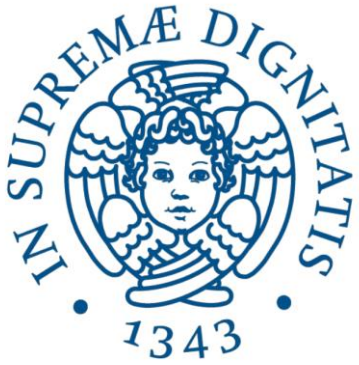


UNIVERSITÀ DI PISA

CUSTOMER CREDIT WORTHINESS

Process Mining Project

A.Y. 2023-2024

Authors:

Basma Adawy

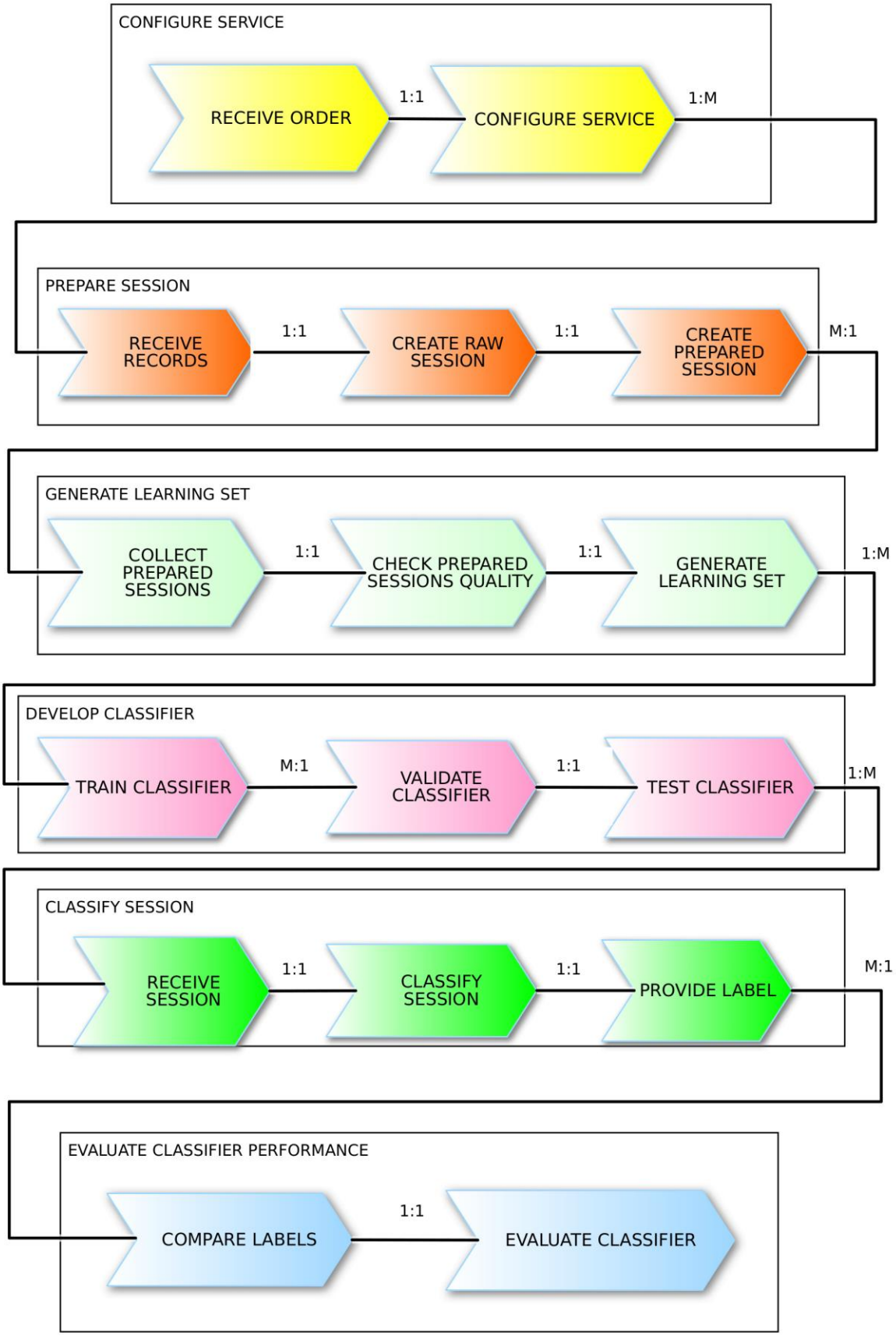
Daniele Laporta

Pietrangelo Manco

Francesco Zingariello

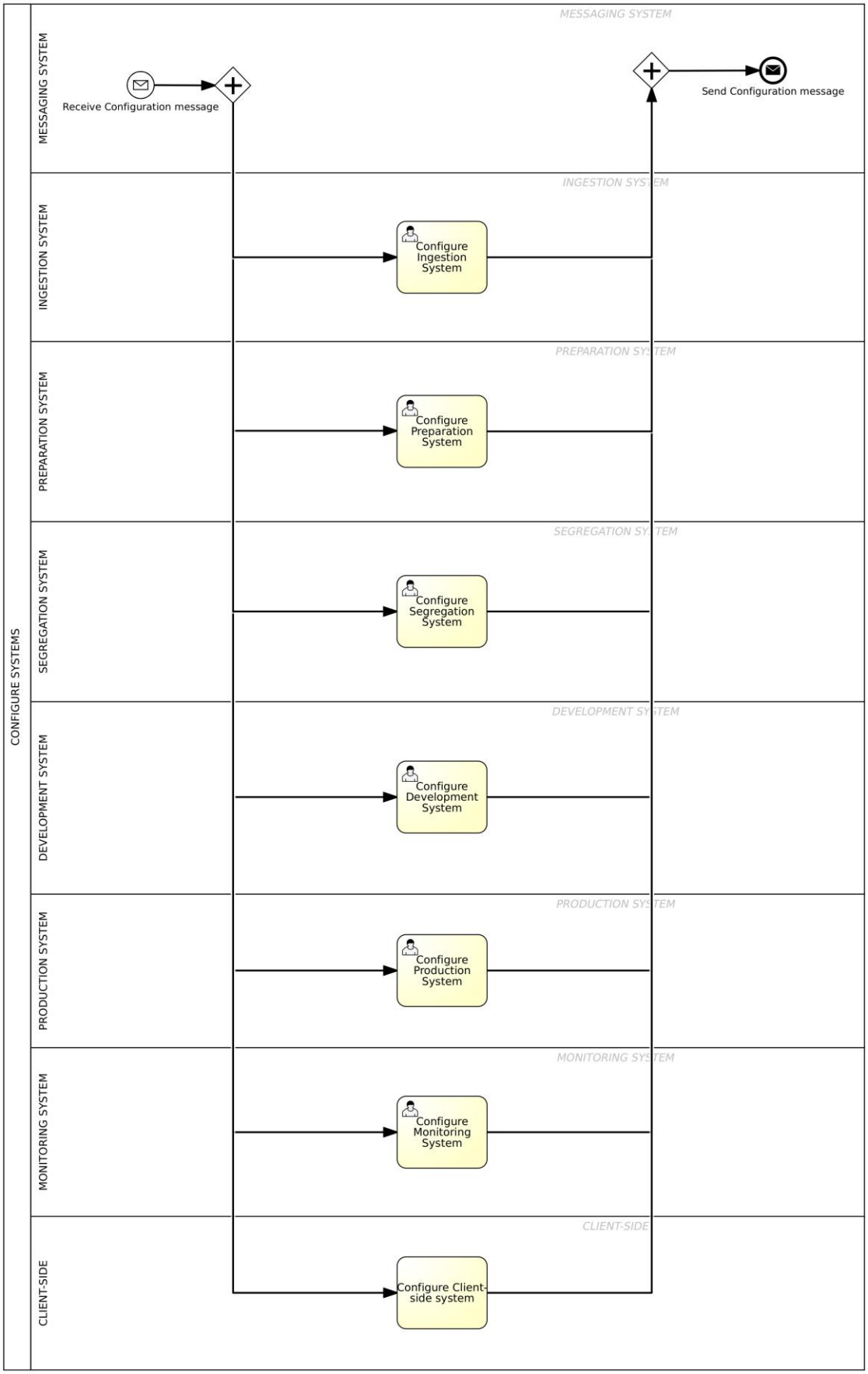
Index

PROCESS LANDSCAPE

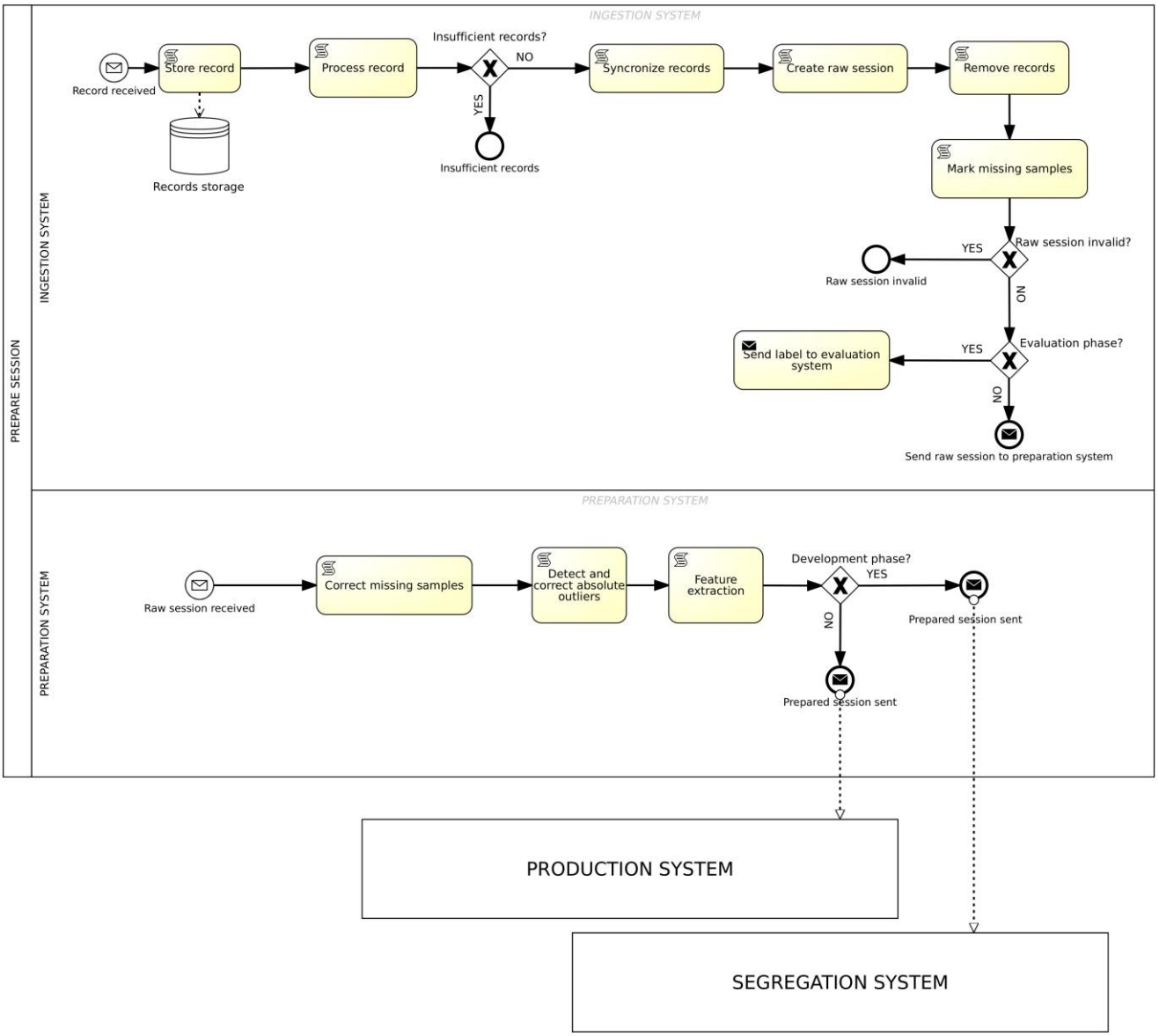


BPMN MODELS

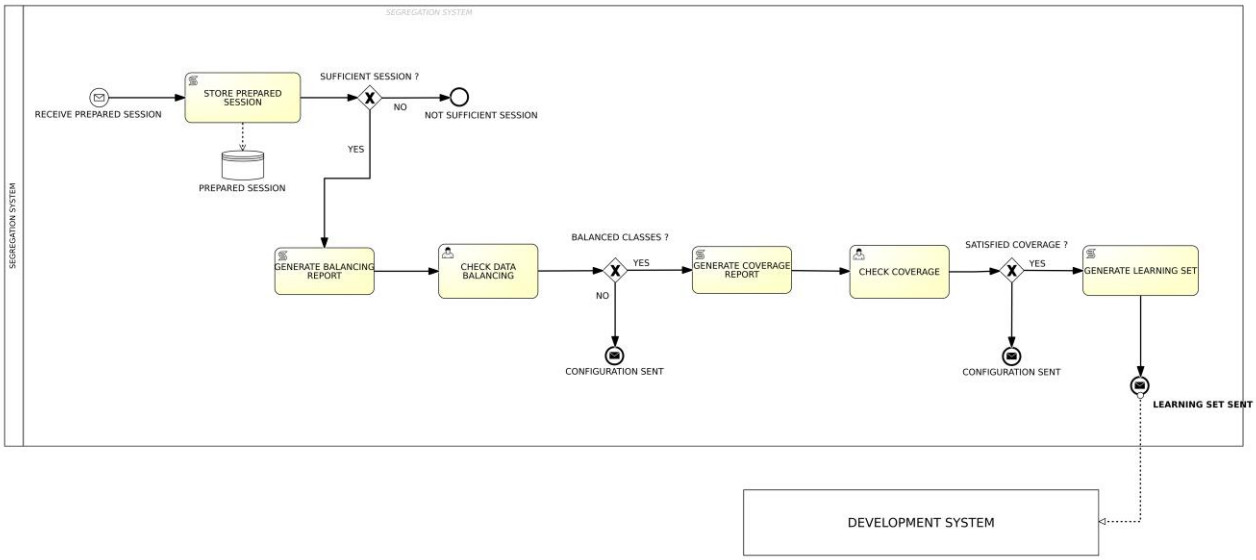
CONFIGURE SYSTEMS (All)



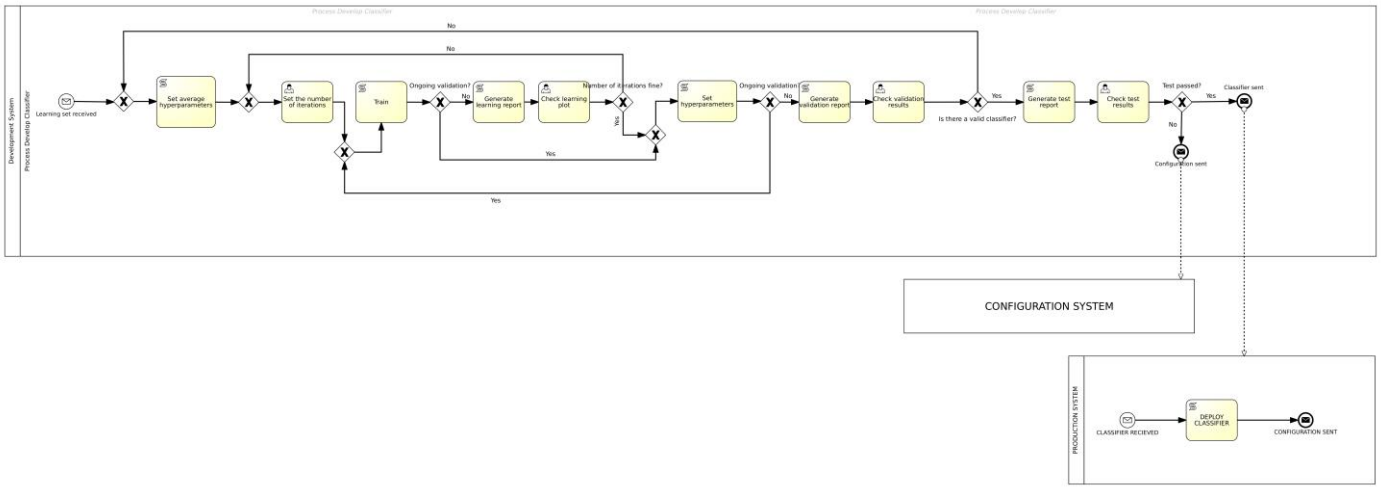
PREPARE SESSION (Daniele Laporta)



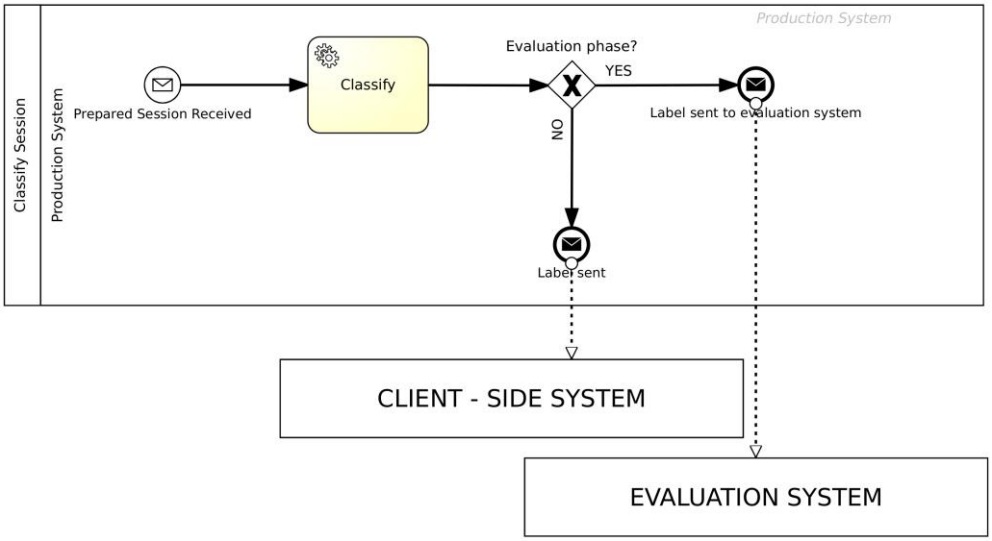
GENERATE LEARNING SET (Basma Adawy)



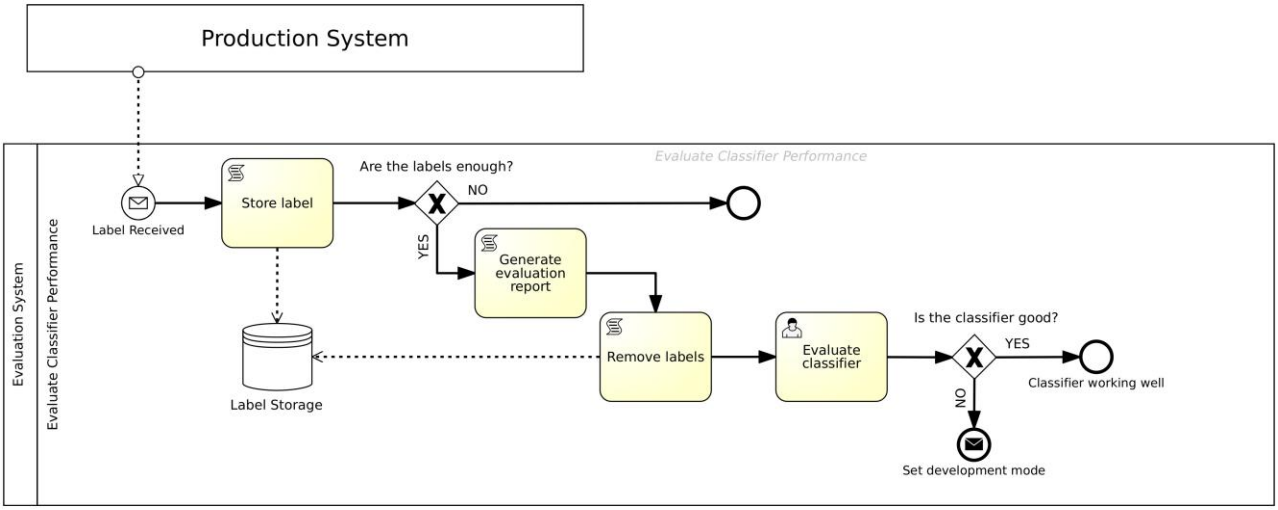
DEVELOP CLASSIFIER (Pietrangelo Manco)



CLASSIFY SESSION (Francesco Zingariello)



EVALUATE CLASSIFIER PERFORMANCE (Francesco Zingariello)



TASK MODELING

Salaries proportion

Actor	Link	Cost (Yearly)	Normalized Cost
Data Analyst	https://www.indeed.com/career/data-analyst/salaries	$76,789	1
Data Engineer	https://www.indeed.com/career/data-engineer/salaries?from=top_sb	$126,124	1.64
ML Engineer	https://www.indeed.com/career/machine-learning-engineer/salaries?from=top_sb	$160,761	2.09
System Administrator	https://www.indeed.com/career/systems-administrator/salaries?from=top_sb	$82,981	1.08

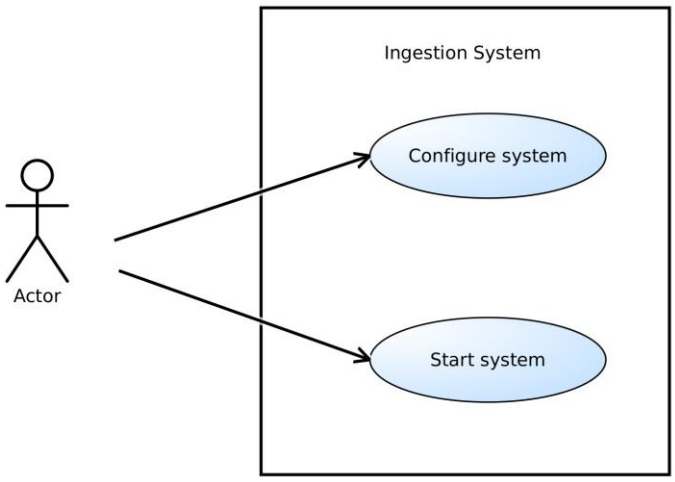
Company Roles:

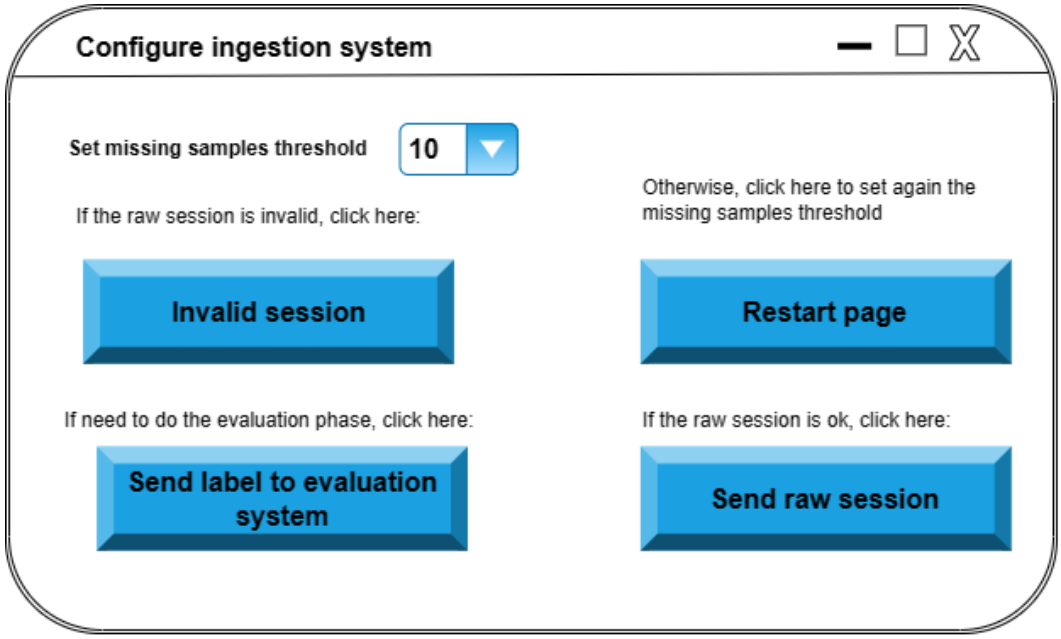
- Data Analyst: A Data Analyst is a professional who is responsible for collecting, analyzing, and interpreting large sets of data to inform business decisions. They use various tools and techniques to extract insights from data and present their findings to stakeholders in a clear and meaningful way. They also play a key role in identifying and defining new process improvement opportunities.
- Data Engineer: A Data Engineer is a professional responsible for the design, development, maintenance, and management of an organization's data infrastructure. They play a key role in helping data scientists and analysts' access and work with large, complex data sets. They work on tasks such as building and maintaining data pipelines, designing, and implementing data storage solutions, and developing and implementing data security measures.
- ML Engineer: A Machine Learning Engineer is a professional with expertise in designing, developing, and deploying machine learning models and systems. These individuals are responsible for applying their knowledge of computer science, statistics, and mathematics to build, implement and maintain machine learning algorithms.
- System Administrator: A system administrator is responsible for the installation, configuration, and ongoing maintenance of an organization's computer systems and servers.

Cognitive Effort values:

- Remember (1): The step can be carried out by remembering another occurrence of the same step;
- Understand (2): The step can be carried out by finding a value in a set of predefined categories;
- Apply (3): The step can be carried out by executing a predefined procedure, encoded by the company;
- Analyze (4): The step can be carried out by finding unknown categories.

Configure ingestion system (Daniele Laporta)





STEP	ACTOR	%	COGNITIVE EFFORT	COST
1. Open the interface to set the parameters	Data Analyst		1	1 * 1 * 1 = 1
2. Show interface	System			
3. Set missing samples threshold	Data Analyst		4	1 * 4 * 1 = 4
4.1 IF raw session invalid				
4.1.1 Click “Invalid Session”	Data Analyst	0.2	1	0.2 * 1 * 1 = 0.2
4.2 ELSE				
4.2.1 IF evaluation phase				
4.2.1.1 Click “Send label to evaluation system”	Data Analyst	0.6		0.6 * 1 * 1 = 0.6
4.2.2 ELSE				
4.2.2.1 Click “restart page” (to set again the missing samples threshold)	Data Analyst	0.2	1	0.2 * 1 * 1 = 0.2
5. Click “Send Raw Session”	Data Analyst		1	1 * 1 * 1 = 1
6. Close report	Data Analyst		1	1 * 1 * 1 = 1
Total cost				8

Cognitive effort description:

1. Remember (1): interact with the user interface.

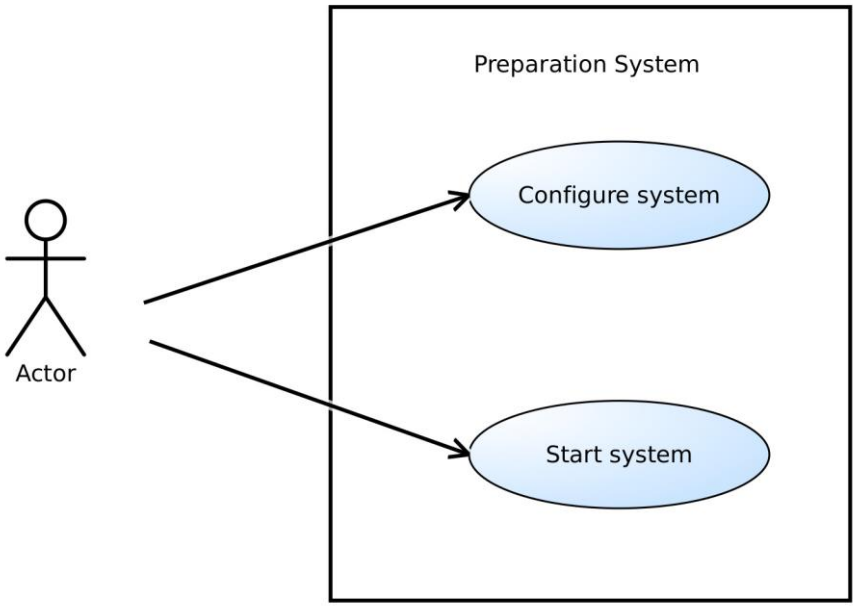
3. Analyze (4): interact with the user interface.

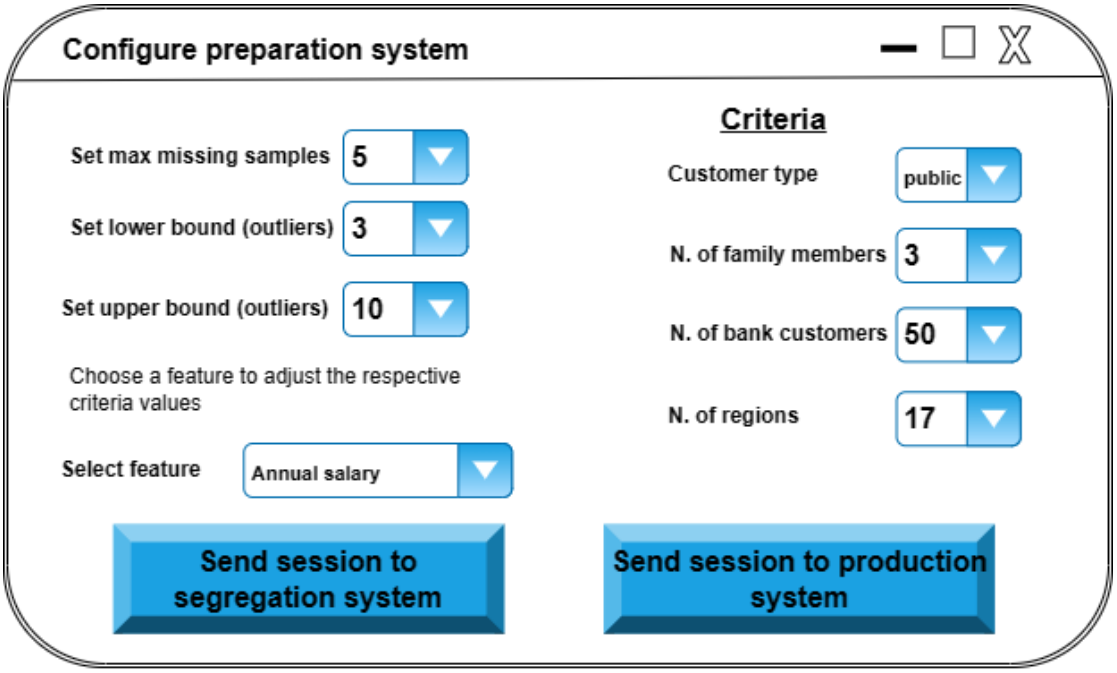
4. Remember (1): interact with the user interface.

5. Remember (1): interact with the user interface.

6. Remember (1): interact with the user interface.

Configure preparation system (Daniele Laporta)





STEP	ACTOR	%	COGNITIVE EFFORT	COST
1. Open the interface to set the parameters	Data Analyst		1	1 * 1 * 1 = 1
2. Show interface	System			
3. Check missing samples				
3.1 Set max missing samples	Data Analyst		2	1 * 2 * 1 = 2
3.2 Correct missing samples	Data Analyst		2	1 * 2 * 1 = 2

4. Check outliers				
4.1 Set lower bound	Data Analyst		2	1 * 2 * 1 = 2
4.2 Set upper bound	Data Analyst		2	1 * 2 * 1 = 2
5. Perform feature extraction				
5.1 Analyse feature “n. of payments” for each type of customer category				
5.1.1 Set max n. of payments allowed for customers of category “public”	Data Analyst		4	1 * 4 * 1 = 4
5.1.2 Set max n. of payments allowed for customers of category “retail”	Data Analyst		4	1 * 4 * 1 = 4
5.1.3 Set max n. of payment for customers of category “internal bank customers”	Data Analyst		4	1 * 4 * 1 = 4
5.2 Analyse feature “n. of payments” for each family member (range 1-5)				
5.2.1 Set n of family members	Data Analyst		2	1 * 2 * 1 = 2
5.3 Analyse feature “n. of payments” based on the n. of customers of the bank				
5.3.1 Set n. of customers of the bank	Data Analyst		4	1 * 4 * 1 = 4
5.4 Analyse feature “n. of payments” based on the variability of regions				
5.4.1 Set n. of regions	Data Analyst		4	1 * 4 * 1 = 4
5.5 Analyse feature “median of payments” for each type of customer category				
5.5.1 Set max n. of payments allowed for customers of category “public”	Data Analyst		4	1 * 4 * 1 = 4
5.5.2 Set max n. of payments allowed for customers of category “retail”	Data Analyst		4	1 * 4 * 1 = 4
5.5.3 Set max n. of payment for customers of category “internal bank customers”	Data Analyst		4	1 * 4 * 1 = 4
5.6 Analyse feature “median of payments” for each family member (range 1-5)				
5.6.1 Set n of family members	Data Analyst		2	1 * 2 * 1 = 2
5.7 Analyse feature “median of payments” based on the n. of customers of the bank				
5.7.1 Set n. of customers of the bank	Data Analyst		4	1 * 4 * 1 = 4
5.8 Analyse feature “median of payments” based on the variability of regions				
5.8.1 Set n. of regions	Data Analyst		4	1 * 4 * 1 = 4
5.9 Analyse feature “annual salary” for each type of customer category				
5.9.1 Set max annual salary allowed for customers of category “public”	Data Analyst		4	1 * 4 * 1 = 4
5.9.2 Set max annual salary allowed for customers of category “retail”	Data Analyst		4	1 * 4 * 1 = 4
5.9.3 Set max annual salary for customers of category “internal bank customers”	Data Analyst		4	1 * 4 * 1 = 4
5.10 Analyse feature “annual salary” for each family member (range 1-5)				

5.10.1 Set n of family members	Data Analyst		2	1 * 2 * 1 = 2
5.11 Analyse feature “annual salary” based on the n. of customers of the bank				
5.11.1 Set n. of customers of the bank	Data Analyst		4	1 * 4 * 1 = 4
5.12 Analyse feature “annual salary” based on the variability of regions				
5.12.1 Set n. of regions	Data Analyst		4	1 * 4 * 1 = 4
5.13 Analyse feature “annual expenses” for each type of customer category				
5.13.1 Set max annual expenses allowed for customers of category “public”	Data Analyst		4	1 * 4 * 1 = 4
5.13.2 Set max annual expenses allowed for customers of category “retail”	Data Analyst		4	1 * 4 * 1 = 4
5.13.3 Set max annual expenses for customers of category “internal bank customers”	Data Analyst		4	1 * 4 * 1 = 4
5.14 Analyse feature “annual expenses” for each family member (range 1-5)				
5.14.1 Set n of family members	Data Analyst		2	1 * 2 * 1 = 2
5.15 Analyse feature “annual expenses” based on the n. of customers of the bank				
5.15.1 Set n. of customers of the bank	Data Analyst		4	1 * 4 * 1 = 4
5.16 Analyse feature “annual expenses” based on the variability of regions				
5.16.1 Set n. of regions	Data Analyst		4	1 * 4 * 1 = 4
5.17 Analyse feature “family members” for each type of customer category				
5.17.1 Set n. of family members allowed for customers of category “public”	Data Analyst		4	1 * 4 * 1 = 4
5.17.2 Set n. of family members allowed for customers of category “retail”	Data Analyst		4	1 * 4 * 1 = 4
5.17.3 Set n. of family members for customers of category “internal bank customers”	Data Analyst		4	1 * 4 * 1 = 4
5.18 Analyse feature “family members” for each family member (range 1-5)				
5.18.1 Set n of family members	Data Analyst		2	1 * 2 * 1 = 2
5.19 Analyse feature “family members” based on the n. of customers of the bank				
5.19.1 Set n. of customers of the bank	Data Analyst		4	1 * 4 * 1 = 4
5.20 Analyse feature “family members” based on the variability of regions				
5.20.1 Set n. of regions	Data Analyst		4	1 * 4 * 1 = 4
5.21 Analyse feature “annual instalment” for each type of customer category				
5.21.1 Set max annual instalment for customers of category “public”	Data Analyst		4	1 * 4 * 1 = 4
5.21.2 Set max annual instalment for customers of category “retail”	Data Analyst		4	1 * 4 * 1 = 4

5.21.3 Set max annual instalment for customers of category “internal bank customers”	Data Analyst		4	1 * 4 * 1 = 4
5.22 Analyse feature “annual instalment” for each family member (range 1-5)				
5.22.1 Set n. of family members	Data Analyst		2	1 * 2 * 1 = 2
5.23 Analyse feature “annual instalment” based on the n. of customers of the bank				
5.23.1 Set n. of customers of the bank	Data Analyst		4	1 * 4 * 1 = 4
5.24 Analyse feature “annual instalment” based on the variability of regions				
5.24.1 Set n. of regions	Data Analyst		4	1 * 4 * 1 = 4
6. IF Development phase	Data Analyst			
6.1 Click “Send session to segregation system”	Data Analyst	0.5	1	0.5 * 1 * 1 = 0.5
7. ELSE				
7.1 Click “Send session to production system”	Data Analyst	0.5	1	0.5 * 1 * 1 = 0.5
8. Close report	Data Analyst		1	1 * 1 * 1 = 1
Total cost				143

Cognitive effort description:

1. Remember (1): interact with the user interface.

3. Remember (1): interact with the user interface.

4. Remember (1): interact with the user interface.

5. Analyse (4): the data analyst views the data distribution and takes decisions based on his experience and data understanding. Specifically, to perform feature extraction, the data analyst needs to specify some criteria, so we have:

- different types of customers (public, retail, internal bank customers),
- the number of family members (range 1-5),
- the number of customers of the bank,
- the variability of regions.

These criteria must be analysed for each of the following features:

- number of payments,
- median of payments,
- annual salary,
- annual expenses,
- family members,
- annual instalment.

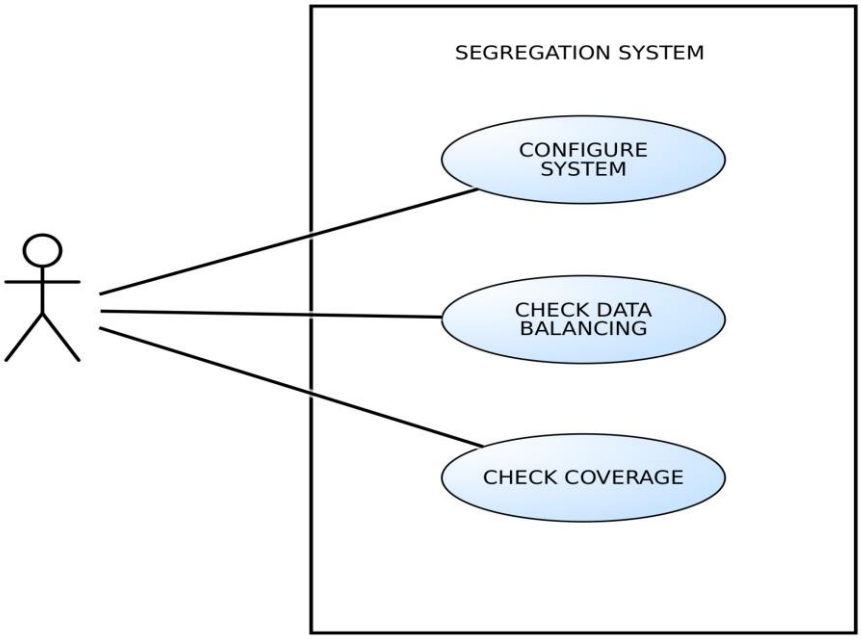
There are 6 features analysed by 4 criteria. All the features have 4 as cognitive cost, except for the setting of the family members number which has a cognitive cost of 2, because is decided out of a specified range.

6. Remember (1): interact with the user interface.

7. Remember (1): interact with the user interface.

8. Remember (1): interact with the user interface.

Configure segregation system (Basma Adawy)



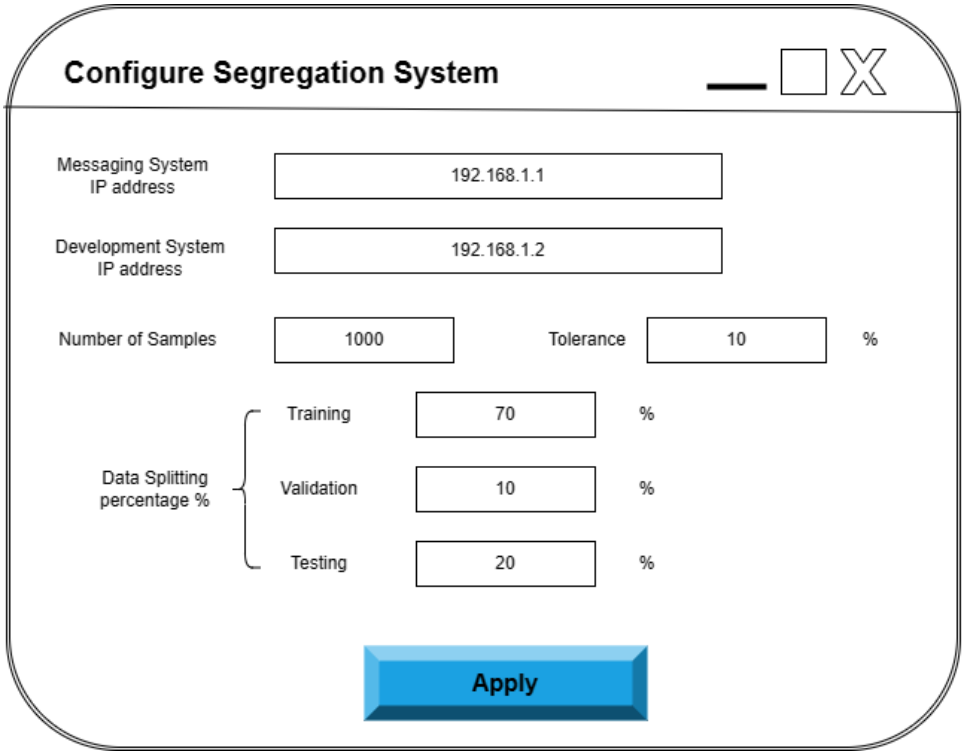
Task description: Segregation system needs to be configured with specific parameters. Those parameters are:

Assign IP addresses to both the development system and the messaging system.

Select the number of samples to be considered before generating the learning sets. These sets are contingent on customer behavior and information.

Define the tolerance threshold for class balancing, expressed as a percentage of the total number of samples. This threshold represents the maximum allowable difference from the average number of samples.

Determine the percentage of data allocated to form the training set, validation set, and test set.



STEP	ACTOR	%	COGNITIVE EFFORT	COST
1. Click on Configure Segregation System	Data Analyst		1	1*1*1 = 1
2. Show configure segregation system interface	System			
3. Select messaging system IP address textbox	Data Analyst		1	1*1*1 = 1
4. Enter the messaging system IP address	Data Analyst		1	1*1*1 = 1
5. Select Development system IP address textbox	Data Analyst		1	1*1*1 = 1
6. Enter the Development system IP address	Data Analyst		1	1*1*1 = 1
7. Select Number of samples textbox	Data Analyst		1	1*1*1 = 1
8. Enter Number of sufficient samples	Data Analyst		4	1*4*1 = 4
9. Select Tolerance textbox	Data Analyst		1	1*1*1 = 1
10. Enter the Tolerance percentage	Data Analyst		1	1*1*1 = 1
11. Select the training split percentage box	Data Analyst		1	1*1*1 =1
12. Enter the training split percentage	Data Analyst		1	1*1*1 =1
13. Select the training split percentage box	Data Analyst		1	1*1*1 =1
14. Enter the validation split percentage	Data Analyst		1	1*1*1 =1
15. Select the test split percentage box	Data Analyst		1	1*1*1 = 1

16. Enter the test split percentage	Data Analyst		1	1*1*1 =1
17. Click Apply	Data Analyst		1	1*1*1 = 1
18. Close the Segregation System Configuration interface	Data Analyst		1	1*1*1 =1
TOTAL COST				20

Cognitive effort description:

1. Remember (1): interact with the user interface.

3. Remember (1): interact with the user interface.

4. Remember (1): interact with the user interface to enter the messaging system IP address.

5. Remember (1): interact with the user interface.

6. Remember (1): interact with the user interface to enter the Development system IP address.

7. Remember (1): interact with the user interface.

8. Analyse (4): The Data analyst enters the number of sufficient samples depending on his experience and understanding of the data.

9. Remember (1): interact with the user interface.

10. Remember (1): interact with the user interface to enter the Tolerance percentage.

11. Remember (1): interact with the user interface.

12. Remember (1): interact with the user interface to enter the training split percentage.

13. Remember (1): interact with the user interface.

14. Remember (1): interact with the user interface to enter the validation split percentage.

15. Remember (1): interact with the user interface.

16. Remember (1): interact with the user interface to enter the test split percentage.

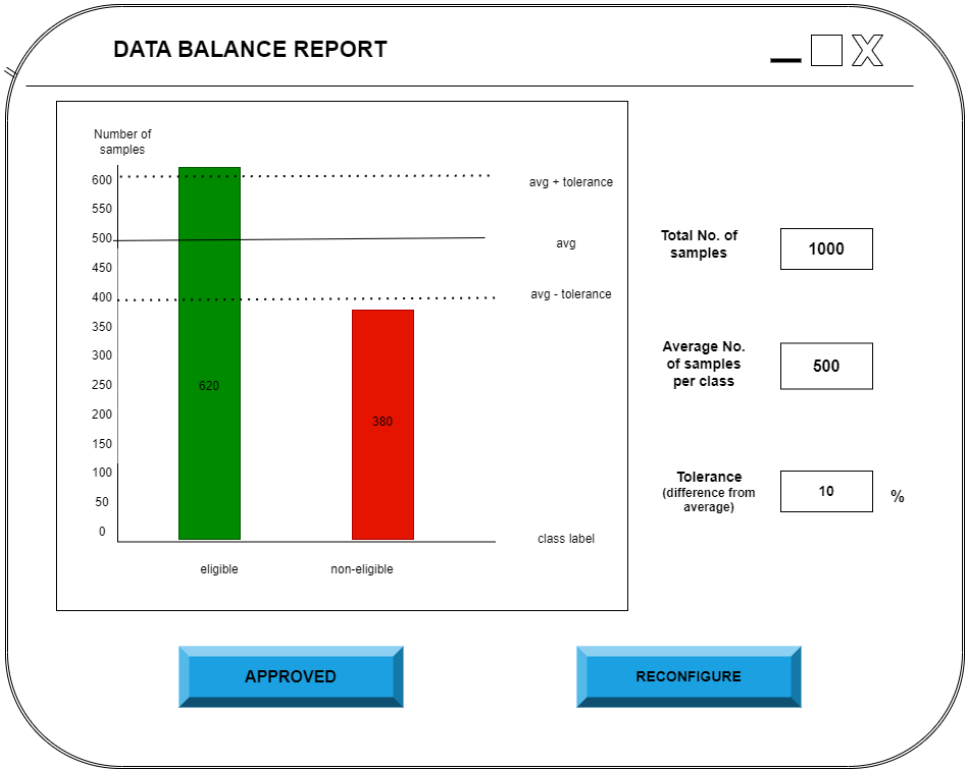
17. Remember (1): interact with the user interface.

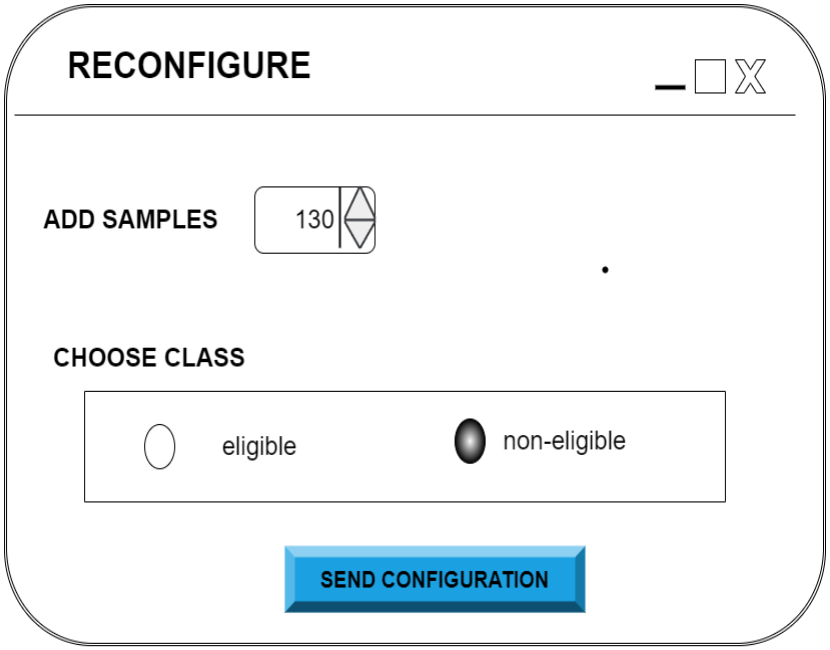
18. Remember (1): interact with the user interface.

Check class balancing interface (Basma Adawy)

Task description: The data analyst must check whether the classes are balanced or not.

The two classes are eligible and non-eligible. During the configuration, a threshold for data balancing is set, we assume this threshold is the Tolerance which is the maximum difference from the average number of samples, expressed in percentage on the total number of samples.





STEP	ACTOR	%	COGNITIVE EFFORT	COST
1. Open the data balance report	Data Analyst		1	1*1*1 = 1
2. Show balance report interface	System			
3. Investigate the data balance report	Data Analyst		2	1*2*1 = 2
3.1 If balancing is true		0.9		
3.1.1 Click APPROVED	Data Analyst	0.9	1	0.9*1*1 = 0.9
3.2 Else		0.1		

3.2.1 Click RECONFIGURE	Data Analyst	0.1	1	0.1*1*1 = 0.1
3.2.2 Show reconfigure interface	System			
3.2.3 Set up reconfiguration message	Data Analyst	0.1	4	1*4*0.1 = 0.4
3.2.4 Click on SEND CONFIGURATION	Data Analyst	0.1	1	1*1*0.1 = 0.1
3.2.5 Send the reconfigure message to the messaging system	System			
4. Close the balance report	Data Analyst		1	1*1*1 = 1
5. END	System			
TOTAL COST				5.5

Cognitive effort description:

1. Remember (1): interact with the user interface.

3. Understand (2): data analyst understands the data balance report, deciding if the samples are enough or not and classes are balanced or no, in the report when a class isn't satisfying the threshold criteria for tolerance it's colour will be red but the analyst if free to approve or reconfigure.

3.1.1 Remember (1): interact with the user interface, if the data is balanced click approved to check coverage.

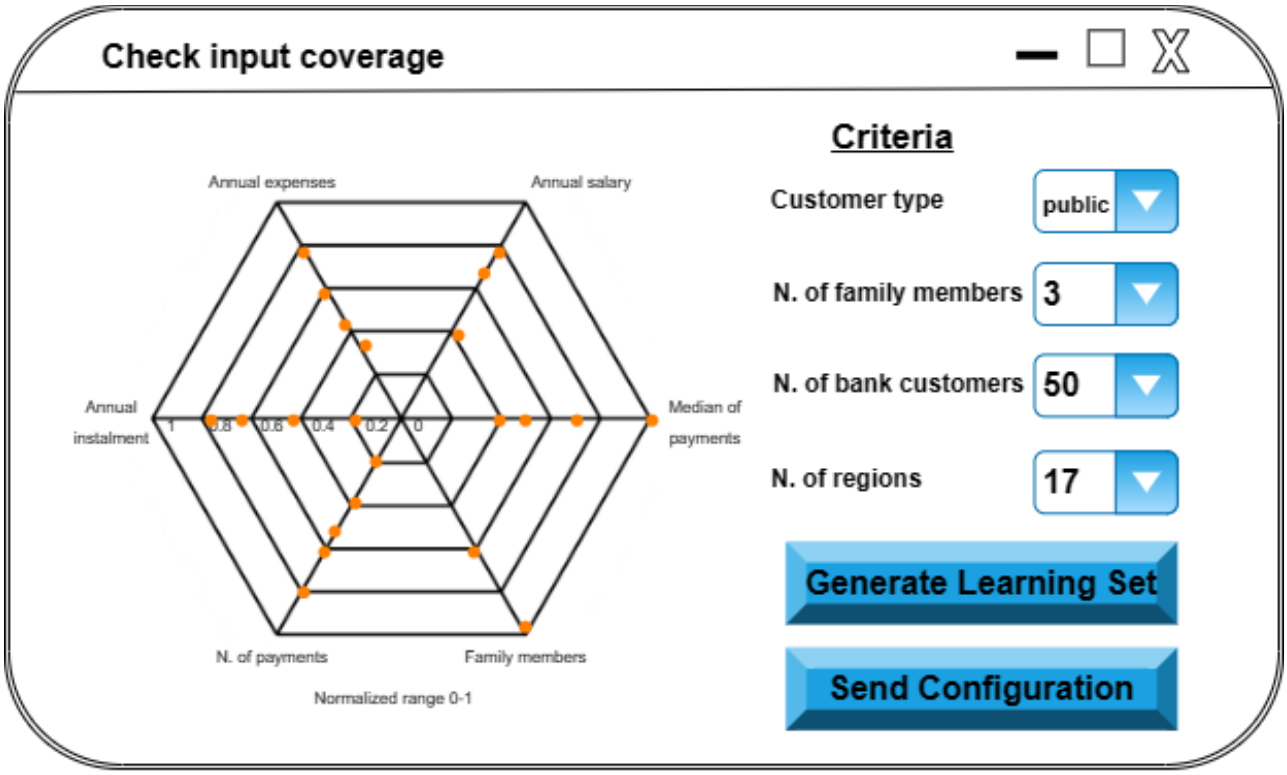
3.2.1 Remember (1): interact with the user interface, if the data is not balanced click reconfigure.

3.2.3 Analyse (4): data analyst analyses the data by setting up reconfiguration message. The data analyst increases or decrease the number of samples in a class depending on his understand and experience to send this reconfiguration to the messaging system.

3.2.4 Remember (1): interact with the user interface.

4. Remember (1): interact with the user interface.

Check input coverage interface (Daniele Laporta)



STEP	ACTOR	%	COGNITIVE EFFORT	COST
1. Open the input coverage graph	Data Analyst		1	1 * 1 * 1 = 1
2. Show radar diagram interface	System			
3. Analyse the radar diagram	Data Analyst		4	1 * 4 * 1 = 4
3.1 IF input data is enough		0.9		
3.1.1 Generate learning set	Data Analyst	0.9	1	0.9 * 1 * 1 = 0.9
3.2 ELSE		0.1		
3.2.1 End, Send configuration	Data Analyst	0.1	1	0.1 * 1 * 1 = 0.1
4. System shows a confirmation message	System			
5. Close report	Data Analyst		1	1 * 1 * 1 = 1
Total cost				7

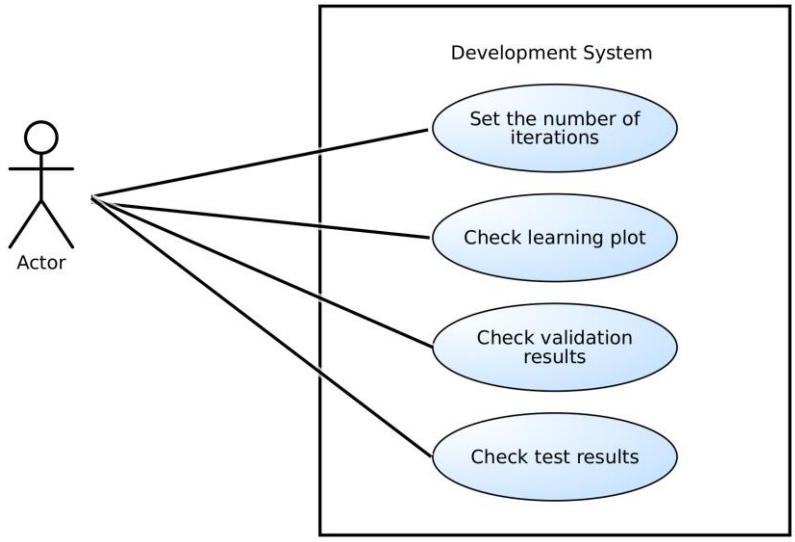
Cognitive effort description:

1. Remember (1): interact with the user interface.

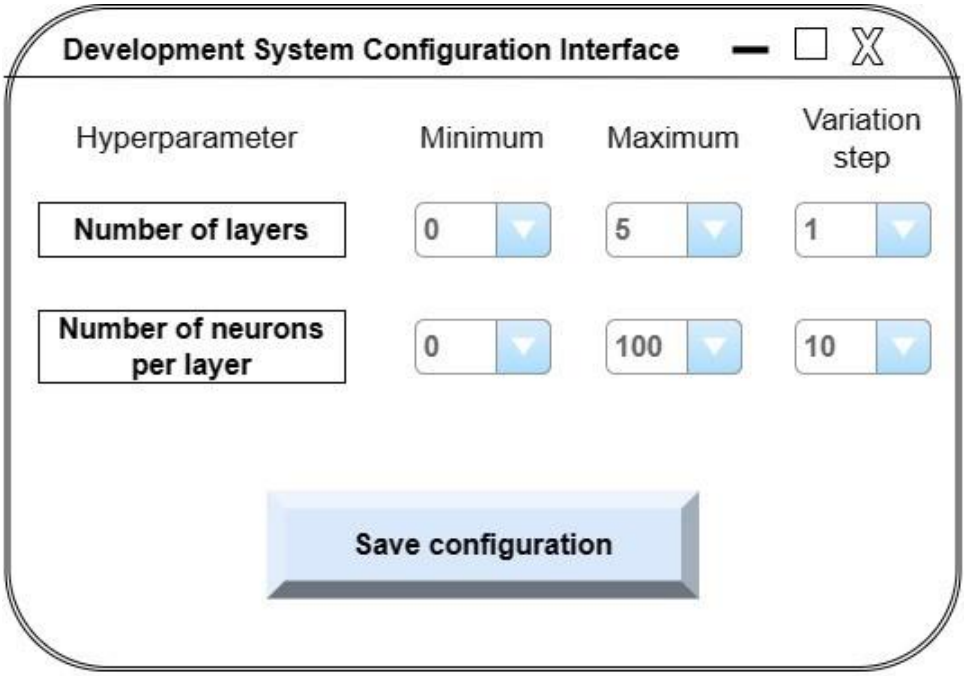
3. Analyse (4): data analyst analyses the radar diagram, deciding if the input coverage is enough or not. The data analyst should evaluate each feature of the radar diagram and assess if the coverage is uniformly distributed. So, generate the learning set for the upcoming development system or end the process and send the configuration.

5. Remember (1): interact with the user interface.

Development System Configuration (Pietrangelo Manco)



Task description: Development system needs its hyperparameters to be configured. Those parameters are the number of layers and the number of neurons per layer.



STEP	ACTOR	%	COGNITIVE EFFORT	COST
1. Open the interface to configure the hyperparameters	M.L. Engineer		1	1 * 2.09
2. Click on the textbox to set the number of layers interval	M.L. Engineer		1	1 * 2.09
3. Set the number of layers interval	M.L. Engineer		4	4 * 2.09
4. Click on the textbox to set the number neurons per layer interval	M.L. Engineer		1	1 * 2.09
5. Set the number of neurons per layer interval	M.L. Engineer		4	4 * 2.09
6. Save the configuration	M.L. Engineer		1	1 * 2.09

7. Close the interface	M.L. Engineer		1	1 * 2.09
Total cost				27.17

Cognitive effort description:

1. Remember (1): interact with the development system configuration interface to open it.

2. Remember (1): interact with the interface to click on the textbox of the first hyperparameter.

3. Analyse (4): the Machine Learning Engineer must decide the correct interval of layers to set, using his knowledge.

4. Remember (1): interact with the interface to click on the textbox of the second hyperparameter.

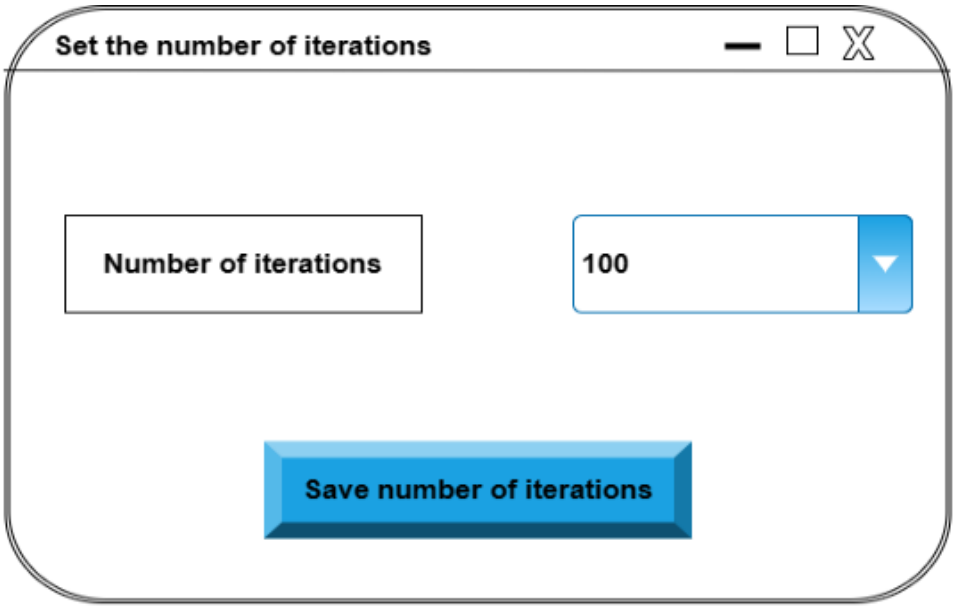
5. Analyse (4): the Machine Learning Engineer must decide the correct interval of neurons per layer to set, using his knowledge.

6. Remember (1): interact with the interface to click on the button to save the new configuration.

7. Remember (1): interact with the interface to close it.

Set number of iterations (Pietrangelo Manco)

Task description: the Machine Learning Engineer set the current number of iterations, based on the parameter received.



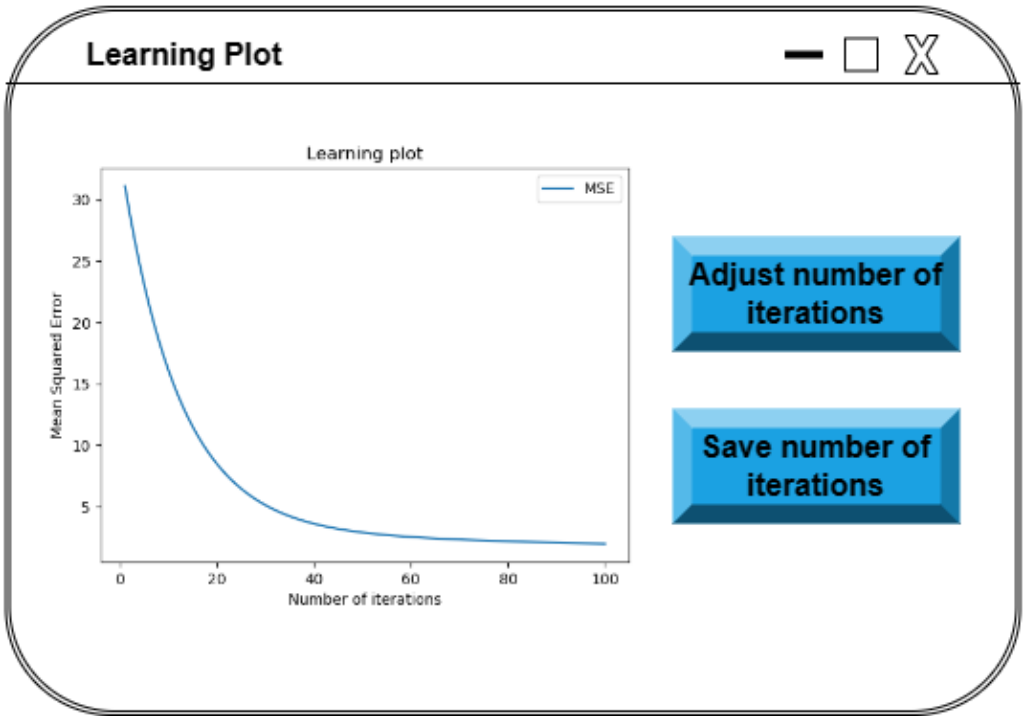
STEP	ACTOR	%	COGNITIVE EFFORT	COST
1. Open the interface to set the number of iterations	M.L. Engineer		1	1 * 2.09
2. Adjust the number of iterations based on the parameter	M.L. Engineer		1	1 * 2.09
3. Save the current configuration	M.L. Engineer		1	1 * 2.09
4. Close the interface	M.L. Engineer		1	1 * 2.09
Total cost				8.36

Cognitive effort description:

1. Remember (1): interact with the user interface to open it.
2. Remember (1): interact with the user interface to insert the desired number.
3. Remember (1): interact with the user interface to click the save button.
4. Remember (1): interact with the user interface to close it.

Check learning plot (Pietrangelo Manco)

Task description: the Machine Learning Engineer must check the learning plot to decide if the current number of iterations is good or needs to be adjusted. To do so, he must check if the plot is flat for more than half of the horizontal axis: if it is, then he must reduce the number of iterations to avoid overfitting; else, if the plot isn't flat at the end of the horizontal axis, then he must increase the number of iterations; else, the number of iterations is fine.

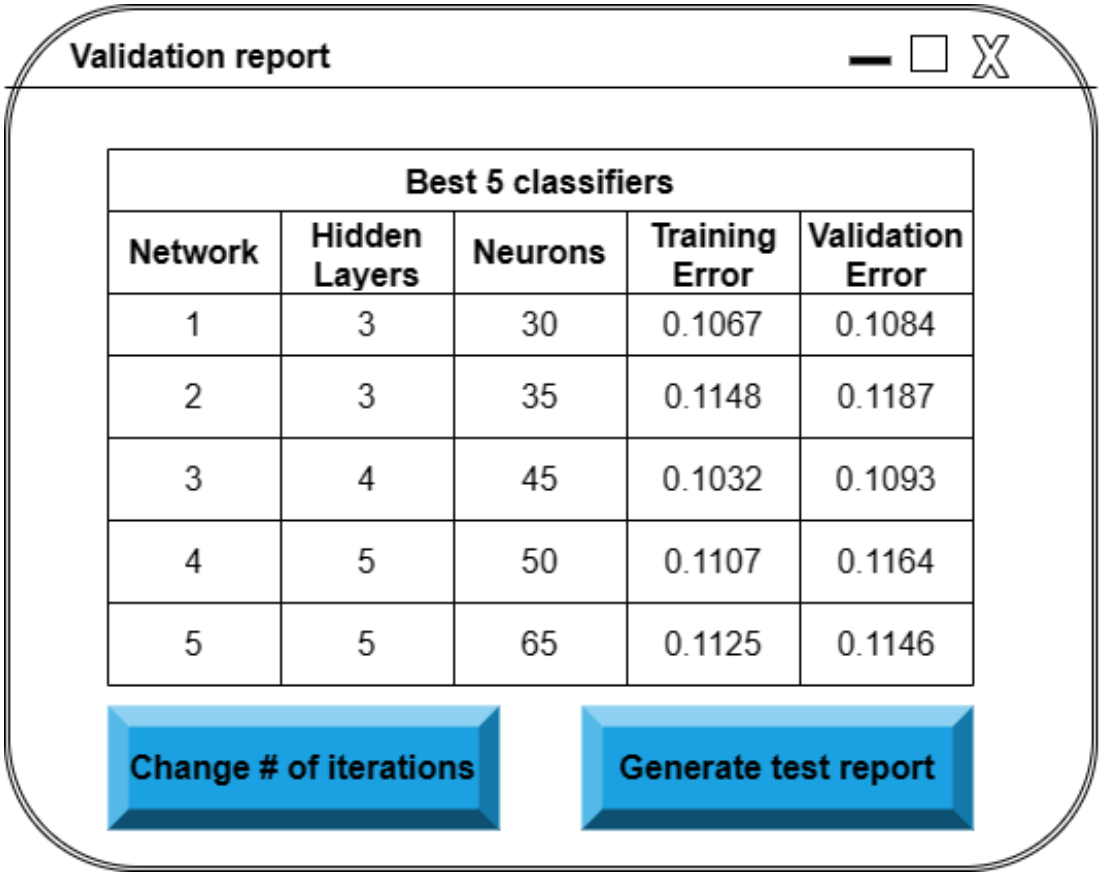


STEP	ACTOR	%	COGNITIVE EFFORT	COST
1. Open training report	M.L. Engineer		1	1*2.09
2. Is the plot flat for more than half of the horizontal axis?	M.L. Engineer	20%	3	0.2*3* 2.09
2.1. Reduce the number of iterations				
3. Is the plot flat at the end of the horizontal axis?	M.L. Engineer	20%	3	0.2*3 * 2.09
3.1. Increase the number of iterations				
4. If none of the above, the number of iterations is fine	M.L. Engineer	60%	1	0.6*1*2.09
5. Save the configuration	M.L. Engineer		1	1*2.09
6. Close the training report	M.L. Engineer		1	1*2.09
Total cost				10.032

Cognitive effort description:

1. Remember (1): interact with the user interface to open the report.
2. Apply (3): apply a rule (reduce the number of iterations) if the condition is met.
3. Apply (3): apply a rule (increase the number of iterations) if the condition is met.
4. Remember (1): check that no above condition is met.
5. Remember (1): interact with the user interface to save the current configuration.
6. Remember (1): interact with the user interface to close the report.

Check validation results (Daniele Laporta)



STEP	ACTOR	COGNITIVE EFFORT	COST
1. Open validation report	ML Engineer	1	1 * 2.09 = 2.09
2. Show list of 5 best classifiers	System		
3. Choose the best 3 classifiers (lower validation error)	ML Engineer	3	3 * 2.09 = 6.27
4. Choose the two with lower n. neurons (among the 3)	ML Engineer	3	3 * 2.09 = 6.27
5. Choose the one with less hidden layers (if draw, choose the one with less validation error	ML Engineer	3	3 * 2.09 = 6.27
6. Select the best network	ML Engineer	1	1 * 2.09 = 2.09
7. Save the best network	ML Engineer	1	1 * 2.09 = 2.09
8. Close the report	ML Engineer	1	1 * 2.09 = 2.09
Total cost			27.17

Cognitive effort description:

1. Remember (1): Interact with the user interface

3. Apply (3): Apply a rule to select the best classifier among the best 5 classifiers

4. Apply (3): Apply a rule to select the best classifier among the best 5 classifiers

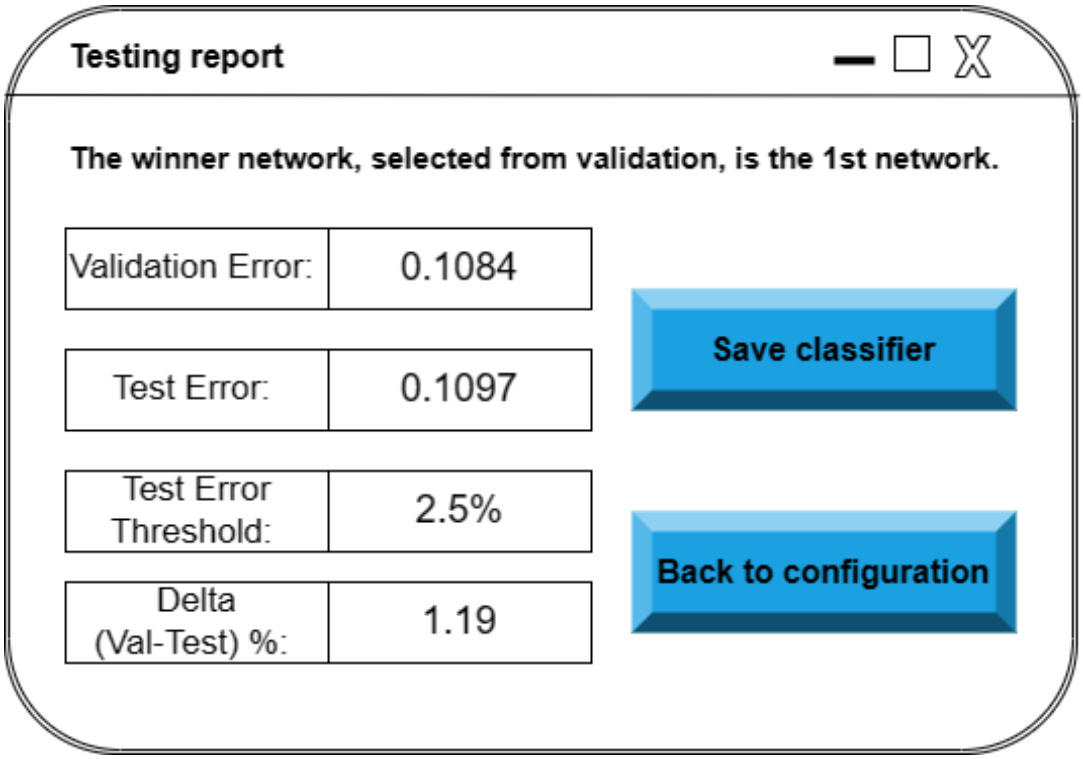
5. Apply (3): Apply a rule to select the best classifier among the best 5 classifiers

6. Remember (1): Interact with the user interface

7. Remember (1): Interact with the user interface

8. Remember (1): Interact with the user interface

Check test results (Daniele Laporta)



STEP	ACTOR	%	COGNITIVE EFFORT	COST
1. Open testing report	ML Engineer		1	1 * 1 * 2.09 = 2.09
2. Show testing report	System			
3. Understand if the error is under the threshold	ML Engineer		2	1 * 2 * 2.09 = 4.18
3.1. IF test error is under the threshold		0.9		
3.1.1 Save the classifier	ML Engineer	0.9	1	0.9 * 1 * 2.09 = 1.881
3.2 ELSE		0.1		
3.2.1 Go back to configuration	ML Engineer	0.1	1	0.1 * 1 * 2.09 = 0.209
4. Close testing report	ML Engineer		1	1 * 1 * 2.09 = 2.09

Total cost	10.45

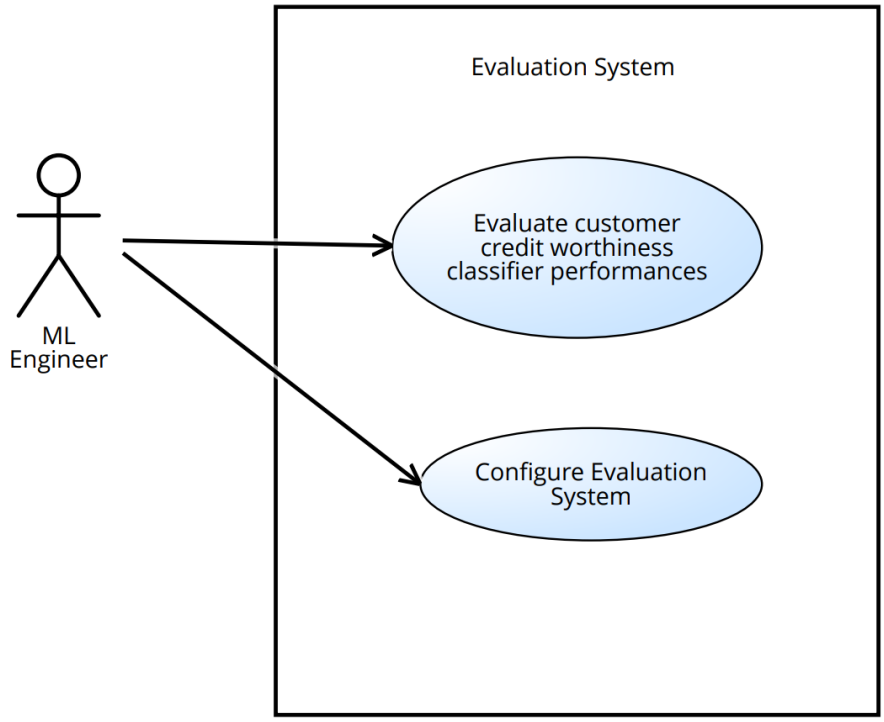
Cognitive effort description:

1. Remember (1): Interact with the user interface

3. Understand (2): The ML engineer needs to understand if the test error is under the threshold. If true, the ML engineer saves the classifier, else clicks the go back to configuration button.

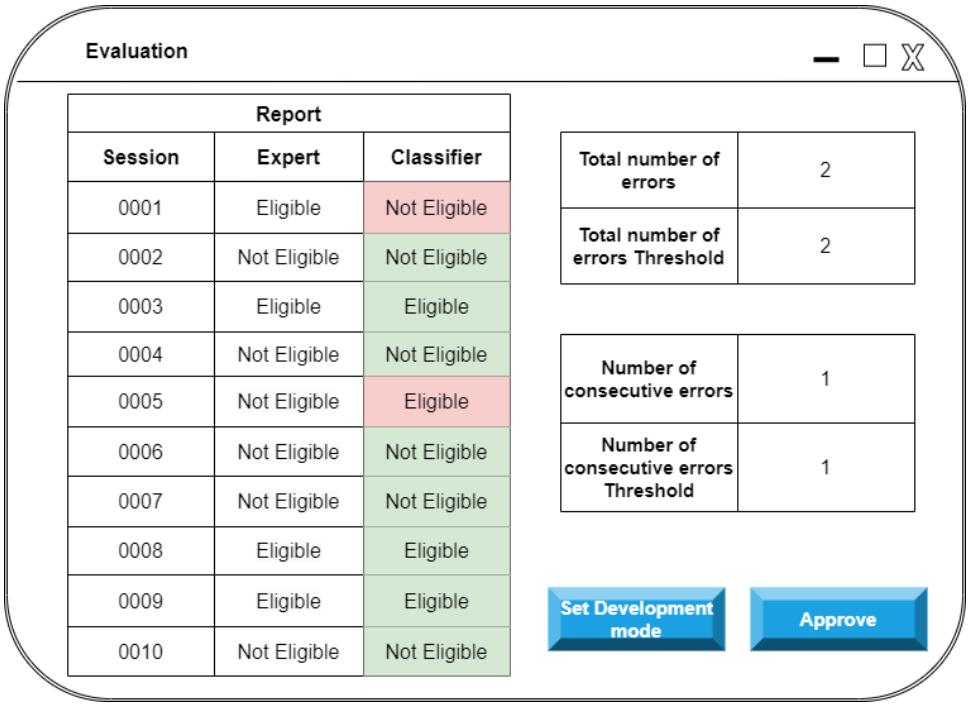
4. Remember (1): Interact with the user interface

Evaluation System Configuration (Francesco Zingariello)



STEP	ACTOR	%	COGNITIVE EFFORT	COST
1. Open the evaluation system configuration	ML Engineer		1	1*1*2.09 = 2.09
3. Displays the monitoring configuration interface System	System			
4. Set number of session used for monitoring	ML Engineer		4	1*4*2.09 = 8.36
5. Set total number of errors threshold	ML Engineer		4	1*4*2.09 = 8.36
5. Set number of consecutive errors threshold	ML Engineer		4	1*4*2.09 = 8.36
7. Close evaluation system configuration	ML Engineer		1	1*1*2.09 = 2.09
Total cost				29.26

Evaluation report interface (Francesco Zingariello)



STEP	ACTOR	%	COGNITIVE EFFORT	COST
1. Open evaluation tool	ML Engineer		1	1 * 1 * 2.09 = 2.09
2. Show evaluation report	System			
3. Understand if the Total number of Errors is under the Total number of errors Threshold	ML Engineer		2	1 * 2 * 2.09 = 4.18
3.1. IF Total number of errors is over the threshold		0.1		
3.1.1 Set development mode	ML Engineer	0.1	1	0.1 * 1 * 2.09 = 0.209
3.2 ELSE		0.9		
3.2.1 Understand if the Number of consecutive errors is under the Number of consecutive errors Threshold	ML Engineer	0.9	2	0.9 * 2 * 2.09 = 3.762
3.2.1.1 IF the Number of Consecutive Errors is under the threshold		0.8		
3.2.1.1.1 Approve classifier	ML Engineer	0.8	1	0.8 * 1 * 2.09 = 1.672
3.2.1.2 ELSE		0.1		
3.2.1.2.1 Set development mode	ML Engineer	0.1	1	0.1 * 1 * 2.09 = 0.209
4. Close testing report	ML Engineer		1	1 * 1 * 2.09 = 2.09
Total cost				14.212

Task description: the Machine Learning Engineer must check the report to approve the classifier based on its performance. To do so, he must check if the total number of errors is under a certain threshold: if it is, then he must check also if the number of consecutive errors is under another threshold, and finally approve the classifier. Otherwise, if one of the two numbers is above its threshold, he must set Development mode again.

Cognitive effort description:

1) Remember (1): interact with the user interface to open the report.

3) Understand (2): The ML engineer needs to understand if the Total number of Errors is under the Total number of errors Threshold.

3.1.1) Remember (1): Interact with user interface to set development mode.

3.2.1) Understand (2): The ML engineer needs to understand if the Number of Consecutive Errors is under the Number of Consecutive Errors threshold.

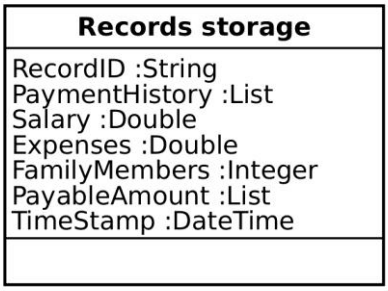
3.2.1.1.1) Remember (1): interact with the user interface to approve the classifier.

3.2.1.2.1) Remember (1): interact with the user interface to set development mode.

4) Remember (1): interact with the user interface to close the report.

DATA MODELING

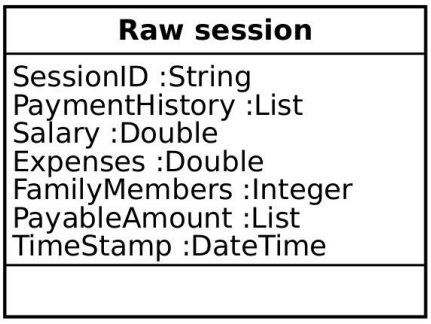
UML class diagram of Records storage (Daniele Laporta)



Store the records received in a storage. This is the first storage produced since the beginning of the process.

- **RecordID**: A unique identifier for the record.
- **PaymentHistory**: A list containing all the payments made.
- **Salary**: The amount of the salary of the customer.
- **Expenses:** The amount of the expenses of the customer.
- **FamilyMembers:** The number of family members of the customer.
- **PayableAmount:** The financing instalment, period, horizon of the customer.
- **Timestamp**: The date and time when the record was registered.

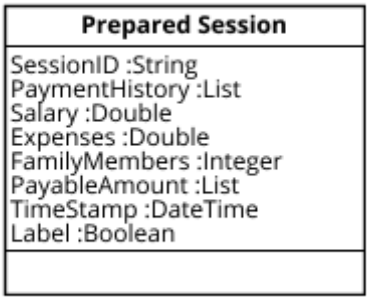
UML class diagram of Raw session (Daniele Laporta)



The session is almost ready. It needs to be cleaned by missing values and outliers and needs to extract features.

- **SessionID**: A unique identifier for the raw session.
- **PaymentHistory**: A list containing all the payments made.
- **Salary**: The amount of the salary of the customer.
- **Expenses**: The amount of the expenses of the customer.
- **FamilyMembers**: The number of family members of the customer.
- **PayableAmount**: The financing instalment, period, horizon of the customer.
- **Timestamp**: The date and time when the record was registered.

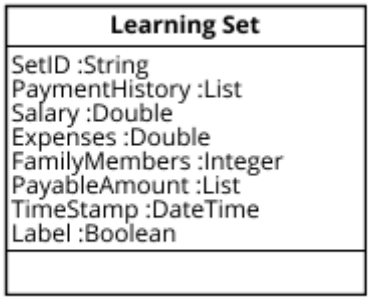
UML class diagram of prepared session storage (Basma Adawy)



Store the data received or processed during a preparation session for the customer credit worthiness analysis.

- **SessionID**: A unique identifier for the raw session.
- **PaymentHistory**: A list containing all the payments made.
- **Salary**: The amount of the salary of the customer.
- **Expenses**: The amount of the expenses of the customer.
- **FamilyMembers**: The number of family members of the customer.
- **PayableAmount**: The financing instalment, period, horizon of the customer.
- **Timestamp**: The date and time when the record was registered.
- **label:** Containing the corresponding output labels associated with the features. Eligibility labels (e.g., eligible, non-eligible).

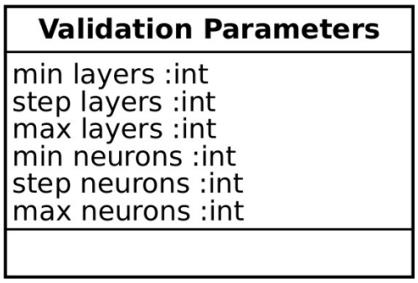
UML class diagram of learning set (Basma Adawy)



The learning set often referred to as a dataset typically includes features and labels.

- **SessionID**: A unique identifier for the raw session.
- **PaymentHistory**: A list containing all the payments made.
- **Salary**: The amount of the salary of the customer.
- **Expenses**: The amount of the expenses of the customer.
- **FamilyMembers**: The number of family members of the customer.
- **PayableAmount**: The financing instalment, period, horizon of the customer.
- **Timestamp**: The date and time when the record was registered.
- **label:** Containing the corresponding output labels associated with the features. Eligibility labels (e.g., eligible, non-eligible).

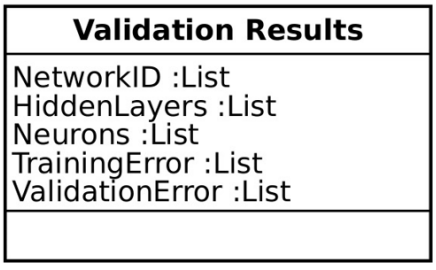
UML class diagram of validation parameters (Pietrangelo Manco)



The validation parameters go as follows:

- **Min layers:** the minimum number of layers to be set.
- **Step layers:** the minimum increment or decrement in layers between iterations.
- **Max layers:** the maximum number of layers to be set.
- **Min neurons:** the minimum number of neurons to be set.
- **Step layers:** the minimum increment or decrement in neurons between iterations.
- **Max layers:** the maximum number of neurons to be set.

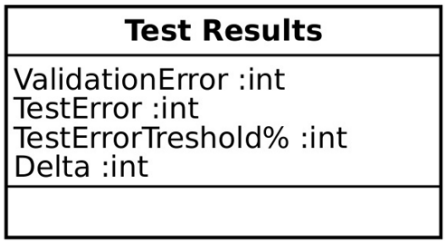
UML class diagram of validation results (Pietrangelo Manco)



The classifiers that are given as a result by the validation process.

- **Networkwide:** a list of the classifiers' identifiers.
- **HiddenLayers:** a list containing the respective classifiers' number of layers.
- **Neurons:** a list containing the corresponding classifiers' neurons.
- **TrainingError:** The error associated to the corresponding classifier's training process.
- **ValidationError:** The error associated to the corresponding classifier's validation process.

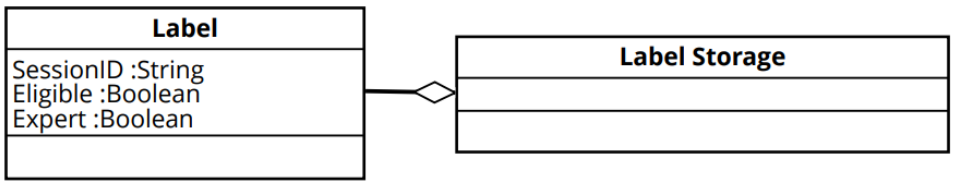
UML class diagram of test results (Pietrangelo Manco)



On of the validation classifiers is chosen as the best one after the testing phase.

- **ValidationError:** The error associated to the winner classifier's validation process.
- **TestError:** The error associated to the winning classifier's testing process.
- **TestErrorTreshold:** the maximum error after the testing process to accept the classifier.
- **Delta:** the difference between the validation error and the test error.

UML class diagram of label and label storage (Francesco Zingariello)

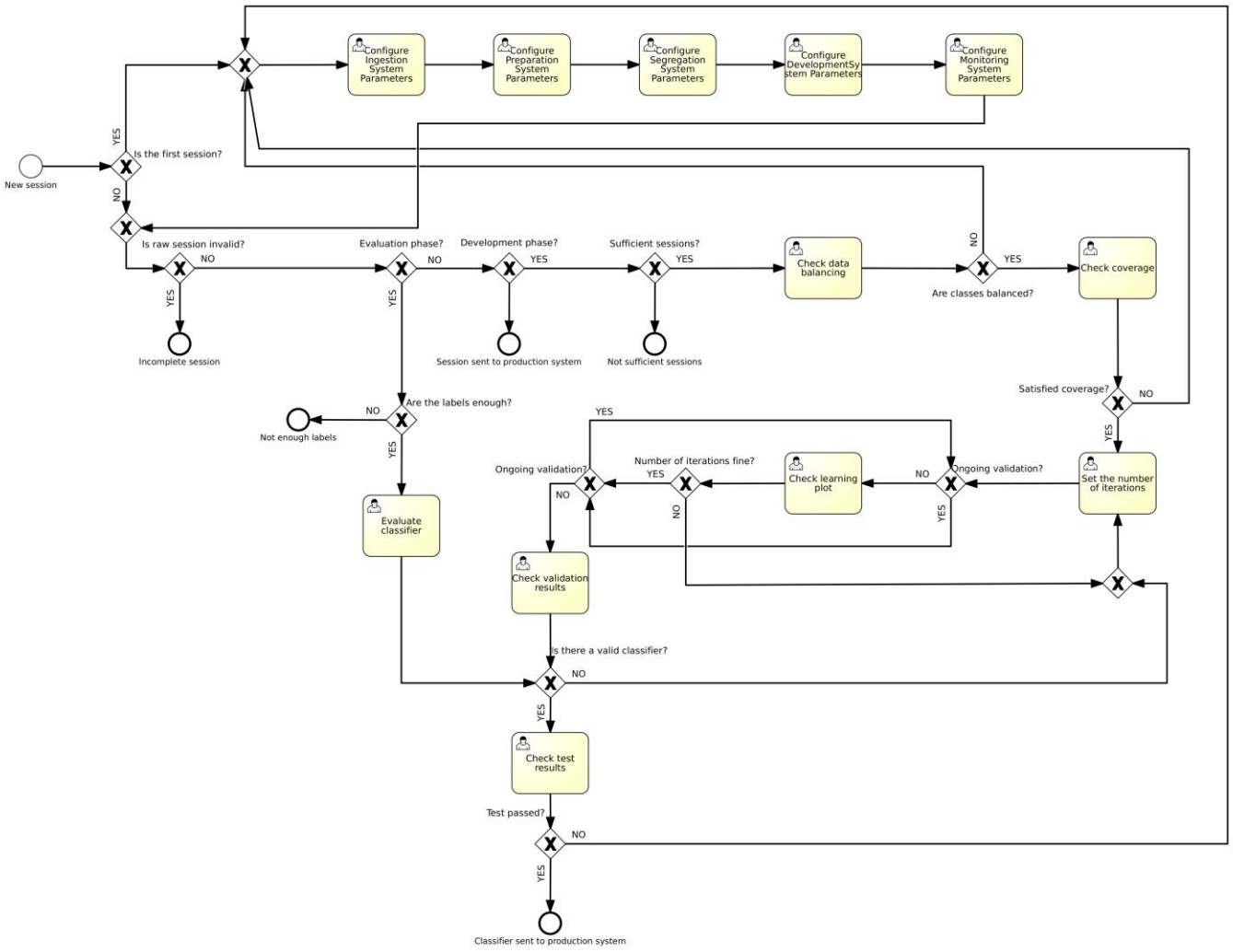


Label storage stores all labels received to produce an evaluation report and decide if a classifier performs well or not.

For Label class, we have:

- **SessionID**: ID of the label's session.
- **Eligible**: the label itself, true if Eligible.
- **Expert**: if it's a label from an expert (true) or from the classifier(false).

AS-IS MODEL (All)



Assumptions:

- We consider as our simulation token the arrival of the new customer credit session.
- We assume 500 sessions for development, 5000 sessions for production, 50 sessions under evaluation.
- We assume a 95% sample of valid raw sessions.
- We have 500 sessions for the development of 1 classifier.
- We assume 5 iterations for the class balance.
- We assume 3 iterations for the input coverage.
- We need 50 sessions per classifier for evaluation.
- We assume 6 iterations for a fine evaluation.
- We need 4 iterations of grid search evaluation, as we compute 2 values for 2 hyperparameters.
- We assume 5 iterations to decide the correct number of iterations in development.
- We assume the test gets passed 99% of the time.
- Inter arrival time fixed to 40 seconds.
- Simulation done with 10000 tokens, as it's bimp's limit.
- Sessions scaled to 1110 due to the simulator's limit.
- In Resources, timetable 24/7.
- Time 00-23.59.

Tasks & gateways percentages

Exclusive gateway (XOR)	True %	False %	Computation
Is it the first session?	0.09	99.91	1/1110
Is raw session invalid?	5	95	56/1110
Evaluation phase?	0.9	99.1	10/1110
Development phase?	9.09	90.91	100/1100(total – evaluation)
Sufficient sessions?	1	99	1/100
Are classes balanced?	20	80	1/5 (given)
Satisfied coverage?	33	67	1/3 (given)
Are the labels enough?	10	90	1/10
Is the classifier good?	14	86	1/7 (given)
Ongoing validation? (1)	80	20	4/5
Number of iterations fine?	20	80	1/5 (given)
Ongoing validation? (2)	80	20	4/5
Is there a valid classifier?	14	86	1/7 (given)
Test passed?	99	1	99/100 (given)

Simulation Results

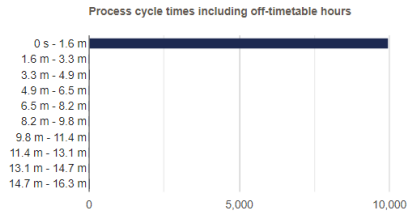
General information

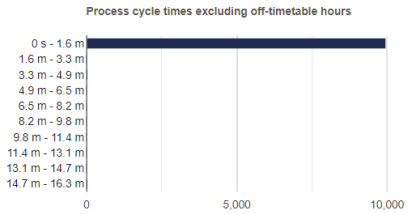
Completed process instances 10000

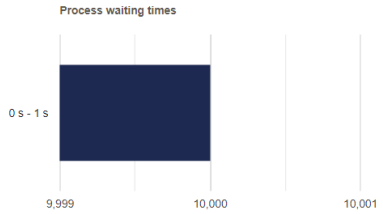
Total cost 0 EUR

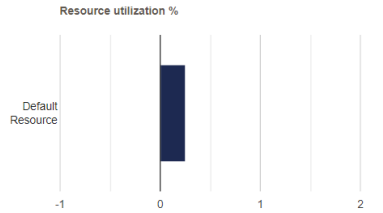
Total simulation time 3.6 weeks

Charts









Scenario Statistics

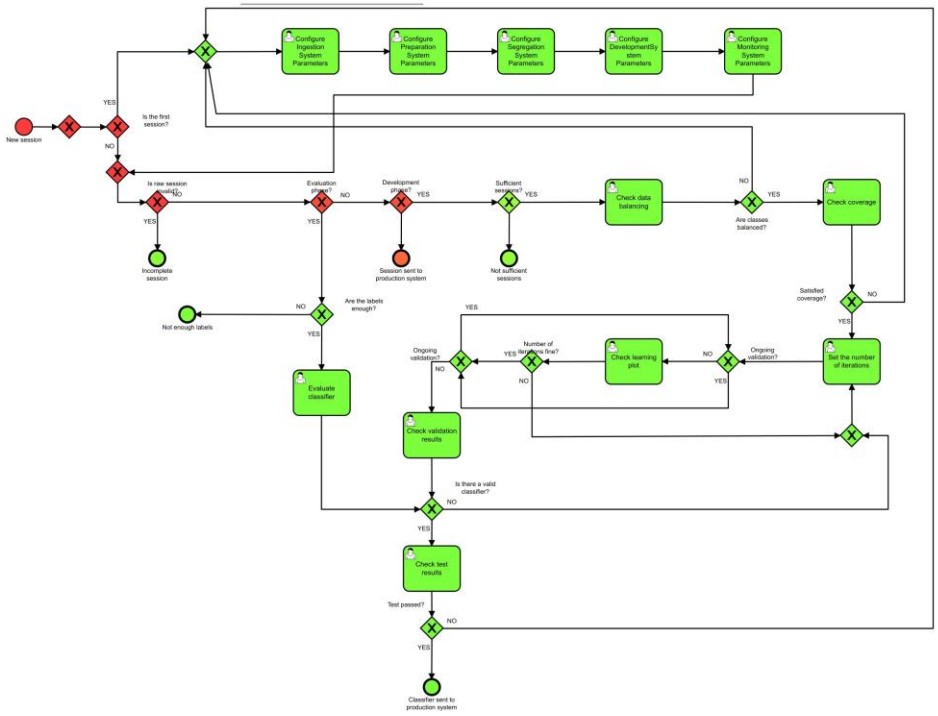
	Minimum	Maximum	Average
Process instance cycle times including off-timetable hours	0 seconds	16.3 minutes	0.5 seconds
Process instance cycle times excluding off-timetable hours	0 seconds	16.3 minutes	0.5 seconds
Process instance costs	0 EUR	0 EUR	0 EUR

Activity Durations, Costs, Waiting times, Deviations from Thresholds

Name		**Waiting time**			**Duration**			**Duration over threshold**			**Cost**			**Cost over threshold**		
	Count	Min	Avg	Max	Min	Avg	Max	Min	Avg	Max	Min	Avg	Max	Min	Avg	Max
Check coverage	2	0 s	0 s	0 s	6.7 s	6.9 s	7.2 s	0 s	0 s	0 s	0	0	0	0	0	0
Check data balancing	11	0 s	0 s	0 s	5.2 s	5.5 s	5.7 s	0 s	0 s	0 s	0	0	0	0	0	0
Check learning plot	32	0 s	0 s	0 s	9.6 s	10 s	10.5 s	0 s	0 s	0 s	0	0	0	0	0	0
Check test results	7	0 s	0 s	0 s	10.4 s	10.6 s	10.9 s	0 s	0 s	0 s	0	0	0	0	0	0
Check validation results	34	0 s	0 s	0 s	25.9 s	27.4 s	28.3 s	0 s	0 s	0 s	0	0	0	0	0	0
Configure DevelopmentSystem Parameters	15	0 s	0 s	0 s	26 s	27.4 s	28.5 s	0 s	0 s	0 s	0	0	0	0	0	0
Configure Ingestion System Parameters	15	0 s	0 s	0 s	7.7 s	8 s	8.4 s	0 s	0 s	0 s	0	0	0	0	0	0
Configure Monitoring System Parameters	15	0 s	0 s	0 s	28.2 s	29.5 s	30.6 s	0 s	0 s	0 s	0	0	0	0	0	0
Configure Preparation System Parameters	15	0 s	0 s	0 s	2.3 m	2.4 m	2.5 m	0 s	0 s	0 s	0	0	0	0	0	0
Configure Segregation System Parameters	15	0 s	0 s	0 s	19.1 s	19.9 s	20.9 s	0 s	0 s	0 s	0	0	0	0	0	0
Evaluate classifier	6	0 s	0 s	0 s	13.8 s	14.2 s	14.6 s	0 s	0 s	0 s	0	0	0	0	0	0
Set the number of iterations	58	0 s	0 s	0 s	8 s	8.4 s	8.8 s	0 s	0 s	0 s	0	0	0	0	0	0

Heatmap

Heatmap based on Counts

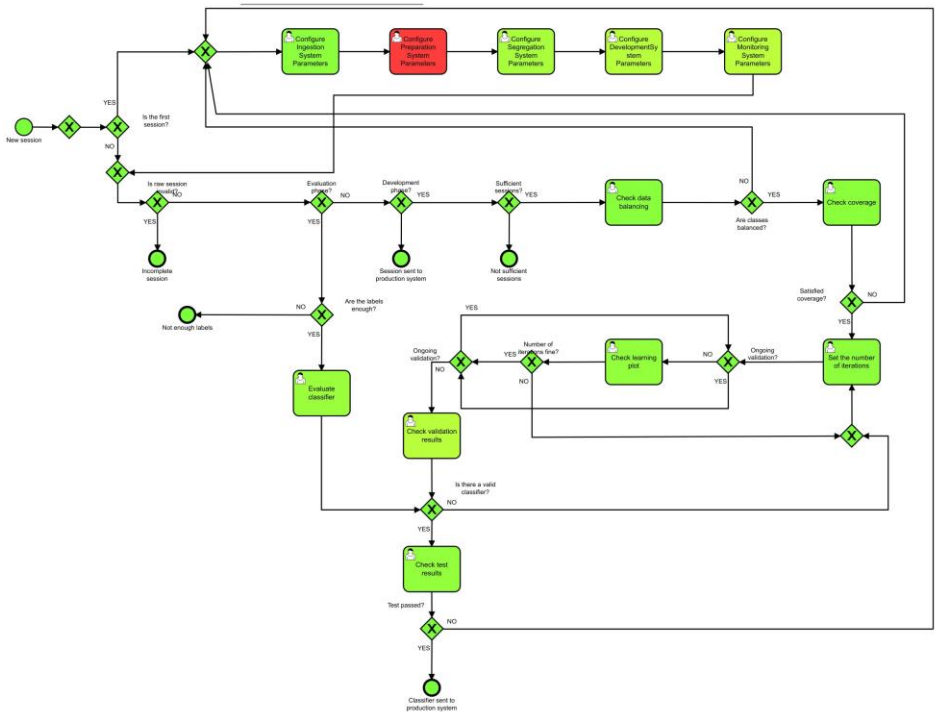


Legend

Color	Value
	0
	1112
	2224
	3337
	4449
	5561
	6673
	7786
	8898
	10010

Heatmap

Heatmap based on Durations



Legend

Color	Value
	0 s
	16 s
	32 s
	49 s
	1.1 m
	1.4 m
	1.6 m
	1.9 m
	2.2 m
	2.4 m

The first heatmap shows that the region with most iterations ends at ‘Development phase? No’, and that is expected as most of the sessions go to the production system (the proportion is 1000 production, 100 development and 10 evaluation).

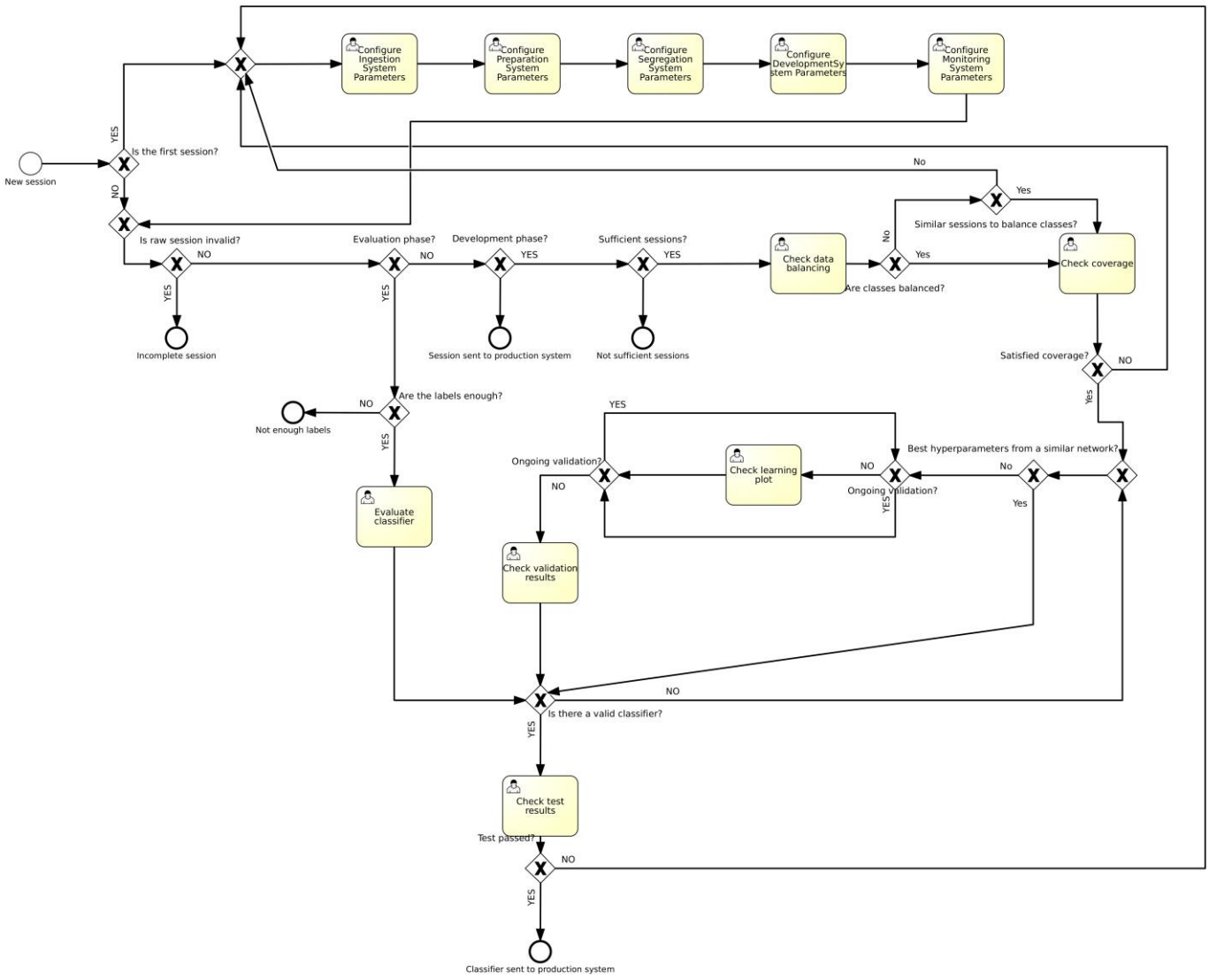
The second heatmap reported above shows that the ‘Configure Preparation System Parameters’ is the costliest task, and that has to do with the great number of features to analyze.

The simulation results highlight a duration of 3.6 weeks, with an average process instance cycle time of 0.5 seconds and 16.3 minutes at maximum.

TO-BE MODEL (All)

For the to-be modeling we added the following 3 improvements:

- handoff level, we assumed to have access to some sessions from a network like ours, to reduce the number of sessions sent to reconfiguration because of class balance.
- service level, we assumed to have access to some hyperparameters from networks like ours, to reduce the number of grid searches performed.
- task level, we reduce the cognitive effort in "Set the number of iterations” human task, replacing it with a script task giving a default setting according to a threshold to overcome. Doing so, we saved a cost of 8.36.



Simulation Results

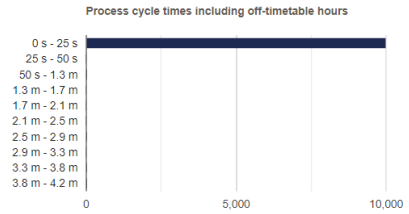
General information

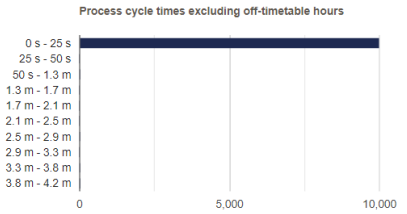
Completed process instances 10000

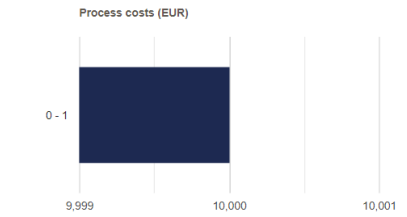
Total cost 0 EUR

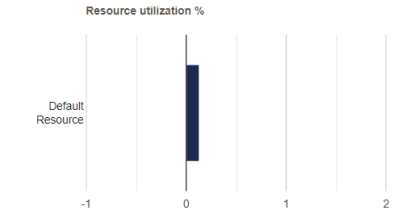
Total simulation time 3.6 weeks

Charts









Scenario Statistics

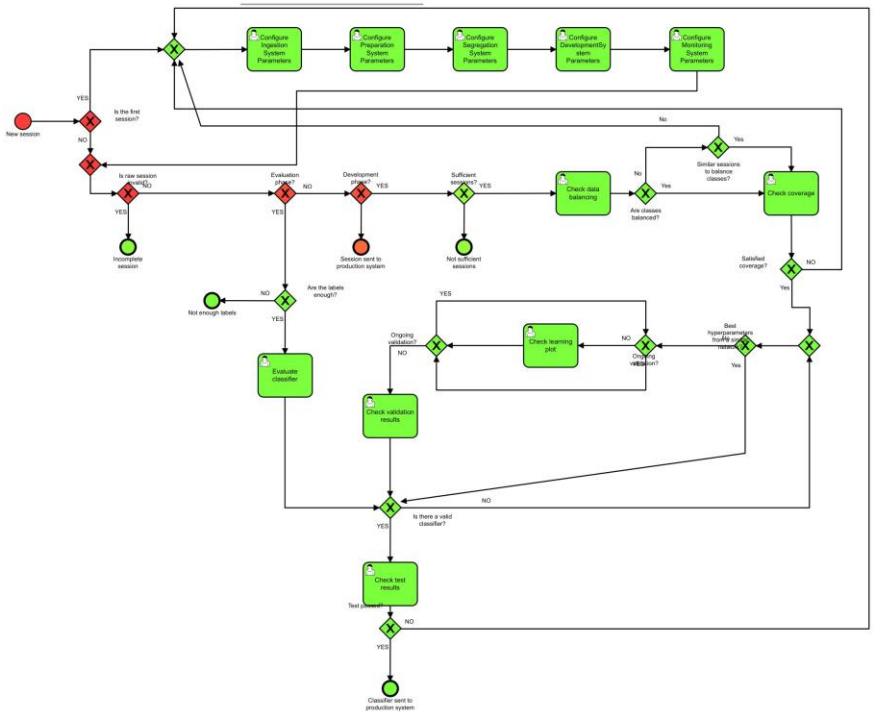
	Minimum	Maximum	Average
Process instance cycle times including off-timetable hours	0 seconds	4.1 minutes	0.3 seconds
Process instance cycle times excluding off-timetable hours	0 seconds	4.1 minutes	0.3 seconds
Process instance costs	0 EUR	0 EUR	0 EUR

Activity Durations, Costs, Waiting times, Deviations from Thresholds

Name		**Waiting time**			**Duration**			**Duration over threshold**			**Cost**			**Cost over threshold**		
	Count	Min	Avg	Max	Min	Avg	Max	Min	Avg	Max	Min	Avg	Max	Min	Avg	Max
Check coverage	4	0 s	0 s	0 s	6.7 s	6.9 s	7.1 s	0 s	0 s	0 s	0	0	0	0	0	0
Check data balancing	6	0 s	0 s	0 s	5.4 s	5.5 s	5.6 s	0 s	0 s	0 s	0	0	0	0	0	0
Check learning plot	3	0 s	0 s	0 s	9.8 s	9.9 s	10.1 s	0 s	0 s	0 s	0	0	0	0	0	0
Check test results	3	0 s	0 s	0 s	10.1 s	10.4 s	10.9 s	0 s	0 s	0 s	0	0	0	0	0	0
Check validation results	5	0 s	0 s	0 s	26 s	26.8 s	27.9 s	0 s	0 s	0 s	0	0	0	0	0	0
Configure DevelopmentSystem Parameters	11	0 s	0 s	0 s	25.9 s	27.1 s	28.2 s	0 s	0 s	0 s	0	0	0	0	0	0
Configure Ingestion System Parameters	11	0 s	0 s	0 s	7.6 s	8 s	8.3 s	0 s	0 s	0 s	0	0	0	0	0	0
Configure Monitoring System Parameters	11	0 s	0 s	0 s	27.8 s	29.1 s	30.6 s	0 s	0 s	0 s	0	0	0	0	0	0
Configure Preparation System Parameters	11	0 s	0 s	0 s	2.3 m	2.4 m	2.5 m	0 s	0 s	0 s	0	0	0	0	0	0
Configure Segregation System Parameters	11	0 s	0 s	0 s	19.1 s	20.1 s	21 s	0 s	0 s	0 s	0	0	0	0	0	0
Evaluate classifier	3	0 s	0 s	0 s	14.3 s	14.5 s	14.8 s	0 s	0 s	0 s	0	0	0	0	0	0

Heatmap

Heatmap based on Counts

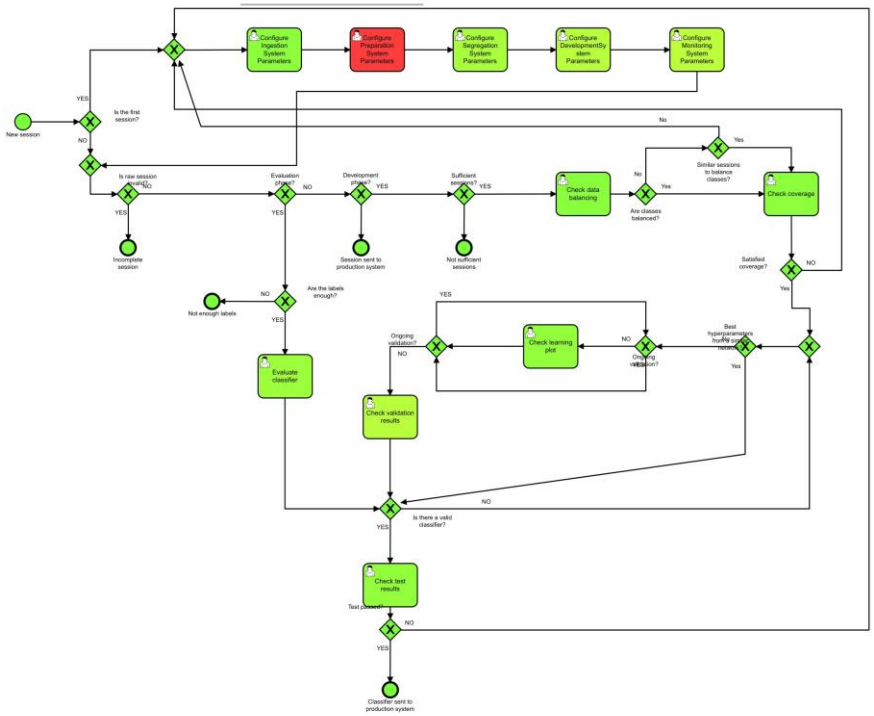


Legend

Color	Value
	0
	1112
	2224
	3335
	4447
	5559
	6671
	7782
	8894
	10006

Heatmap

Heatmap based on Durations



Legend

Color	Value
	0 s
	16 s
	32 s
	48 s
	1.1 m
	1.3 m
	1.6 m
	1.9 m
	2.1 m
	2.4 m

AS-IS & TO-BE COMPARATIVE DISCUSSION (All)

Comparing the AS-IS and TO-BE results, we can notice that the process instance cycle time decreases from 16.3 minutes to 4.1 minutes (average from 0.5s to 0.3s) and the total simulation time stays constant at 3.6 weeks.

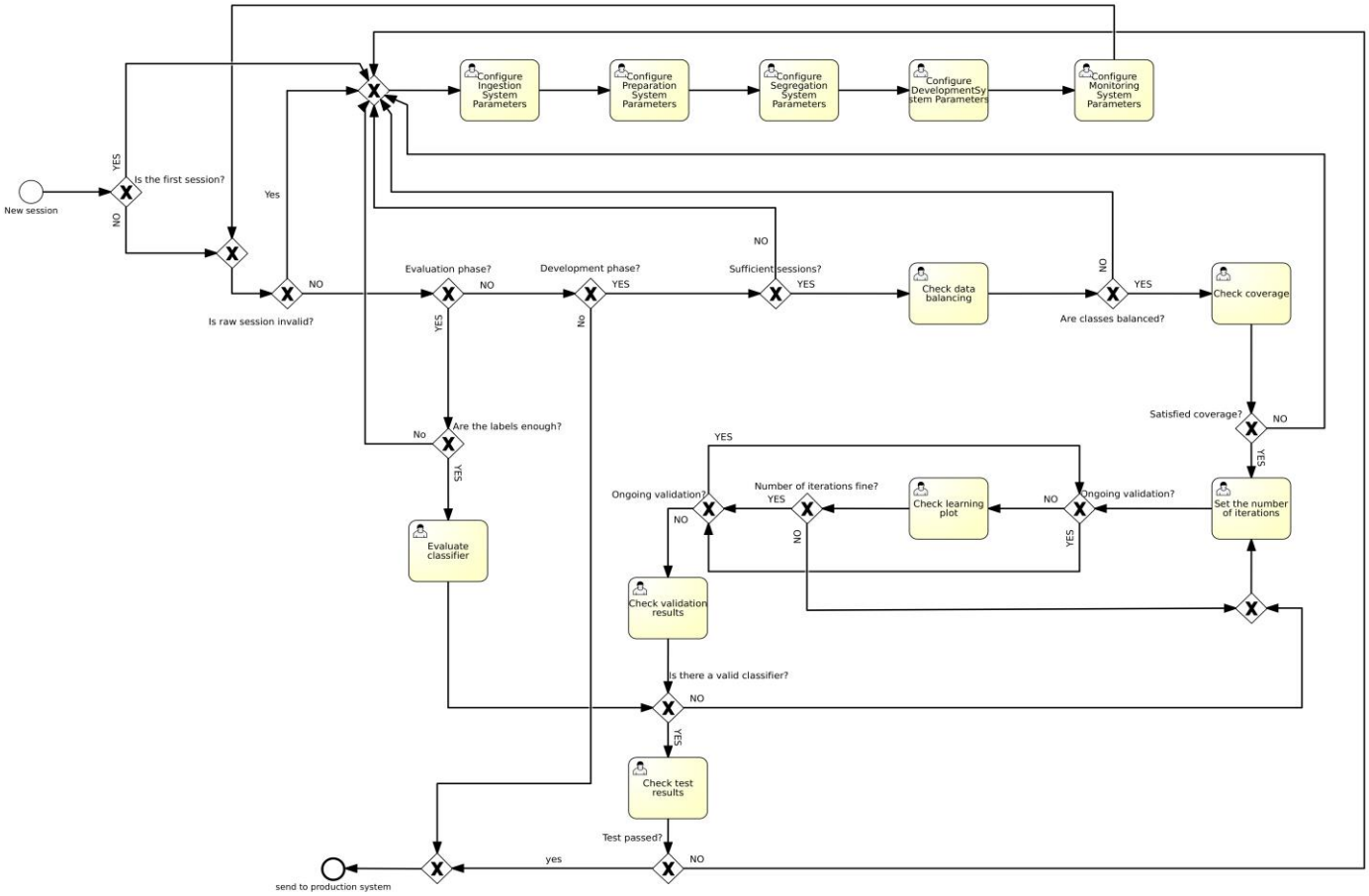
Simulating many times, we obtained different process instance cycle times but the to-be always lower than the as-is scenario.

Comparing the duration heatmaps, we notice that the “Configure Preparation System Parameters” task is still the most expensive one as expected.

The counts heatmaps are almost the same and that’s expected as every simulation starts with 10000 tokens and the overall model is quite the same, except for the added improvements.

PROCESS MINING

Normative Model (Basma)



Starting from the As-Is Model, we built the collapsed as-is (normative) model characterized by a unique start and a unique end event.

In particular, the end points “Incomplete session”, “Not enough labels” and “Not sufficient sessions” were substituted with arrows going back to the configuration step.

Instead, the end point “Session sent to production system” goes directly to the unique end point which is “send to production system”.

For the BIMP simulation, a default cost of 1 euro and a default duration of 1 sec were assigned to each task. Gateways were assigned a probability of 50% each, 10 resources were added, and 100 input tokens were used.

Simulation Results

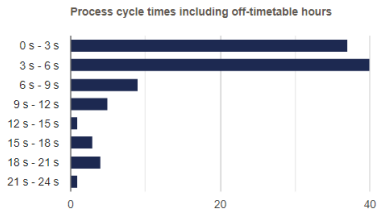
General information

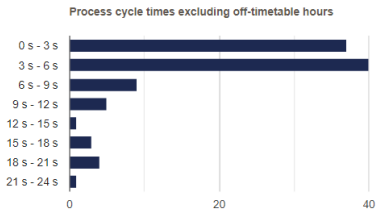
Completed process instances 100

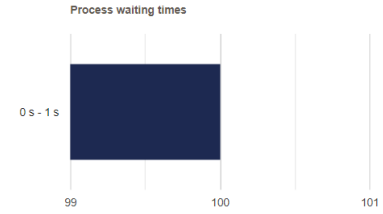
Total cost 468 EUR

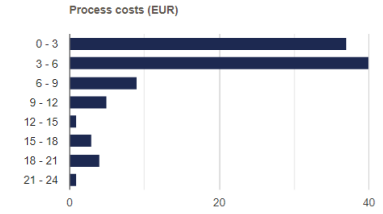
Total simulation time 1.1 hours

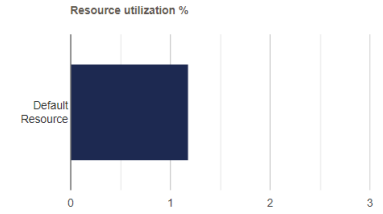
Charts











Scenario Statistics

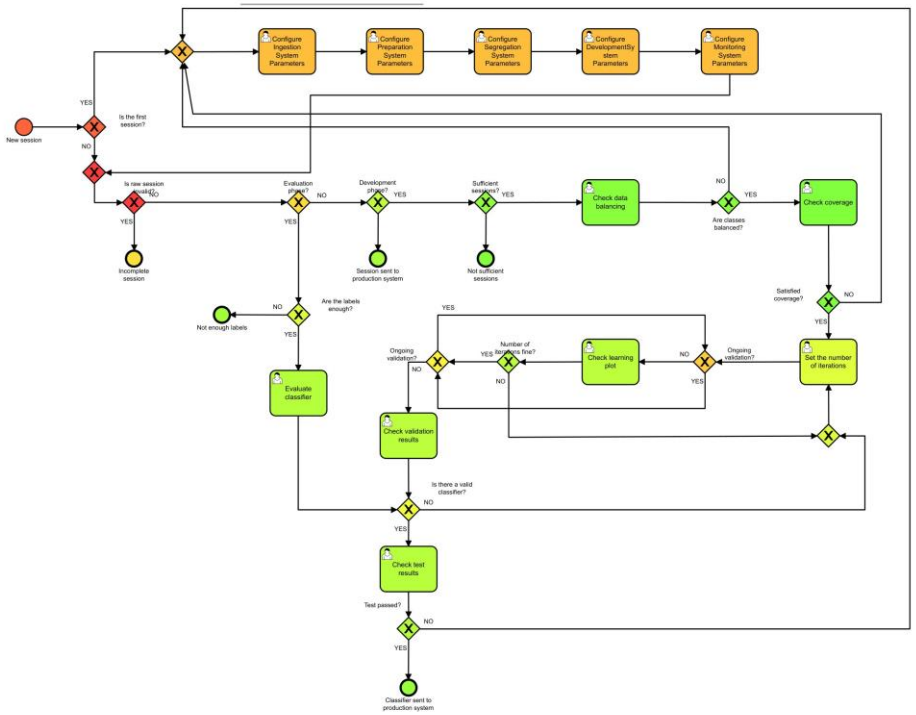
	Minimum	Maximum	Average
Process instance cycle times including off-timetable hours	0 seconds	21 seconds	4.7 seconds
Process instance cycle times excluding off-timetable hours	0 seconds	21 seconds	4.7 seconds
Process instance costs	0 EUR	21 EUR	4.7 EUR

Activity Durations, Costs, Waiting times, Deviations from Thresholds

Name		**Waiting time**			**Duration**			**Duration over threshold**			**Cost**			**Cost over threshold**		
	Count	Min	Avg	Max	Min	Avg	Max	Min	Avg	Max	Min	Avg	Max	Min	Avg	Max
Check coverage	2	0 s	0 s	0 s	1 s	1 s	1 s	0 s	0 s	0 s	1	1	1	0	0	0
Check data balancing	5	0 s	0 s	0 s	1 s	1 s	1 s	0 s	0 s	0 s	1	1	1	0	0	0
Check learning plot	23	0 s	0 s	0 s	1 s	1 s	1 s	0 s	0 s	0 s	1	1	1	0	0	0
Check test results	21	0 s	0 s	0 s	1 s	1 s	1 s	0 s	0 s	0 s	1	1	1	0	0	0
Check validation results	22	0 s	0 s	0 s	1 s	1 s	1 s	0 s	0 s	0 s	1	1	1	0	0	0
Configure DevelopmentSystem Parameters	68	0 s	0 s	0 s	1 s	1 s	1 s	0 s	0 s	0 s	1	1	1	0	0	0
Configure Ingestion System Parameters	68	0 s	0 s	0 s	1 s	1 s	1 s	0 s	0 s	0 s	1	1	1	0	0	0
Configure Monitoring System Parameters	68	0 s	0 s	0 s	1 s	1 s	1 s	0 s	0 s	0 s	1	1	1	0	0	0
Configure Preparation System Parameters	68	0 s	0 s	0 s	1 s	1 s	1 s	0 s	0 s	0 s	1	1	1	0	0	0
Configure Segregation System Parameters	68	0 s	0 s	0 s	1 s	1 s	1 s	0 s	0 s	0 s	1	1	1	0	0	0
Evaluate classifier	20	0 s	0 s	0 s	1 s	1 s	1 s	0 s	0 s	0 s	1	1	1	0	0	0
Set the number of iterations	35	0 s	0 s	0 s	1 s	1 s	1 s	0 s	0 s	0 s	1	1	1	0	0	0

Heatmap

Heatmap based on Counts

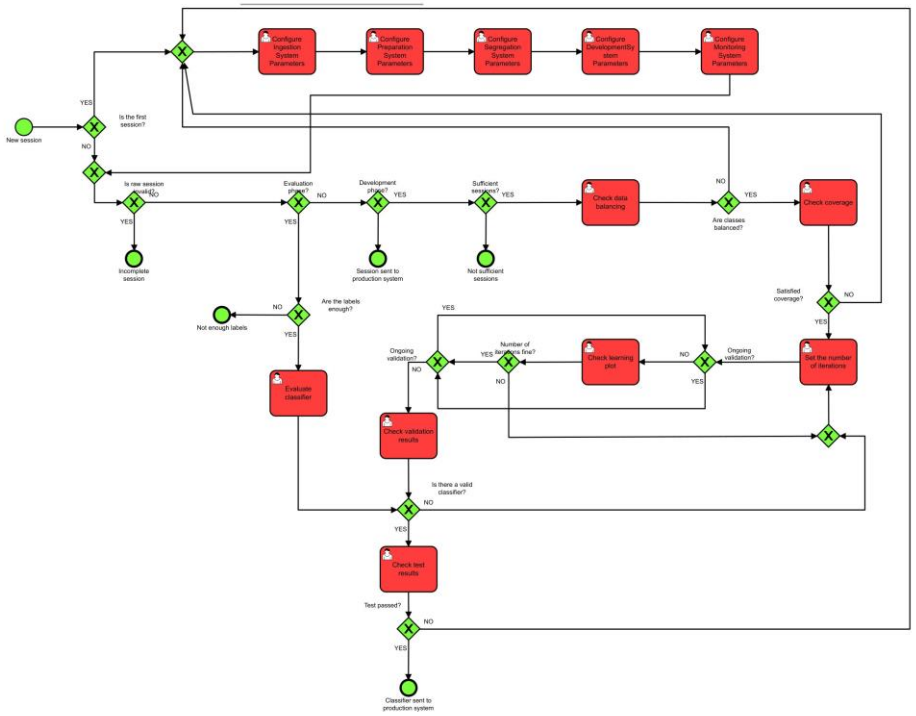


Legend

Color	Value
	0
	13
	25
	38
	51
	63
	76
	89
	101
	114

Heatmap

Heatmap based on Durations



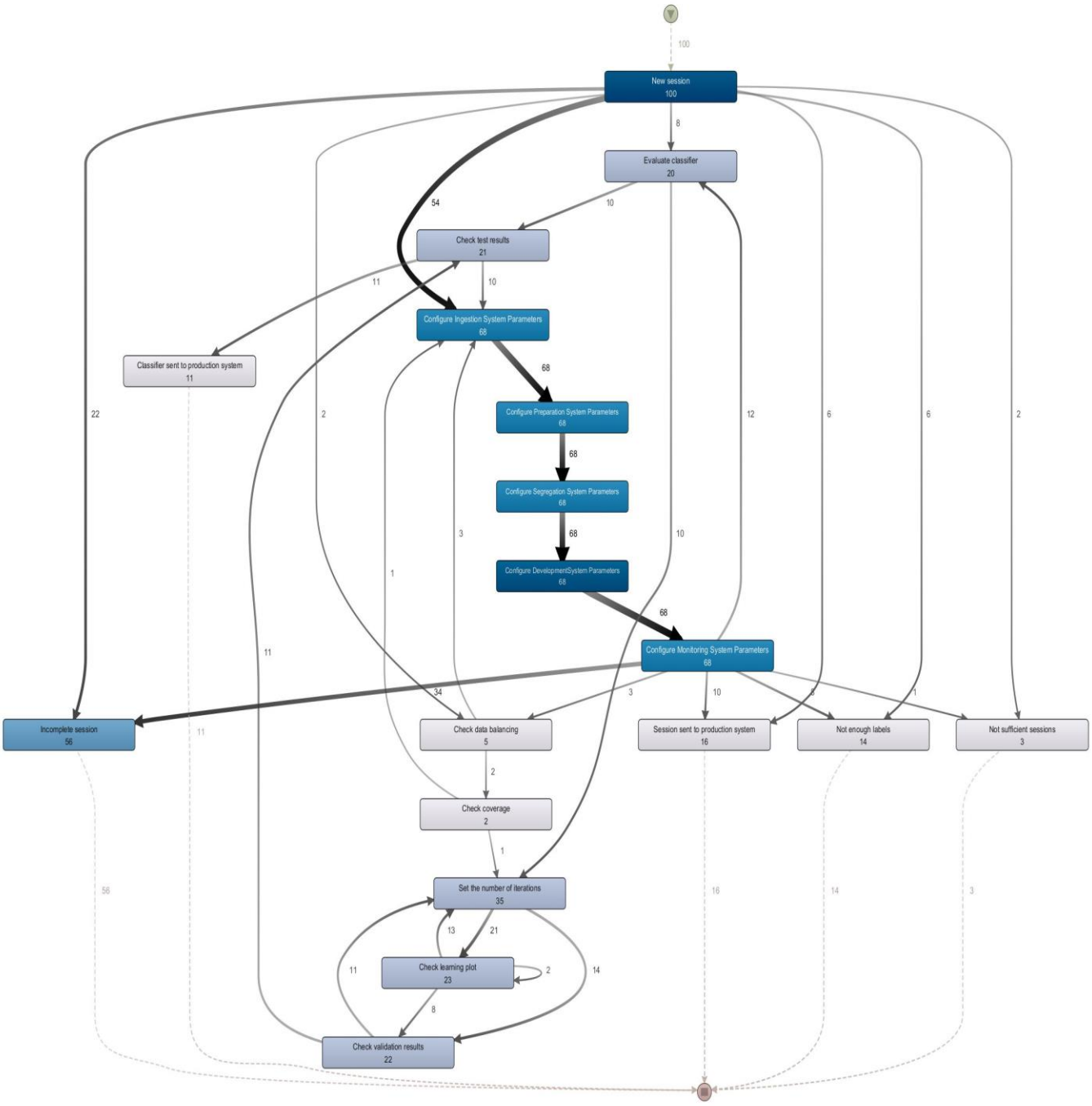
Legend

Color	Value
	0 s
	0 s
	0 s
	0 s
	0 s
	1 s
	1 s
	1 s
	1 s
	1 s

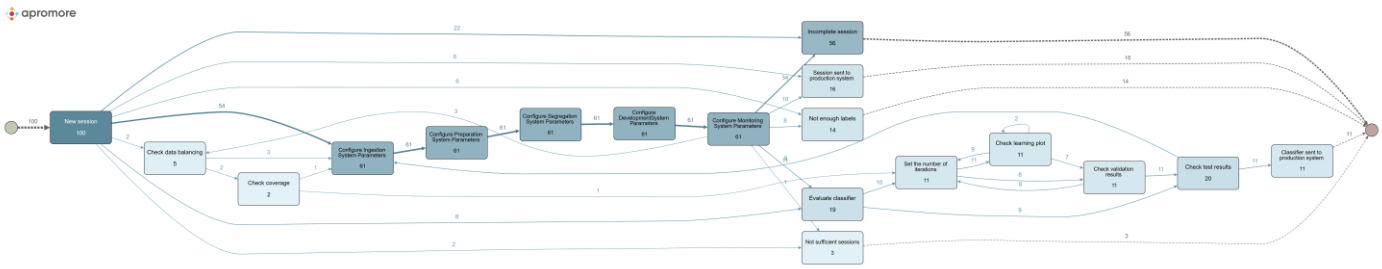
Transition map – Disco vs Apromore(Daniele Laporta)

We exported the generated mxml log of the simulated normative model and we uploaded it on Disco, obtaining the following transition map. We used the same log file on Apromore.

Disco Transition map



Apromore Transition map



Differences between transition maps from Disco and Apromore

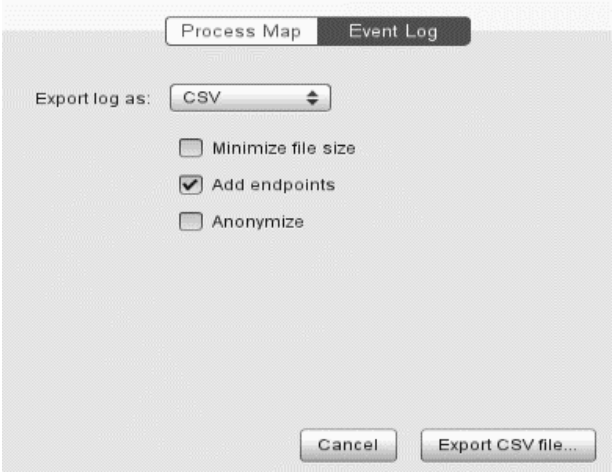
These transition maps reflect the workflow of our company but not the workloads because for the normative model simulated, we used a value of 50% for every exclusive gateway, modifying the normal behaviour of the company.

Comparing the two transition maps, with all the possible paths showed, we can notice that on Disco there are higher number of tokens and that's due to the fact that it considers tokens coming back from loops. Instead on Apromore the token number visualised on a link is just the number of the token divided by the probability of that token going through that link.

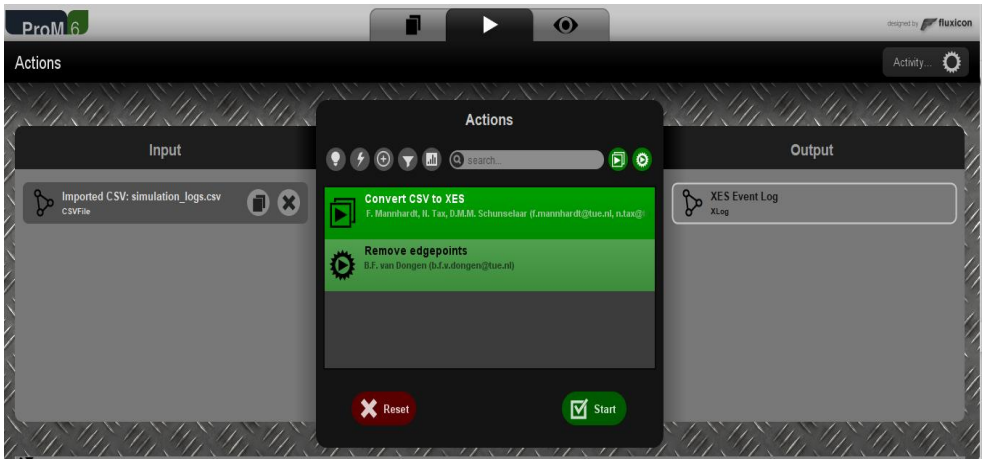
Apromore doesn't emphasize the edges that carry more tokens as Disco does.

Csv to Xes Conversion on Disco + ProM

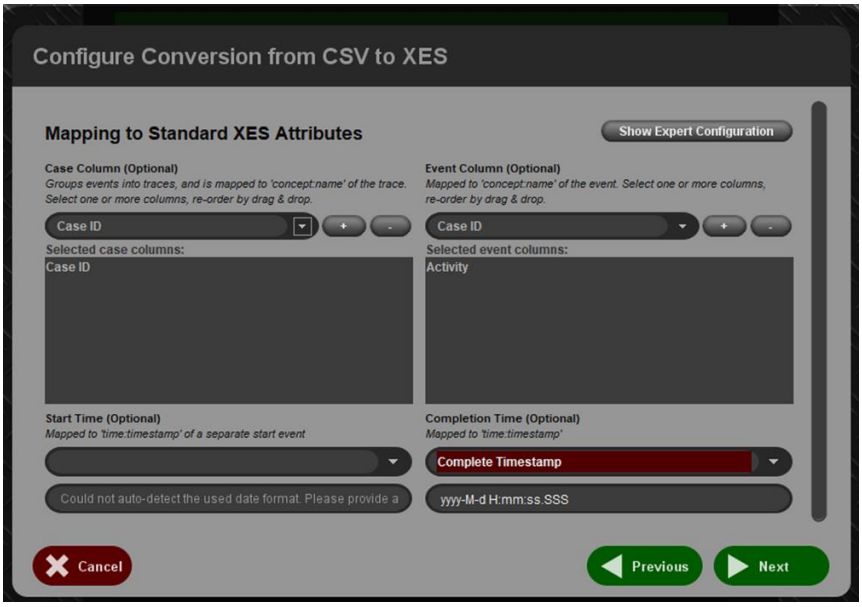
Using Disco software, we exported the transition map as a csv file to be used on ProM.



Furthermore, we converted this csv file using the specific plugin "Convert CSV to XES" of the proM software and we got the xes file to be used as a log in proM.

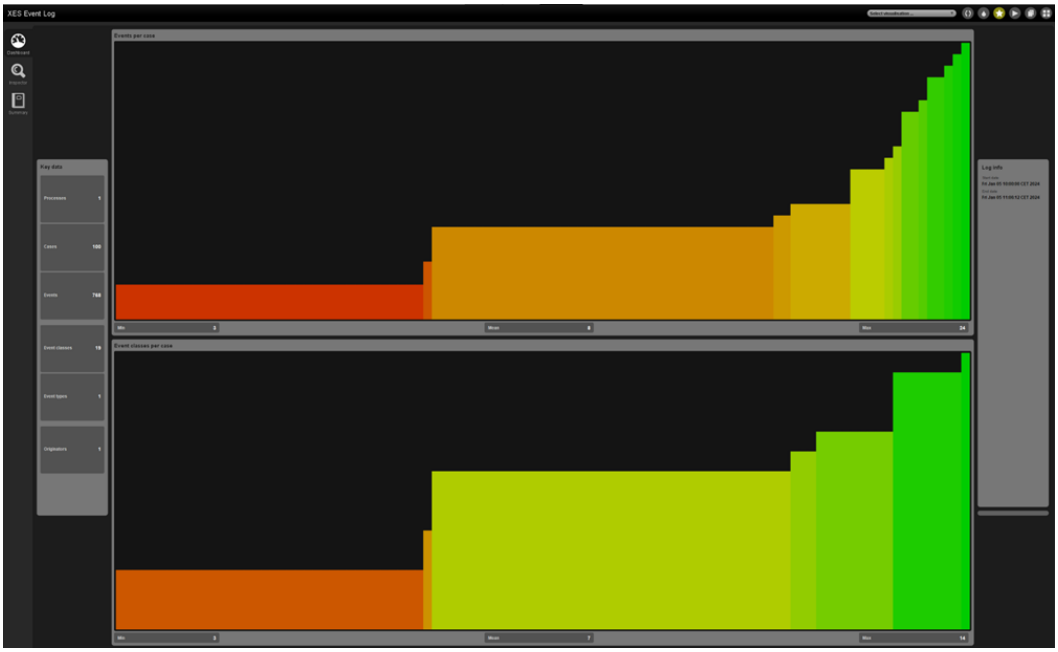


We specified the case column with "Case ID" and the Completion Time with "Complete Timestamp", perfectly recognized by the software with the correct date format.



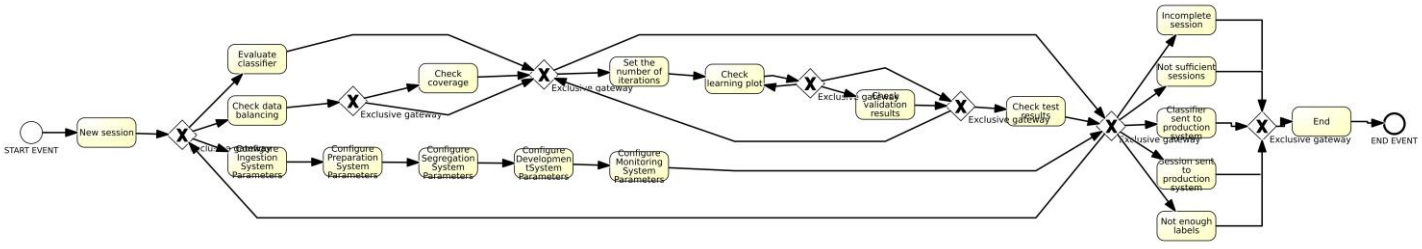
The other additional conversion settings stayed as default. We could choose how the software would handle errors, stopping the conversion when an error is found or continuing it, but this is secondary as no errors are found during the conversion.

So this is the Xes obtained:



Then we used it both on ProM and Apromore to mine a BPMN model.

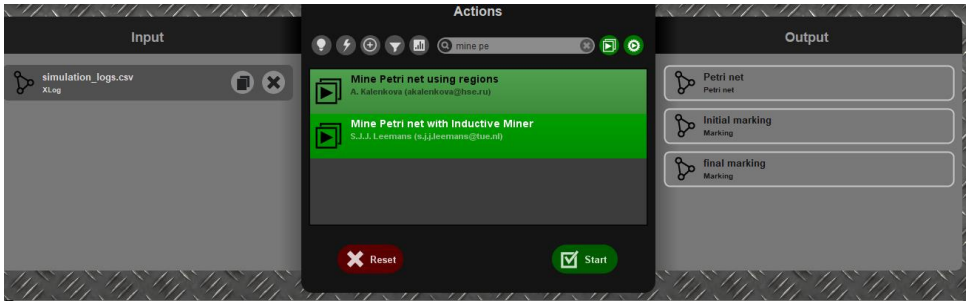
BPMN mining from original logs – ProM(Basma, Francesco Zingariello)



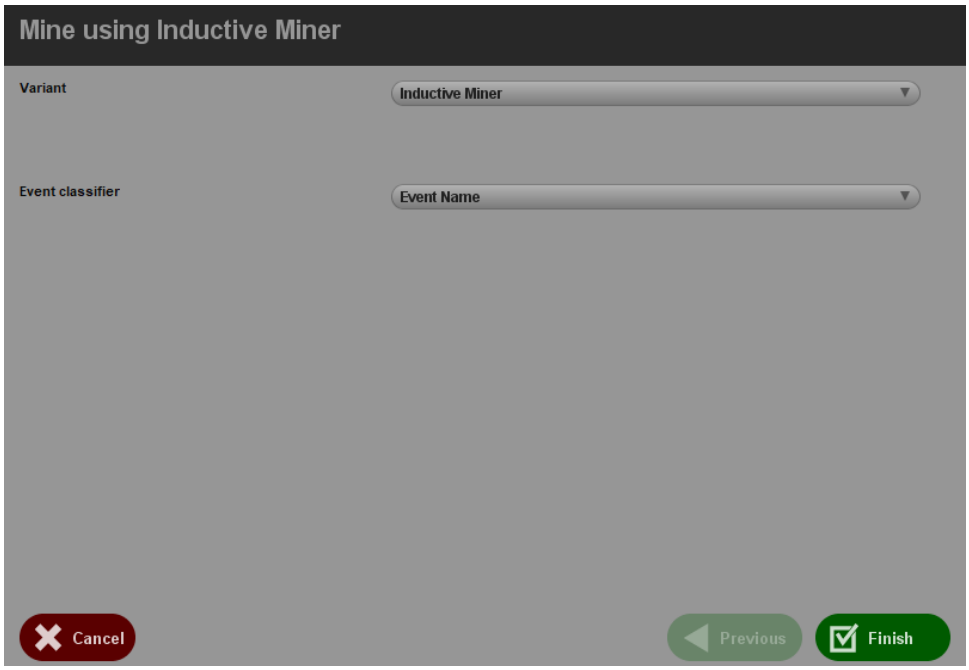
We generated this BPMN model from ProM by following these steps:

- We imported the Xes file in ProM and apply “Mine Petri net with inductive miner” plugin on it.

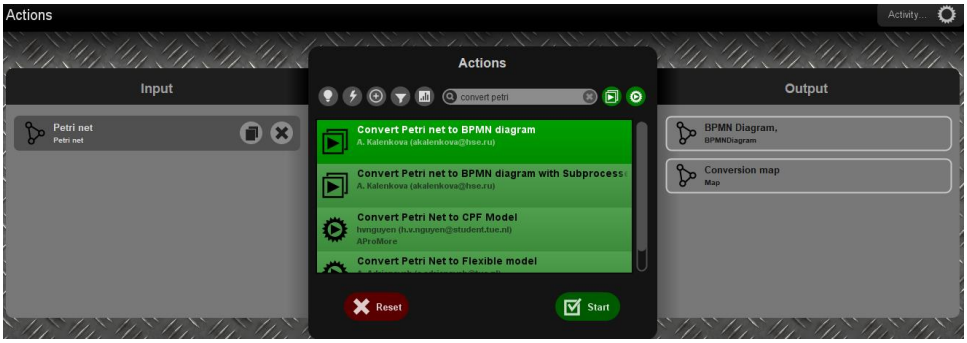
  (We tried using “BPMN Miner” plugin but it didn’t work, so we found another plugin)



- We chose “Inductive miner” to generate a Petri Net representing the event logs used as input.



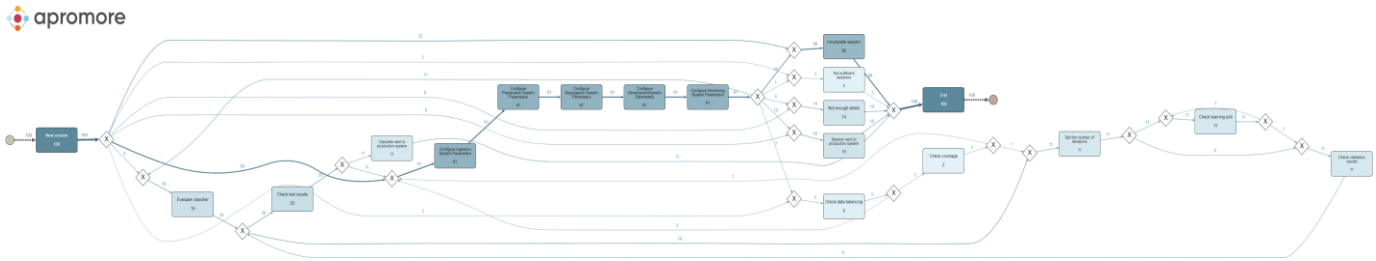
- Then by applying “Convert Petri Net to BPMN diagram” we got the BPMN.



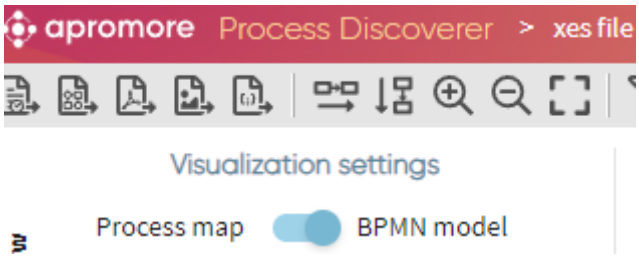
But it didn’t work correctly, we needed to modify it on Signavio as a gateway was missing. In particular, the very last arrows pointed directly to the task “End” (added automatically by Disco) and this causes an error on Signavio too. So we added an exclusive gateway to aggregate them.

When doing the same steps on the modified log, we discover another plugin, called “Convert Petri net to BPMN” by Raffaele Conforti we obtained the right BPMN, without need of any gateway adding.

BPMN mining from original logs - Apromore



We generated this BPMN model from Apromore by uploading the event logs Xes file we got from ProM in the previous steps and then converting it to BPMN.



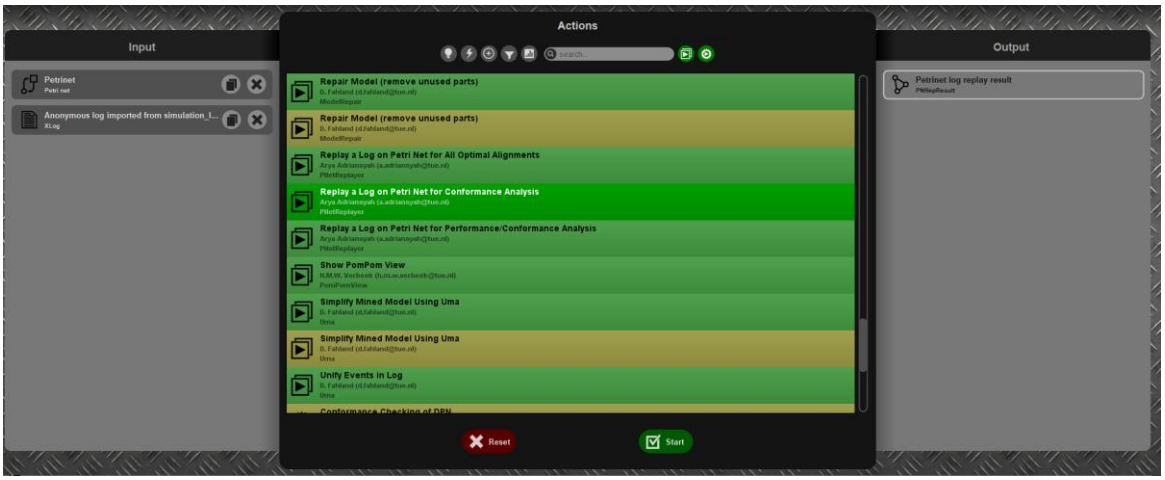
Differences between the BPMN models generated by ProM and Apromore

The BPMN models were generated from the same Xes file which was extracted from BIMP as an xmlx file.

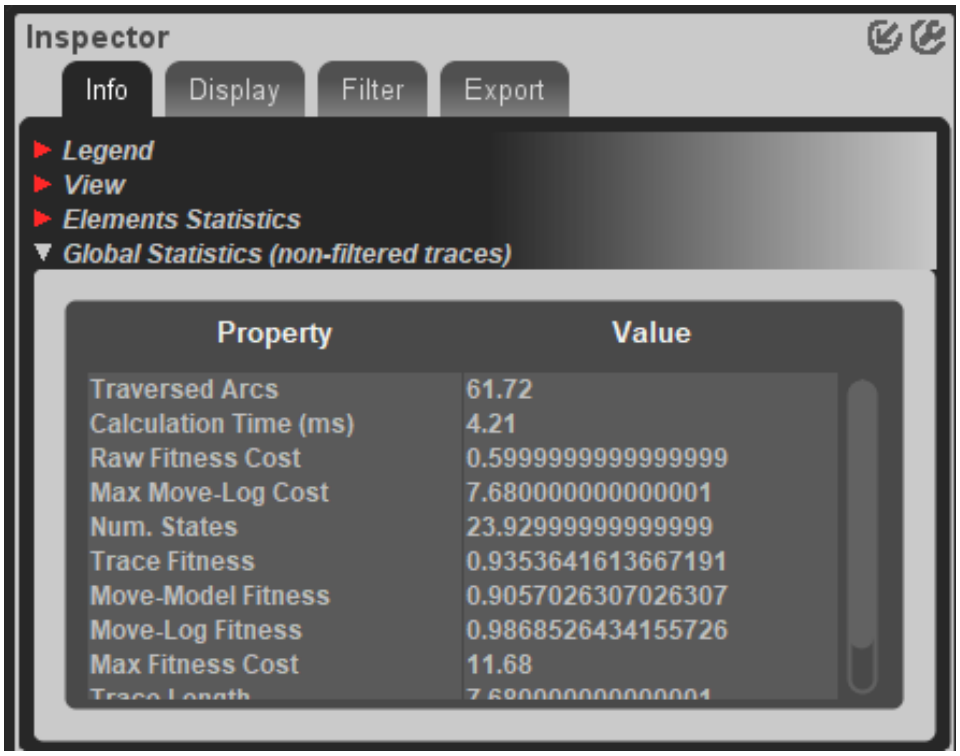
We can notice that the number of the gateways generated from Apromore are more than from ProM.

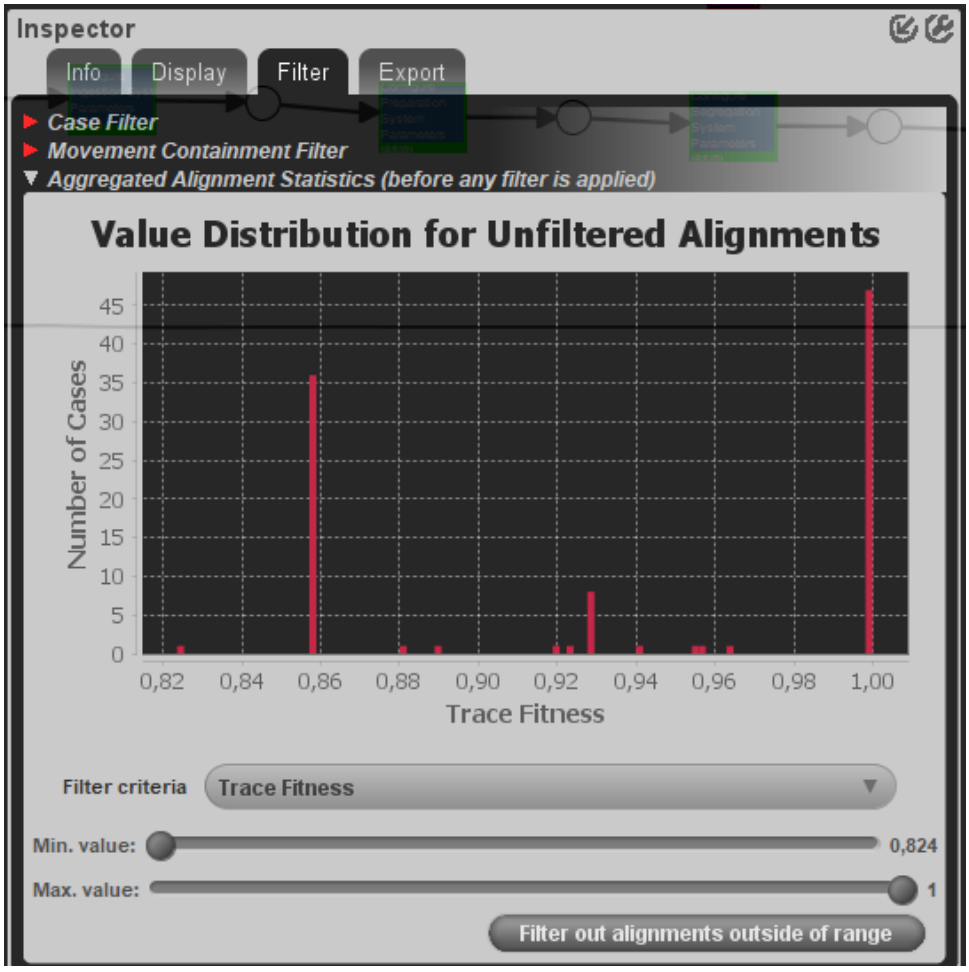
Conformance checking of ProM mined BPMN (quality dimensions) (Francesco Zingariello, Basma Adawy)

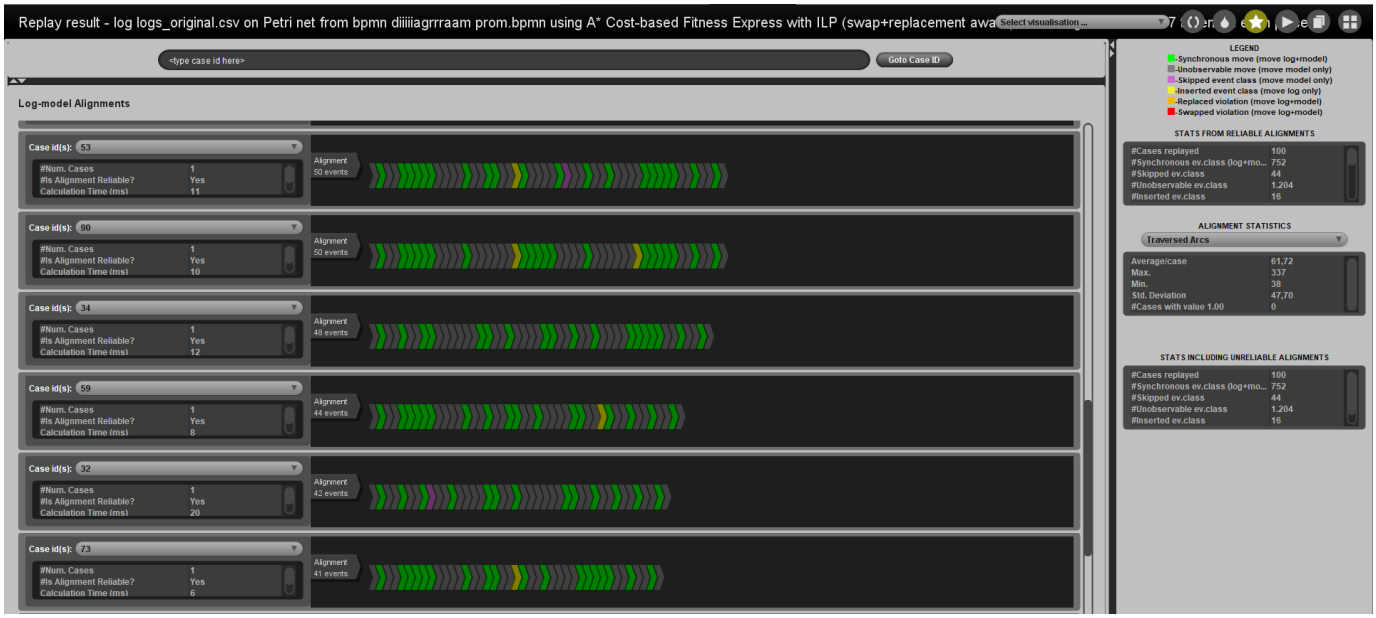
To do the conformance checking we used ProM, giving as input the Petrinet and the Xes log file.



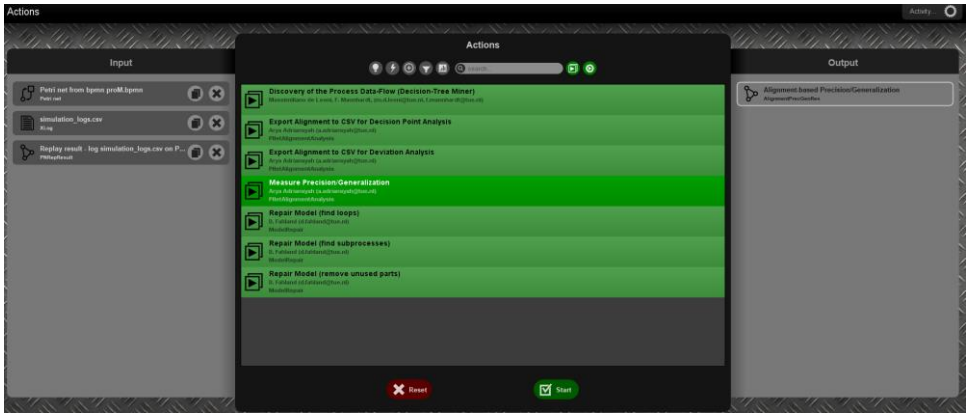
Fitness


Inspector
Info
Display
Filter
Export
Legend
View
Elements Statistics
Global Statistics (non-filtered traces)
Property
Value
Traversed Arcs
61.72
Calculation Time (ms)
4.21
Raw Fitness Cost
0.5999999999999999
Max Move-Log Cost
7.680000000000001
Num. States
23.92999999999999
Trace Fitness
0.9353641613667191
Move-Model Fitness
0.9057026307026307
Move-Log Fitness
0.9868526434155726
Max Fitness Cost
11.68


Inspector
Info
Display
Filter
Export
Case Filter
Movement Containment Filter
Aggregated Alignment Statistics (before any filter is applied)
Value Distribution for Unfiltered Alignments
Number of Cases
45
40
35
30
25
20
15
10
5
0
0,82
0,84
0,86
0,88
0,90
0,92
0,94
0,96
0,98
1,00
Trace Fitness
Filter criteria
Trace Fitness
Min. value:
0,824
Max. value:
1
Filter out alignments outside of range



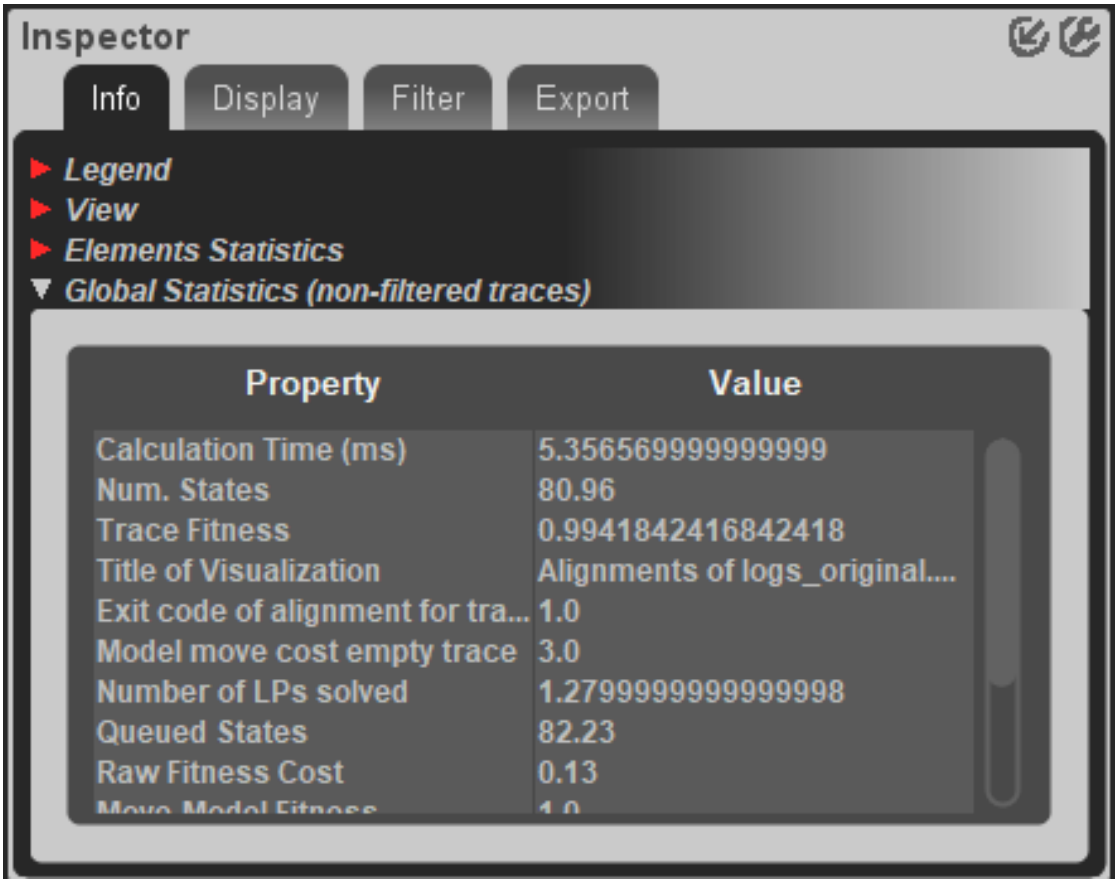
Precision & Generalization

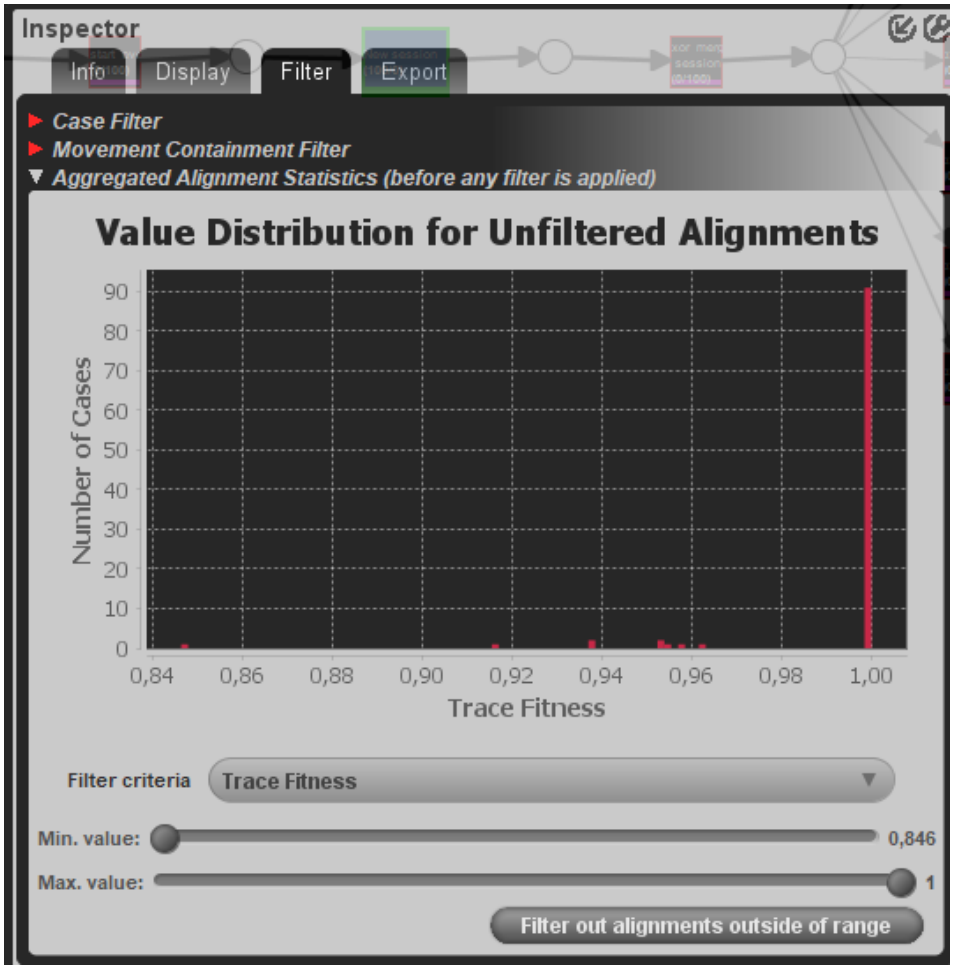


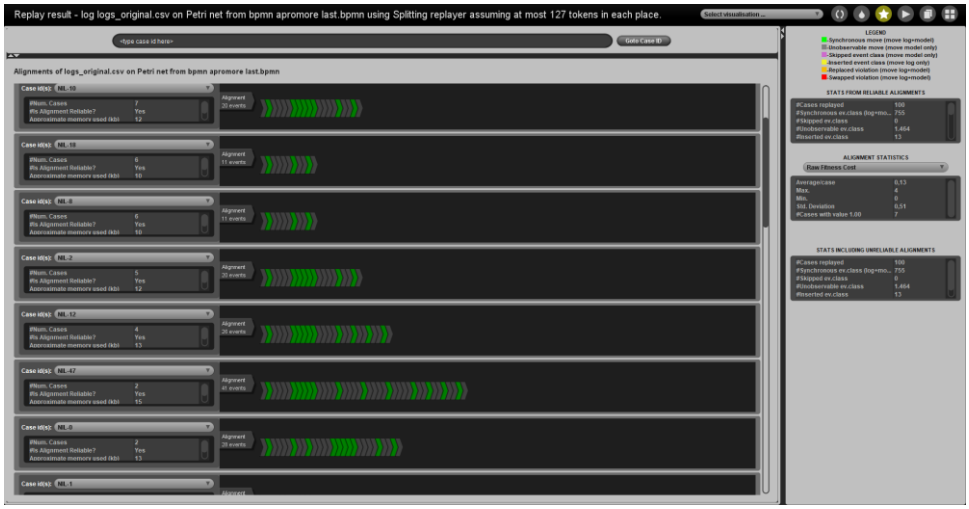
Simplicity

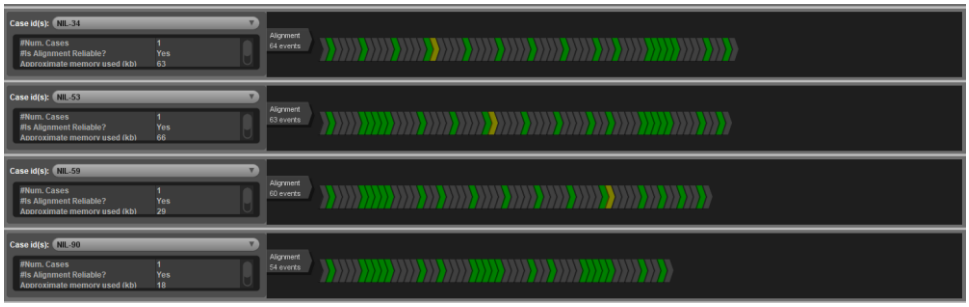
Simplicity: $\sum \#gateways + \#sequence\ flows + \#activities$ = 7 + 39 +19 = 65

Conformance checking of Apromore mined BPMN (quality dimensions) (Francesco Zingariello, Basma Adawy)


Inspector
Info
Display
Filter
Export
Legend
View
Elements Statistics
Global Statistics (non-filtered traces)
Property
Value
Calculation Time (ms)
5.356569999999999
Num. States
80.96
Trace Fitness
0.9941842416842418
Title of Visualization
Alignments of logs_original....
Exit code of alignment for tra...
1.0
Model move cost empty trace
3.0
Number of LPs solved
1.2799999999999998
Queued States
82.23
Raw Fitness Cost
0.13


Inspector
Info
Display
Filter
Export
Case Filter
Movement Containment Filter
Aggregated Alignment Statistics (before any filter is applied)
Value Distribution for Unfiltered Alignments
Number of Cases
90
80
70
60
50
40
30
20
10
0
0,84
0,86
0,88
0,90
0,92
0,94
0,96
0,98
1,00
Trace Fitness
Filter criteria
Trace Fitness
Min. value:
0,846
Max. value:
1
Filter out alignments outside of range





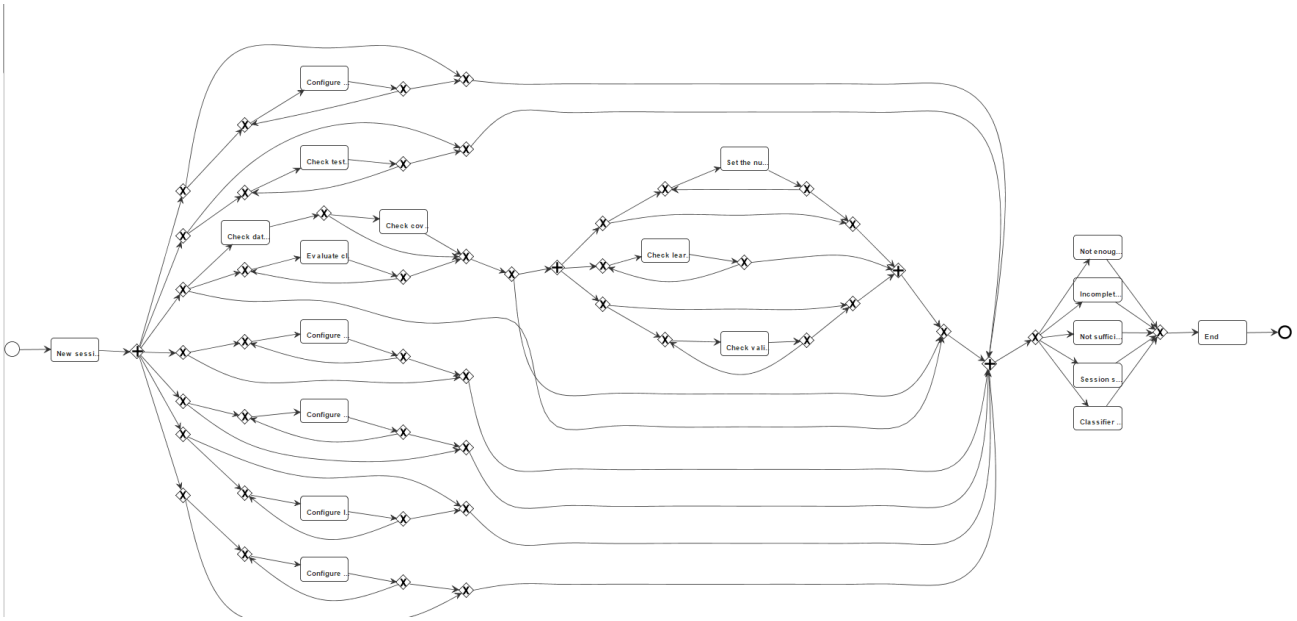
Simplicity: ∑ $\#gateways$ + $\#sequence\ flows$ + $\#activities$ = 19+ 52 +19 = 90

Edit of the CSV log file (Pietrangelo,Daniele Laporta)

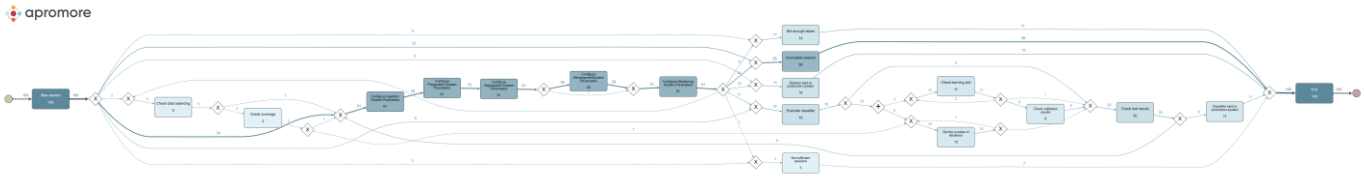
We edited the CSV log with Excel. We removed some tasks to 3 cases to obtain 3 realistic and useful violations to the model. We considered the following modifications:

- "Configure Development System Parameters": under the assumption that we have a cluster of similar customers' networks already configured with the best hyperparameters, we can skip this part of the configuration and use them for our system (records 61, 70 and 87 were removed).
- "Set the number of iterations": under the assumption that we have a script to set a default number of iterations in some cases, sometimes we can skip this activity (the first instances of records 59, 35 and 47 were removed).
- "Check validation results": under the assumption that we have a script to automatically choose the best classifier following certain rules and under certain conditions, sometimes we can skip this activity (the first instances of records 71, 32 and 73 were removed).

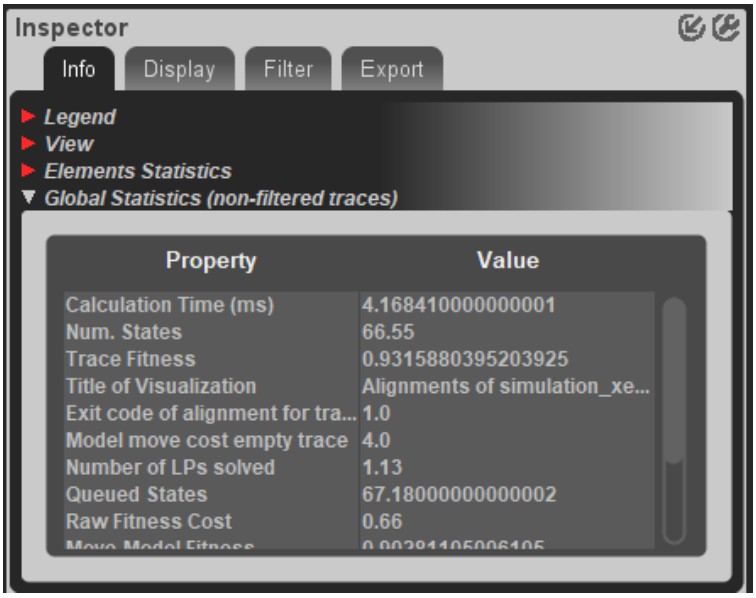
BPMN mining from modified logs – ProM(Francesco Zingariello, Daniele Laporta)

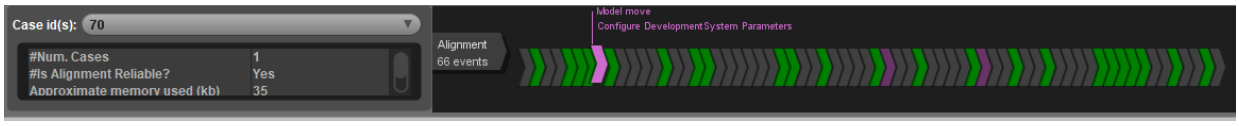


BPMN mining from modified logs - Apromore



Conformance checking of ProM mined BPMN (quality dimensions – modified log)(Pietrangelo,Daniele Laporta)





Precision/Generalization of simulation_xes_edited_z.csv to Petri net from bpmn diiiiiagrrraam prom.bpmn

Precision : 0,60966

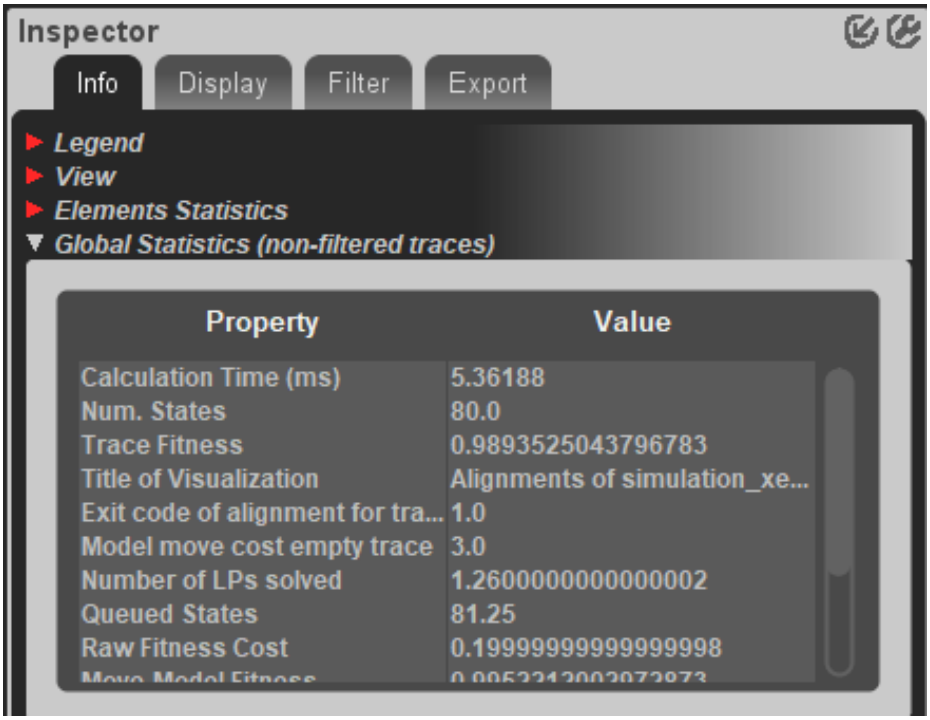
Generalization : 0,97250

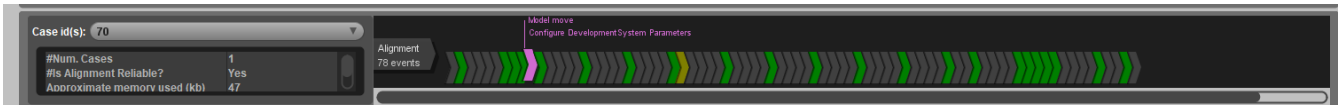
Simplicity: ∑ $\#gateways$ + $\#sequence\ flows$ + $\#activities$ = 48 + 95 + 19 = 162

The pictures show the violated cases, coherent with the modifications done. The fitness reflects the violations applied to the log.

We can notice that the precision increases because we added some violations in the modified log, so there's more extra behavior to control.

Conformance checking of Apromore mined BPMN (quality dimensions – modified log)





Precision/Generalization of simulation_xes_edited_z.csv to Petri net from bpmn apromore last.bpmn

Precision : 0,82966

Generalization : 0,98356

Simplicity: ∑ $\#gateways$ + $\#sequence\ flows$ + $\#activities$ = 22 + 59 + 19 = 100

Transition Maps – Disco vs Apromore (modified log)(All)

In the following transition maps we can notice that there's an arrow, with 3 tokens, that goes directly from the "Configure Segregation System" to the "Configure Monitoring System" and skipping the "Configure Development System", successfully confirming that the violations are correctly applied.