Homework 1

Miriam Wagner
373045

16. Mai 2018

Question 1

I used for solving this exercise Matlab. For the distance I used euclidian distance. My clustering algorithm calculates for every Datavector the distances to every centroid and then checks which centroid is closed. The vector is add to the closest cluster. When all datavectors are put in a cluster the new cluster centroids are calculated.
My abort criterion is the change between the centroids. If they still change I will apply the algorithms again. Because of the computer accuracy I do not use 0, but 10^{-15}. I also let the algorithm stop after 3 rounds, but it is already good enough after 2 rounds.
After the first round all data is clustered in cluster 2 and 3. I decided to let be the first centroid $(-1000, -1000, -1000)$. Than it does not influence the result. The distances in the first round are to see in 1

Tabelle 1: Distances first round

distance_to_cluster_1	distance_to_cluster_2	distance_to_cluster_3
43000,00002	3000,002694	23000,00899
35000,07642	5000,486525	15000,09671
26888,00021	13112,00189	6888,039561
44000,00413	4000,028178	24000,00035
40000,00664	20,15113235	20000,002
47500,00003	7500,000631	27500,00685
31000,02992	9000,088284	11000,02773
45000,0265	5000,212648	25000,02102
37000,00023	3000,000006	17000,00789
15000,23523	25000,12893	5000,409169
45000,00443	5000,027278	25000,00028
42500,00031	2500,000209	22500,00527
44000,00319	4000,02221	24000,00046
43500,00003	3500,002355	23500,00885
25000,03222	15000,04597	5000,051895
40000,00014	7,034833476	20000,01362
42200,02875	2200,477885	22200,02037
41500,00014	1500,016496	21500,01267
33000,00058	7000,000319	13000,00815
47000,00118	7000,004581	27000,0024

Tabelle 2: centroid

centroid_1	centroid_2	centroid_3
-10000	7000	23112
-10000	0,29309	0,000532
-10000	6	3

The new centroids are in 2

Tabelle 3: Distances second round

distance_to_cluster_1	distance_to_cluster_2	distance_to_cluster_3
22116,19085	0	16112,00028
28758,56529	8000,338507	8112,360556
36006,45169	16112,00028	0
21365,89781	1000,180514	17112,01414
24510,81085	3000,093366	13112,02657
18878,30691	4500,000119	20612,00039
32284,64122	12000,0792	4112,262067
20652,19999	2000,613442	18112,0751
27003,89625	6000,001337	10112,00244
47192,82789	28000,12683	11888,32027
20631,35796	2000,10364	18112,01496
22505,10872	500,0250381	15612,00206
21367,51278	1000,147276	17112,01178
21734,0764	500,0000847	16612,00027
37765,65939	18000,04611	1888,500752
24496,12265	3000,001514	13112
22765,44019	801,5423085	15312,09033
23287,55072	1500,003029	14612
30483,445	10000,0018	6112,006626
19220,80594	4000,015673	20112,00478

And the centroids in 4

Tabelle 4: centroids second round and third round

centroid_1	centroid_2	centroid_3
-10000	7000	23112
-10000	0,29309	0,000532
-10000	6	3

Tabelle 5: Cluster 1

amount_req	case_duration	total_activities

Cluster1 contains 5.
Cluster 2 contains as in 6
Cluster 3 in 7

Tabelle 6: Cluster 2

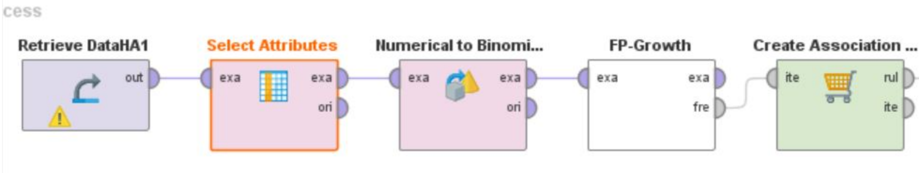
amount_req	case_duration	total_activities
7000	0,29309	6
15000	28,43964	74
6000	0,048206	25
10000	12,95023	26
2500	0,021134	7
5000	29,51885	46
13000	0,515625	10
5000	7,612419	25
7500	0,489861	11
6000	6,503808	22
6500	0,002049	6
10000	0,000799	3
7800	19,1099	52
8500	0,000486	3
3000	6,955382	15

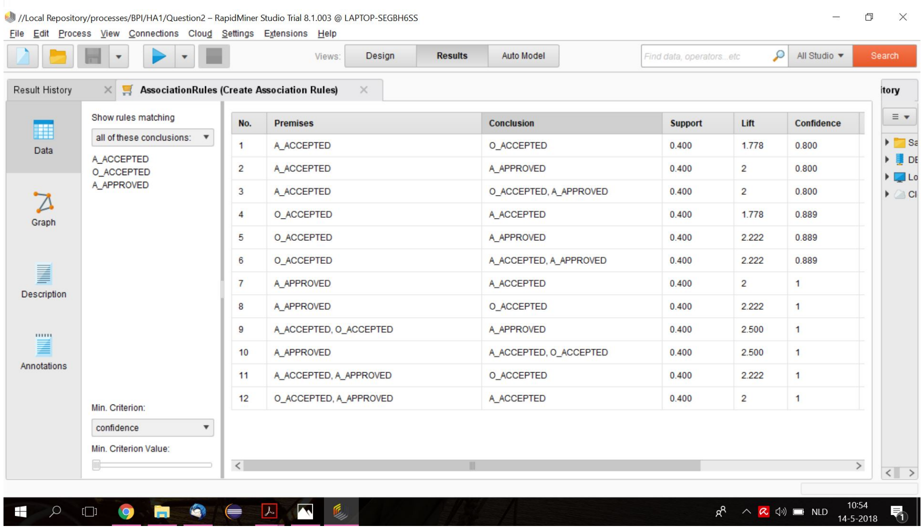
Tabelle 7: Cluster 3

amount_req	case_duration	total_activities
23112	0,000532	3
19000	19,78213	45
35000	19,74352	88
25000	21,14506	41
17000	0,01375	12

Question2

The process is built up like in the picture. First I selected the attributes I am assumed to consider (**Select Attributes**). Then the numerical values are translated to binomial values, because this is needed for the association rules. (**Numerical to Binomial**) RapidMiner expectes a FrequentItemSet for the **Create Association Rule**, so before I could use that I also had to use **FP-growth**.

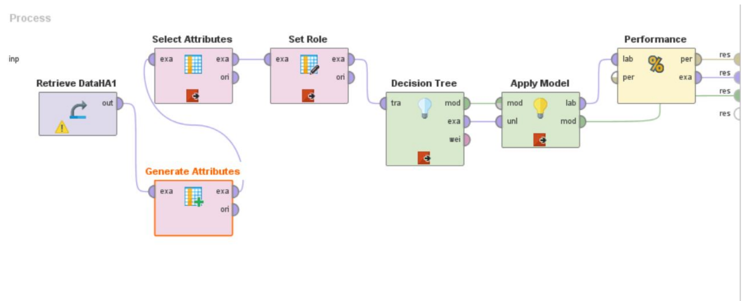




No.	Premises	Conclusion	Support	Lift	Confidence
1	A_ACCEPTED	O_ACCEPTED	0.400	1.778	0.800
2	A_ACCEPTED	A_APPROVED	0.400	2	0.800
3	A_ACCEPTED	O_ACCEPTED, A_APPROVED	0.400	2	0.800
4	O_ACCEPTED	A_ACCEPTED	0.400	1.778	0.889
5	O_ACCEPTED	A_APPROVED	0.400	2.222	0.889
6	O_ACCEPTED	A_ACCEPTED, A_APPROVED	0.400	2.222	0.889
7	A_APPROVED	A_ACCEPTED	0.400	2	1
8	A_APPROVED	O_ACCEPTED	0.400	2.222	1
9	A_ACCEPTED, O_ACCEPTED	A_APPROVED	0.400	2.500	1
10	A_APPROVED	A_ACCEPTED, O_ACCEPTED	0.400	2.500	1
11	A_ACCEPTED, A_APPROVED	O_ACCEPTED	0.400	2.222	1
12	O_ACCEPTED, A_APPROVED	A_ACCEPTED	0.400	2	1

I would pick the rule $\{A\_ACCEPTED, O\_ACCEPTED\} \Rightarrow \{A\_APPROVED\}$ and $\{A\_APPROVED\} \Rightarrow \{A\_ACCEPTED, O\_ACCEPTED\}$, because they have the highest lift, confidence and support. When you have a closer look you will see that the sets are probably of the rules have a back- and forth relationsip. Such that you can summarize it in one rule. Then you could look for the next best rules, where lift, support and confidence is the highest.

Question 3



The process
first changes the numerical attribute TotalActivites to a nominal one by **Generate Attributes** and **Select Attributes**. The generating consideres the old TotalActivites and contains the rule that all data, where TotalActivites is lower or the same as 40 it should be assigned to "Low" otherwise "High". The **Set Role** gives the new attribute as label, so RapidMiner knows what should have be the outcome of the Decision Tree. **Decision Tree** generates the decision Tree. Then **Apply Model** for **Performance** checking. The output is then the model and the performance of the model on the data.
The found decision tree is

If you check the Confusionmatrix

accuracy: 100.00%

	true Low	true High	class precision
pred. Low	14	0	100.00%
pred. High	0	6	100.00%
class recall	100.00%	100.00%	

you see, that this decision tree classifys the data perfectly. So you can predict by just knowing the case duration the total activities. If the case duration is higher, than also the total activities are high. This seems to be logical, if you have to do a lot this takes most of the times longer and otherwise around, if you do not need long you mostly did not do a lot of different things in the time.

Question 4

1.

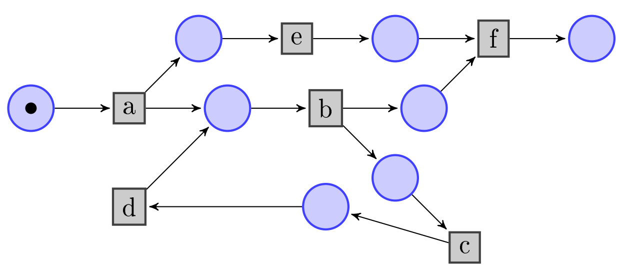
$L1 = [\langle a,b,e,f\rangle, \langle a,b,e,c,d,b,f\rangle, \langle a,b,c,e,d,b,f\rangle, \langle a,b,c,d,e,b,f\rangle, \langle a,e,b,c,d,b,f\rangle]$

The $\alpha-$Algorithm gives the following:

$$T_L = \{a,b,c,d,e,f\}$$
$$T_I = \{a\}$$
$$T_O = \{f\}$$

	a	b	c	d	e	f
a	#	→	#	#	→	#
b	←	#	→	←	||	→
c	#	←	#	→	||	#
d	#	→	←	#	||	#
e	←	||	||	||	#	→
f	#	←	#	#	←	#

$X_L = \{(\{a\},\{b\}), (\{a\},\{c\}), (\{a\},\{e\}), (\{b\},\{d\}), (\{c\},\{d\}), (\{e\},\{f\}), (\{a,e\},\{f\}), (\{f\},\{d\}), (\{a,e\},\{f,d\}), (\{a,$
$Y_L = \{(\{a\},\{b\}), (\{a\},\{c\}), (\{b\},\{d\}), (\{c\},\{d\}), (\{a,e,h,f,g\},\{f\}), (\{a,d\},\{b\})\}$



2.

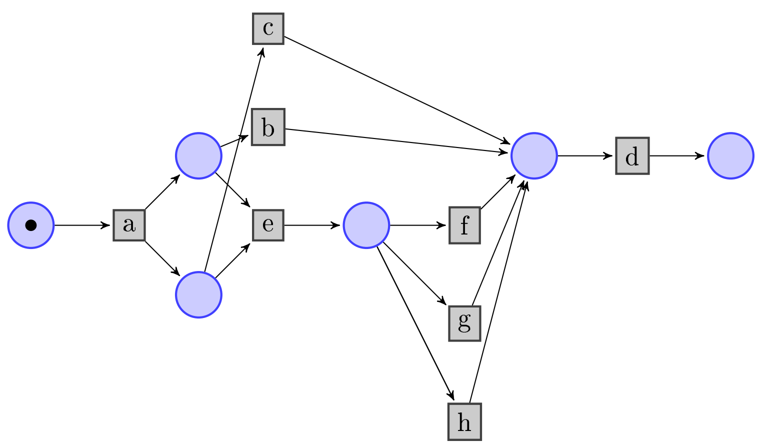
$$L2 = [\langle a,b,c,d\rangle, \langle a,c,b,d\rangle, \langle a,e,f,d\rangle, \langle a,e,g,d\rangle, \langle a,e,h,d\rangle]$$

The $\alpha-$Algorithm gives the following:

$$T_L = \{a,b,c,d,e,f,g,h\}$$
$$T_I = \{a\}$$
$$T_O = \{d\}$$

	a	b	c	d	e	f	g	h
a	#	→	→	#	→	#	#	#
b	←	#	||	→	#	#	#	#
c	←	||	#	→	#	#	#	#
d	#	←	←	#	#	←	←	←
e	←	#	#	#	#	→	→	→
f	#	#	#	→	←	#	#	#
g	#	#	#	→	←	#	#	#
h	#	#	#	→	←	#	#	#

$$
\begin{aligned}
X_L = \{&(\{a\},\{b\}),(\{a\},\{c\}),(\{a\},\{e\}),(\{a\},\{b,e\}),(\{a\},\{c,e\}),(\{b\},\{c\}),(\{b\},d),(\{c\},\{b\}),\\
&(\{c\},\{d\}),(\{e\},\{f\}),(\{e\},\{g\}),(\{e\},\{h\}),(\{g\},\{d\}),(\{h\},\{d\}),(\{f\},\{d\}),\\
&(\{g,h\},\{d\}),(\{g,f\},\{d\}),(\{h,f\},\{d\}),(\{g,f,h\},\{d\})\}\\
Y_L = \{&(\{a\},\{b,e\}),(\{a\},\{c,e\}),(\{g,f,h\},\{d\})\}
\end{aligned}
$$

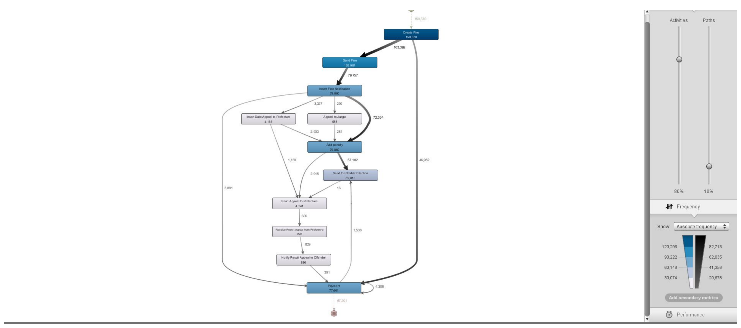


Question 5

a)

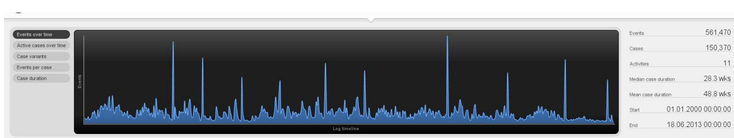
The process has 15370 cases and 561470 events. Median number of events is 5 (and average number The average duration is 48.8 weeks.

b)



The loop tells us, that a part of all people had to pay at least two times. If you check the log data, you can see, that mostly it happens in the next 30 days.
It happen 4306 and for 4014 cases. So there are cases, where it happens more than one time. You also can see, that it happens at most 14 times.

c)



You can see that the distribution is going up and down a little bit, but there are 10 laces to see where more happens. Further in the end activity gets lower and the distance between the 6th and 7th lace is higher than between the others. It is always around the typical paydays, so probably a lot of people then have the money to pay the fine.

d)

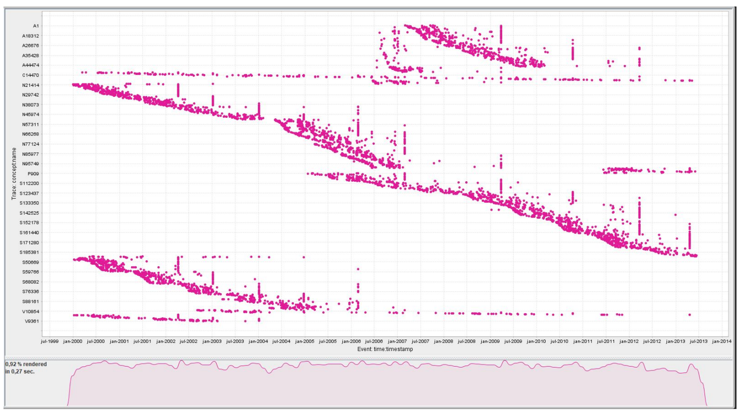
There are 231 variants. The third most frequent variant has 20385 instances. It just contains the behaviour create and send fine. Nothing more happens there.

e)

It is just possible to 43% or 56%. The average case duration shorts to 45.2 weeks and the median to 20.9weeks in both cases. So in average the cases are faster finished.

The median is the instance in the middel. So if you write down all instances sorted, it is the one in the middel. The average is the sum of all instances divided by the number of instances. The average can change a lot for big or extree small oultiers. The median shows more a real duration in the middle of all instance durations.

f)

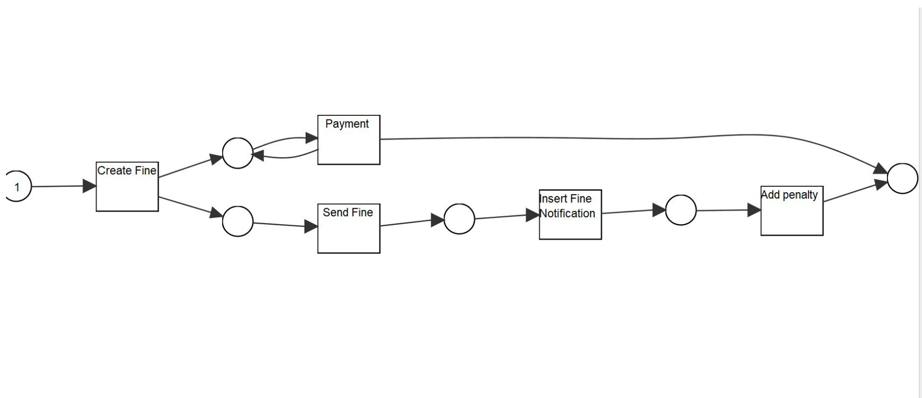


In the first dotted chart you see when which case is active.

Question5fc.jpg

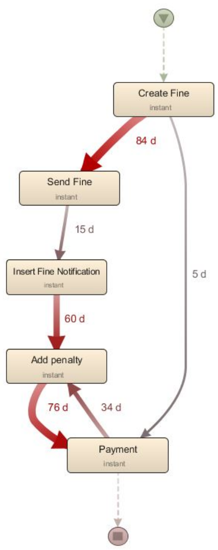
For interpreting the dot chart for the question c) I changed the y-axis to the event names so I can see when which events happening. I also zoomed in so I can see the months of the years. Now you can check better the dates of the peaks.
You can see that insert fine notification is strongly connected to the lashes to see in c). Also a little bit the send fine.
Payment happens close to always but still is a little bit bundled at the peaks.
Furthermore I would say, that in disco it is more easy to see when peaks happen, but in prom better to see what is happen on the peak days.

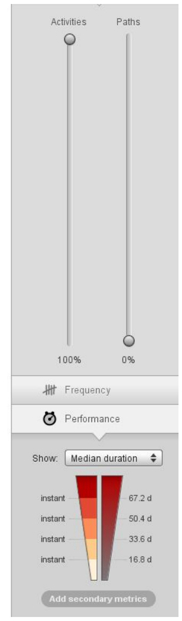
g)



If you apply the alpha-algoirthm on the not filtered data you can not see so clear, that in the resulting chart the payment can happen always and also infinite often. It sounds weird, because you do not expect someone paying before he gets the fine, but looking at the data it happens.
Also in the filtered version the people or pay after the fine is created or pay to late, so that they get a penalty.

h)





84days is the median for send fine. I chose the median, because then you get a better idea what happens most of the times. The mean is 85.1 days.
What you also can see, is that after 60days always a penalty is added.