.918 + \frac{5}{12} \cdot 0.722 + \frac{1}{12} \cdot 0$$

$$= 0.76 = \text{info}(\text{meat})$$

$$\text{gain}(\text{meat}) = \text{info}(D) - \text{info}(\text{meat}) = 0.98 - 0.76 = 0.22$$

July 2020 — DECISION TREE

Given the following dataset in which "**class**" represents the class attribute, compute the Information Gain for the numerical attribute "**spicyness**".

NOTE:

- **The split value is computed as the average of the nearby values for example, the split between 0.7 and 0.8 will be 0.75.**
- **Entropy must be computed using the logarithm with base 2.**
- **The result must be reported using at least a two digit precision.**

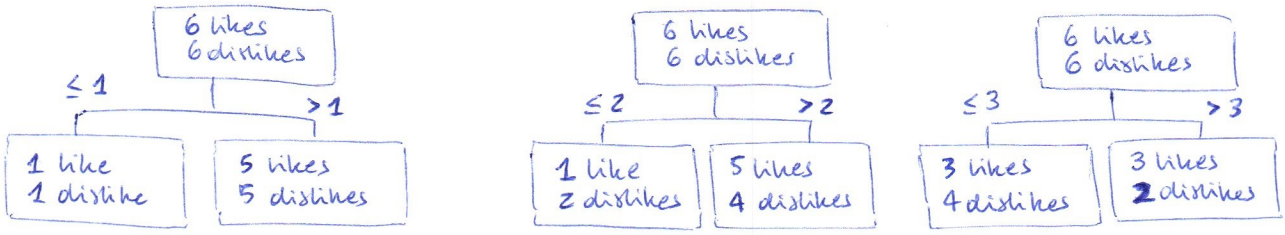
meat	cooking	side	spicyness	class
pork	roasted	fried	1.0	dislike
veal	roasted	veggie	1.0	like
chicken	fried	fried	2.0	dislike
pork	fried	baked	3.0	like
chicken	fried	fried	3.0	dislike
veal	fried	baked	3.0	dislike
pork	roasted	baked	3.0	like
chicken	fried	baked	4.0	dislike
chicken	roasted	fried	4.0	like
chicken	roasted	fried	4.0	like
pork	fried	veggie	4.0	like
veal	fried	baked	4.0	dislike

1. Entropy Before Splitting = 1
2. Splitting Threshold =
3. Information Gain =

Possible splits:

1.0	1.0	2.0	3.0	3.0	3.0	3.0	4.0	4.0	4.0	4.0	4.0
dislike	like	dislike	like	dislike	dislike	like	dislike	like	like	like	dislike

Entropy(D) = info(D) = info([6,6]) = entropy($[\frac{1}{2}, \frac{1}{2}]$) = 1 maximal confusion (50%, 50%)



Split 1:

(≤ 1) we got 50%, 50%, maximum confusion ⇒ entropy, info = 1

(> 1) entropy = 1

info(split 1) = $\frac{2}{12} \cdot 1 + \frac{10}{12} \cdot 1 = 1$ ⟹ **no gain**

Split 2:

info(≤ 2) = info([1,2]) = entropy($[\frac{1}{3}, \frac{2}{3}]$) = $-\frac{1}{3}\log_2(\frac{1}{3}) - \frac{2}{3}\log_2(\frac{2}{3}) = 0.918$

info(> 2) = info([5,4]) = entropy($[\frac{5}{9}, \frac{4}{9}]$) = $-\frac{5}{9}\log_2(\frac{5}{9}) - \frac{4}{9}\log_2(\frac{4}{9}) = 0.991$

info(split 2) = $\frac{3}{12} \cdot 0.918 + \frac{9}{12} \cdot 0.991 = 0.973$

gain(spiciness, ≤ 2, > 2) = info(D) − info(spiciness, ≤ 2, > 2) = 1 − 0.973 = 0.027

Split 3 :

info (≤3) = info([3,4]) = entropy([3/7, 4/7]) = [..]

info (>3) = info([3,2]) = entropy([

info (split 3) = [..]

gain (spiciness, ≤3, >3) = [..]