UNIVERSITÀ DI PISA

Msc. In Artificial Intelligence and Data Engineering

MOBILITY BEHAVIOR MONITORING

Process Mining and Intelligence Project

Fabio Malloggi

Lorenzo Tonelli

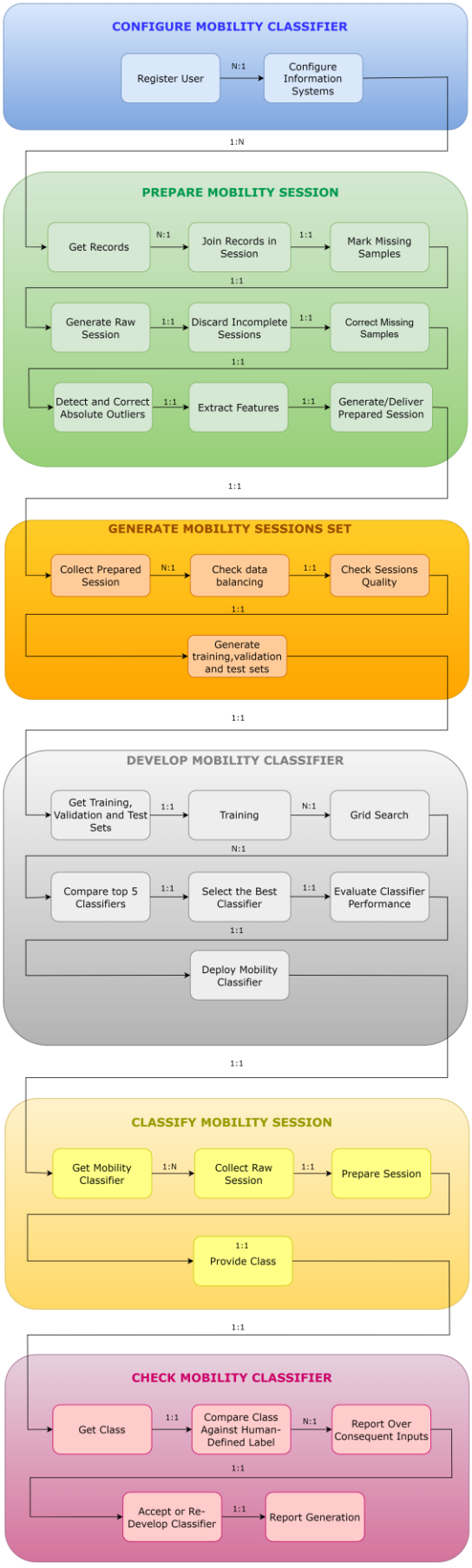
Ludovica Cocchella

Francesco Marabotto

Academic Year 2022/2023

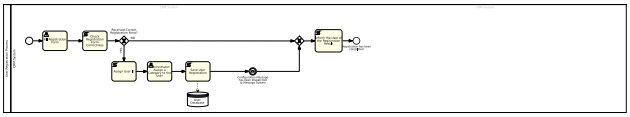
Contents

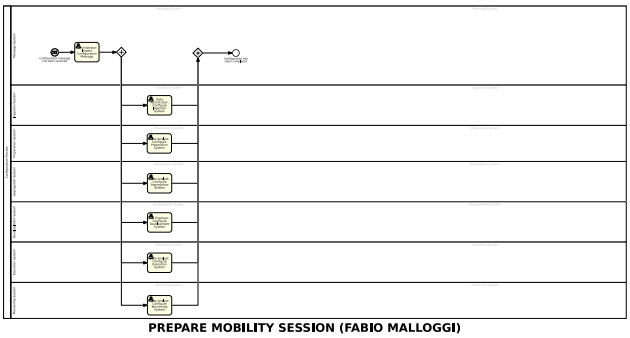
1. PROCESS LANDSCAPE



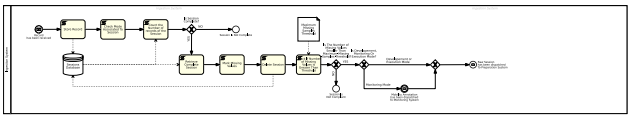
2. BPMN Service Level Diagram

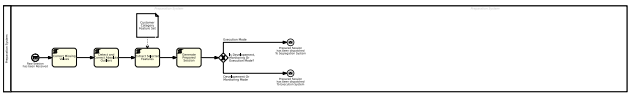
CONFIGURE MOBILITY CLASSIFIER (TUTTI)



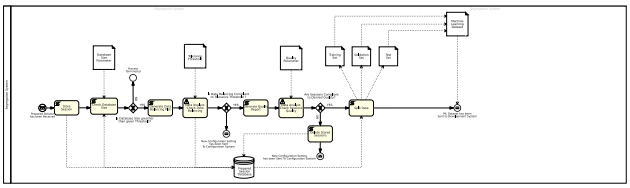


PREPARE MOBILITY SESSION (FABIO MALLOGGI)

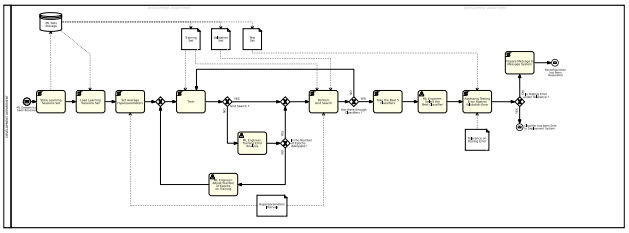


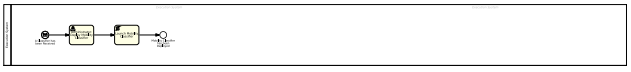


GENERATE MOBILITY SESSIONS SET (LORENZO TONELLI)



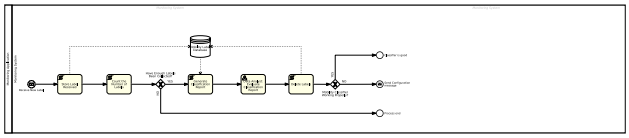
DEVELOPE MOBILITY CLASSIFIER (FRANCESCO MARABOTTO)





CLASSIFY MOBILITY SESSION (LUDOVICA COCCHELLA)

CHECK MOBILITY CLASSIFIER (LUDOVICA COCCHELLA)



3. USE CASES

3.1. Actors Involved

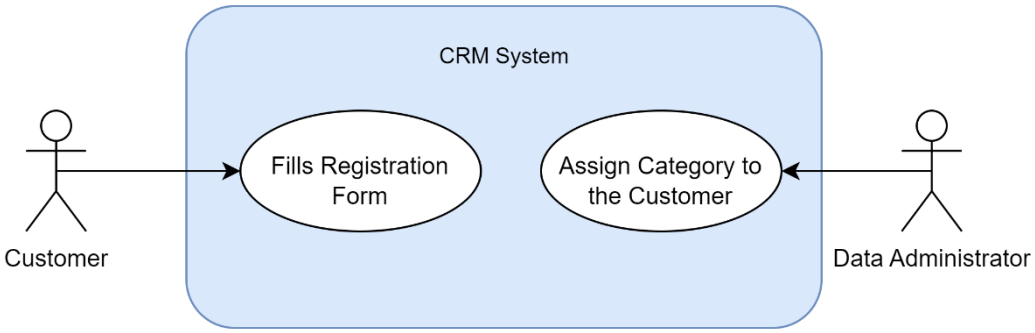
ACTOR	LINK	COST ($/hour)	NORMALIZED COST
ADMINISTRATOR	https://www.prospects.ac.uk/job-profiles/information-systems-manager	18 (Annual 36.000$)	1
DATA ANALYST	https://www.prospects.ac.uk/job-profiles/data-analyst	21 (Annual 42.000$)	1.17
DATA ADMINISTRATOR	https://www.prospects.ac.uk/job-profiles/database-administrator	27 (Annual 54.000$)	1.5
ML ENGINEER	https://www.prospects.ac.uk/job-profiles/machine-learning-engineer	33 (Annual 66.000$)	1.83
CUSTOMER		0	0

£ to $ conversion: 1 £ = 1.2 $

COST ($/hour) = Annual Salary ($) / (50 weeks * 40 hours per week)

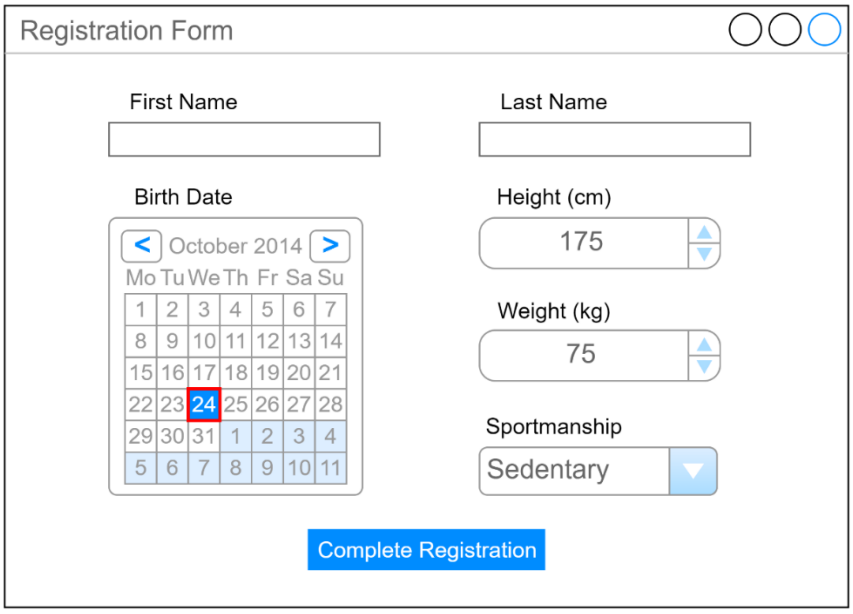
50 weeks because 2 weeks are considered as public holidays.

3.2. CONFIGURE MOBILITY CLASSIFIER – Use Cases



3.2.1. Customer Registration – Use Case (Fabio Malloggi)

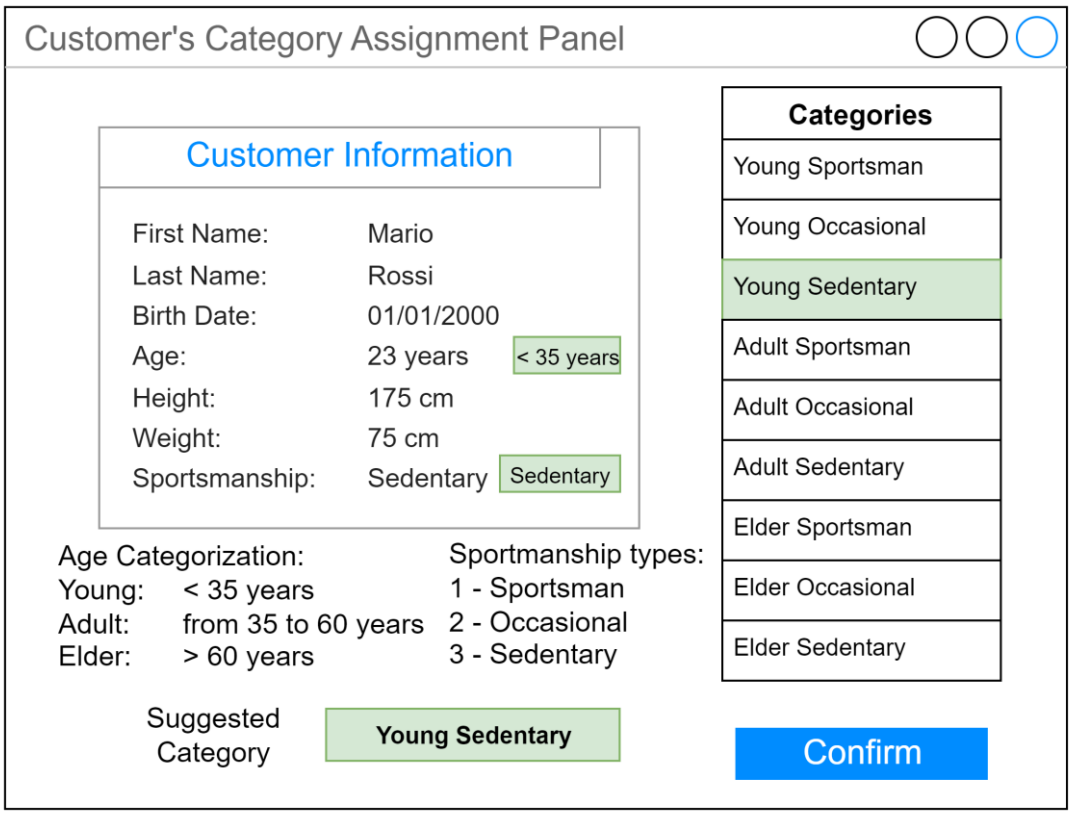
SUB-TASK	ACTOR	ACTION	COGNITIVE EFFORT	COST
1	**Customer**	The **Customer** clicks the "Register User" button	**Remember (1)**	1*0 = 0
2	**CRM System**	The **System** displays the Registration Form (Figure below)		
3	**Customer**	The **Customer** fills the form by entering his "Customer Parameters" Class elements (shown below) in the appropriate boxes.	**Remember (1)**	1*0 = 0
4	**Customer**	The Customer complete the registration by clicking the "Complete Registration" Button	**Remember (1)**	1*0 = 0
Total "Customer Registration"				**0**



Customer Parameters
First Name
Last Name
Birth Date
Height (cm)
Weight (kg)
Sportmanship

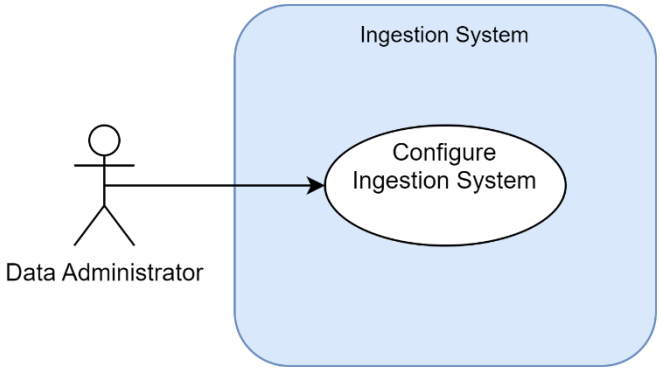
3.2.2. Customer's Category Assignment – Use Case (Fabio Malloggi)

SUB-TASK	ACTOR	ACTION	COGNITIVE EFFORT	COST
1	**CRM System**	The **System** notifies the Data Administrator that a new Customer has been accepted.		
1	**Administrator**	The **Administrator** clicks the "Customer's Category Assignment" button	**Remember (1)**	1*1 = 1
2	**CRM System**	The **System** displays the Customer's Category Assignment Panel (Figure below).		
3	**Administrator**	The **Administrator** selects the Category of the customer based on the suggested category given by the system.	**Understand (2)**	2*1 = 2
4	**Administrator**	The **Administrator** confirm the assignment by clicking the "Confirm" Button	**Remember (1)**	1*1 = 1
Total "Customer's Category Assignment"				**4**



3.3. PREPARE MOBILITY SESSION – Use Cases

3.3.1. Configure Ingestion System – Use Case (Fabio Malloggi)

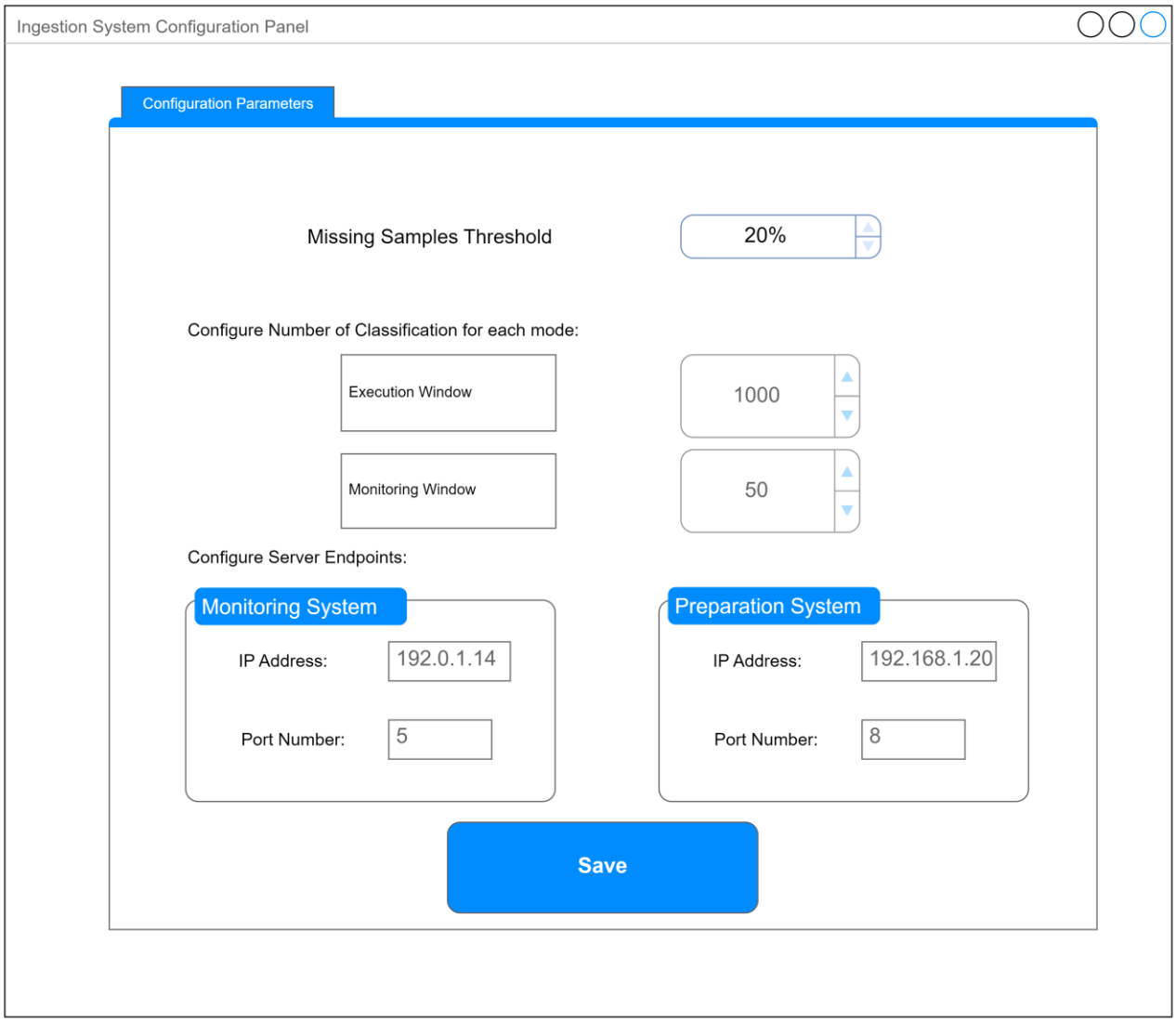


Configure Ingestion System

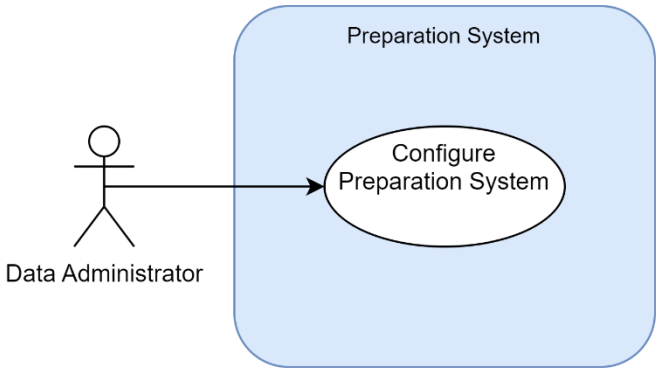
SUB-TASK	ACTOR	ACTION	COGNITIVE EFFORT	COST
1	**System**	The **System** shows the Configuration Message		
2	**Data Analyst**	The **Data Analyst** clicks the "Ingestion System Configuration" button.	**Remember (1)**	1*1.17 = 1.17
3	**System**	The **System** shows the Ingestion System Configuration Panel (Figure below).		
4	**Data Analyst**	The **Data Analyst** exploits his knowledge to set the Missing Samples Threshold New Value	**Analyze (4)**	4*1.17 = 4,68
5	**Data Analyst**	The **Data Analyst** sets the Execution Window value received through the message system.	**Remember (1)**	1*1.17 = 1.17
6	**Data Analyst**	The **Data Analyst** sets the Monitoring Window value received through the message system.	**Remember (1)**	1*1.17 = 1.17
7	**Data Analyst**	The **Data Analyst** sets the Endpoint IP address and Port of the Monitoring System received through the message system.	**Remember (1)**	1*1.17 = 1.17
8	**Data Analyst**	The **Data Analyst** sets the Endpoint IP address and Port of the Preparation System received through the message system.	**Remember (1)**	1*1.17 = 1.17
9	**Data Analyst**	The **Data Analyst** confirms the new setting by clicking on the "Save" Button	**Remember (1)**	1*1.17 = 1.17

Total "Configure Ingestion System" **11.7**

The previous use case is part of the larger process of configuring various systems, which is an approximate representation of a human glue mechanism. Indeed, those steps which require a low cognitive effort (steps 5 to 8), imply a previous human interaction in which the data analyst received the information needed.

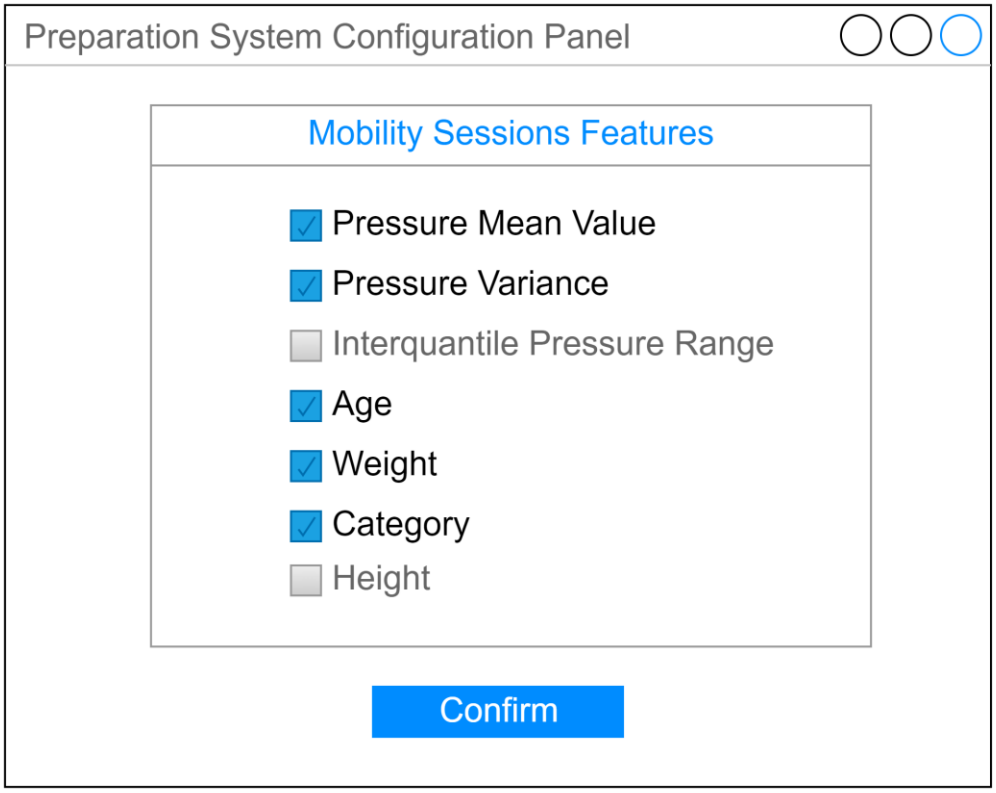


3.3.2. Configure Preparation System – Use Case (Fabio Malloggi)

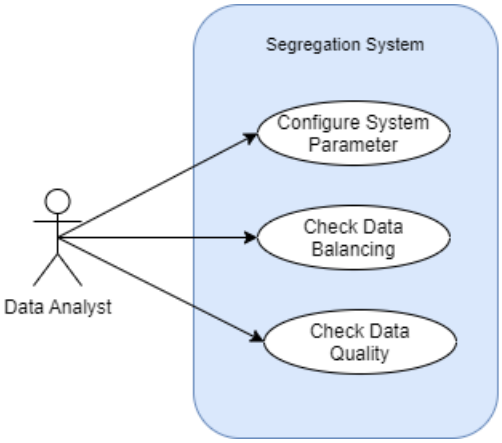


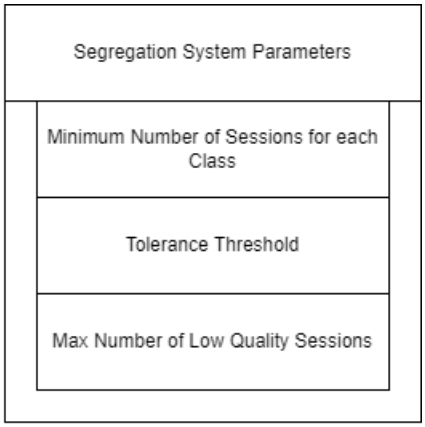
Configure Preparation System

SUB-TASK	ACTOR	ACTION	COGNITIVE EFFORT	COST
1	**Data Analyst**	The **Data Analyst** clicks the “Preparation System Configuration” button.	**Remember (1)**	1*1.17 = 1.17
2	**System**	The **System** shows the Preparation System Configuration Panel (Figure below).		
3	**Data Analyst**	The **Data Analyst** select the Features Set among the listed features.	**Understand (2)**	2*1.17 = 2.34
4	**Data Analyst**	The **Data Analyst** confirm the configuration by clicking the “Confirm” Button.	**Remember (1)**	1*1.17 = 1.17
Total “Configure Preparation System”				**4.68**



3.4. GENERATE MOBILITY SESSIONS SET – Use cases





3.4.1 CONFIGURE SEGREGATION SYSTEM – Use Case (Lorenzo Tonelli)

Segregation System Parameters

MinSessionsClass: For each class defined in our Mobility Behavior Monitoring Application, a minimum number of sessions has been defined in order to build a sufficiently large Machine Learning Dataset.
ToleranceTh: This is a percentage value (e.g 10%, 5%) used to define a interval in which we tolerate data unbalancing. The tolerance threshold is applied on the average of ALL classes (not for each couple of classes).
MaxLowQualitySessions: Before sending the dataset to Development System, Data Analyst check the data quality of sessions. The number of low quality sessions we can accept is defined by this parameter. To compare the session quality is used a description quality file given during configuration time.

SUB-TASK	ACTOR	ACTION	COGNITIVE EFFORT	COST
1	**System**	The **System** shows the Configuration Message		
2	**Data Analyst**	The **Data Analyst** clicks the "Segregation System Configuration" button.	**Remember (1)**	1*1.17 = 1.17
3	**System**	The **System** shows the Segregation System Configuration Panel (Figure below).		
4	**Data Analyst**	The **Data Analyst** exploits his knowledge to set the MaxLowQualitySessions.	**Analyze (4)**	4*1.17 = 4,68
5	**Data Analyst**	The **Data Analyst** sets the ToleranceTh parameter.	**Remember (1)**	1*1.17 = 1.17
6	**Data Analyst**	The **Data Analyst** sets MinSessionsClass parameter.	**Remember (1)**	1*1.17 = 1.17
7	**Data Analyst**	The **Data Analyst** sets the Endpoint IP address and Port of the Message System.	**Remember (1)**	1*1.17 = 1.17
Total "Configure Preparation System"				**9.36**

3.4.2 CHECK DATA BALANCING – Use Case (Lorenzo Tonelli)

Check Data Balancing

SUB-TASK	ACTOR	ACTION	COGNITIVE EFFORT	COST
1	**Data Analyst**	Data Analyst selects the button to show Data Balancing Plot	**Remember (1)**	1*1.17 = 1.17
2	**System**	System shows Data Balancing Plot		
3	**Data Analyst**	Data Analyst observes the horizontal line on the plot	**Understand (2)**	2*1.17 = 2.34
4 – IF (50%)		If the horizontal line is red on the plot		
4.1	**Data Analyst**	Data Analyst calculate how many new sessions are missing	**Apply (3)**	0.5*3*1.17 = 1.76
4.1a	**Data Analyst**	Data Analyst for each class under threshold compute difference between that and class size		
4.1b	**Data Analyst**	Data Analyst adds up the results		
4.2	**Data Analyst**	Data Analyst insert in the text area the number of new sessions to collect	**Remember (1)**	0.5*1.17*1 = 0.59
4.3	**Data Analyst**	Data Analyst clicks on “Collect New Sessions” button	**Remember (1)**	0.5*1.17*1 = 0.59
4.4	**System**	System sends a message to Configuration System		
4.5	**System**	System returns to collect new sessions		
5 – ELSE (50%)		(The horizontal line is green on the plot)		
5.1	**Data Analyst**	Data Analyst observes the excess line on the plot	**Understand (2)**	0.5*2*1.17 = 1.17
5.2 – IF (20%)		If each size class is greater than the excess threshold		
5.2.1	**Data Analyst**	Data Analyst clicks on “Start Checking Data Quality” button	**Remember (1)**	0.5*0.2*1.17*1 = 0.12
5.2.2	**System**	System sends a message to Development System		
5.3 – ELSE (80%)		(There is at least one class with size less than excess threshold)		
5.3.1	**Data Analyst**	Data Analyst calculate how many new sessions are missing to reach excess threshold	**Apply (3)**	0.5*0.8*3*1.17 = 1.4
5.3.1a	**Data Analyst**	Data Analyst for each class under excess threshold compute		

		difference between that and class size		
5.3.1b	**Data Analyst**	Data Analyst insert in the text area the number of new sessions to collect		
5.3.2	**Data Analyst**	Data Analyst clicks on “Collect New Sessions” button	**Remember (1)**	0.5*0.8*1*1.17=0.47
5.3.3	**System**	System sends a message to Configuration System		
5.3.4	**System**	System returns to collect new sessions		

Total “Check Data Balancing” **9.61**

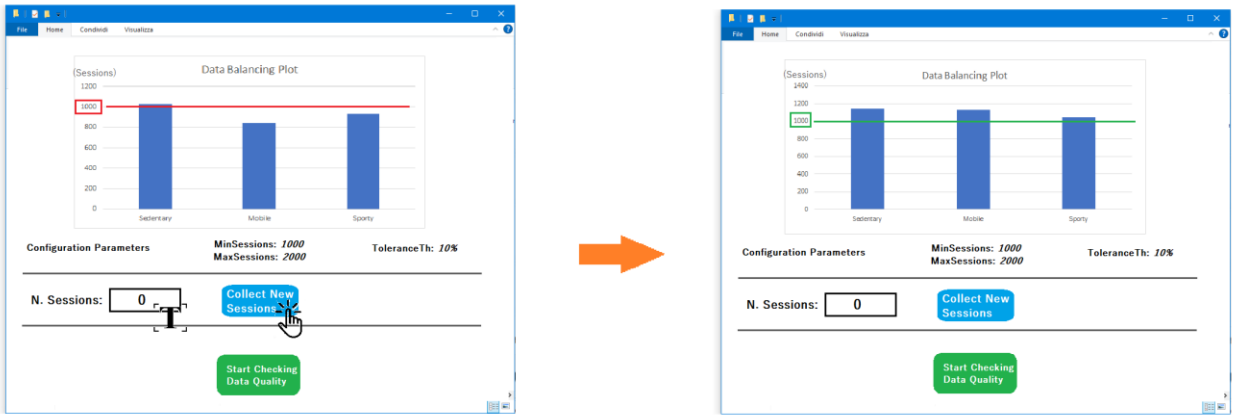


Figure 1 – The plot on the left shows that there are classes with not enough sessions. The user specifies the number of new sessions desired and then click on button “Collect New Sessions”. The plot on the right shows that new sessions has been stored in the segregation system and the minimum number of sessions for each class has been reached.

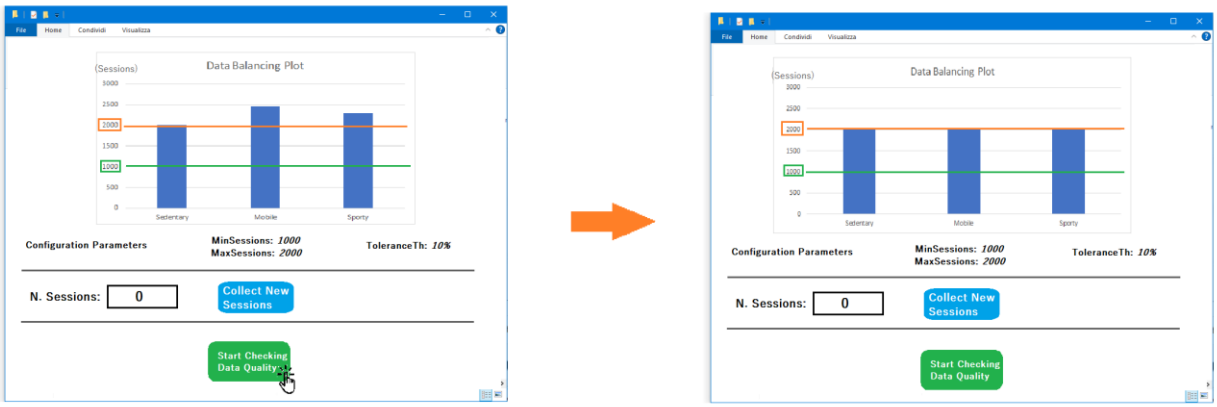
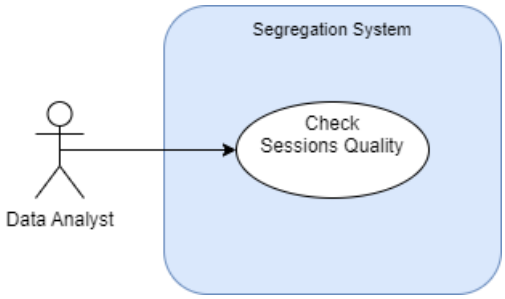


Figure 2 – The data distribution is highly unbalanced due to ‘Sporty’ class. The user can cope that removing some sessions in random way on the ‘Sporty’ class. The user has to define the number of sessions to remove; specify the name of class on which remove sessions; finally click on the “Remove Sessions button”. On the right we can see that data are balanced.

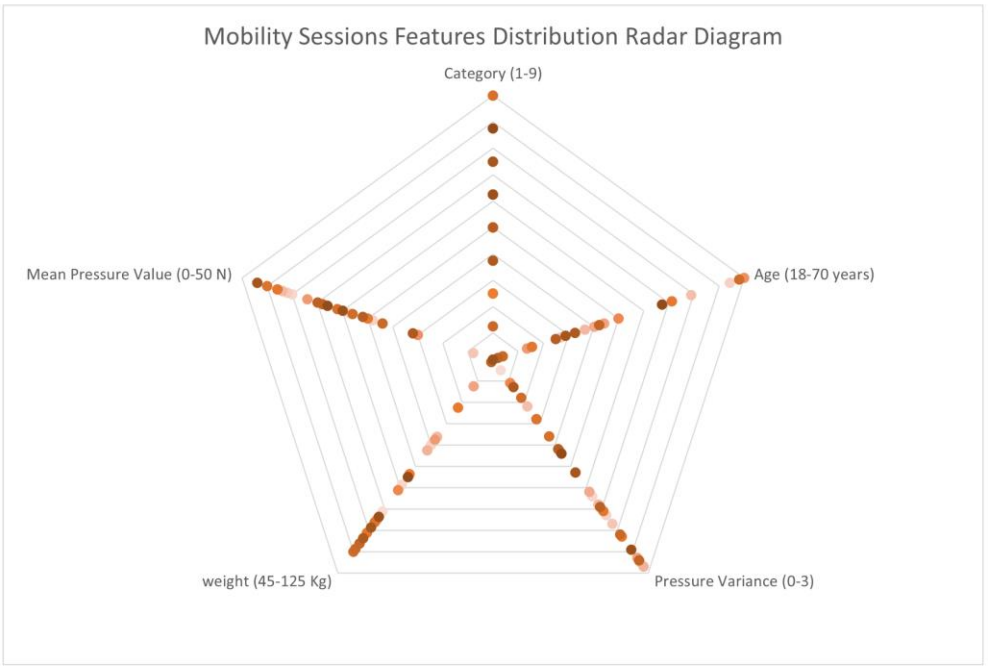
3.4.3 CHECK SESSIONS QUALITY – Use Case (Fabio Malloggi)



Check Sessions Quality

SUB-TASK	ACTOR	ACTION	COGNITIVE EFFORT	COST
1	**System**	The **System** notify the Data Analyst that a new balanced ML dataset is ready.		
2	**Data Analyst**	The **Data Analyst** clicks the Quality Report Button	**Remember (1)**	1*1.17 = 1.17
3	**System**	The **System** shows Sessions Quality Report Panel		
4	**Data Analyst**	The **Data Analyst** analyzes the Mobility Sessions Features Radar Diagram	**Analyze (4)**	1.17*4 = 4.68
5 – IF (70%)		IF all features are uniformly distributed (Figure 1 below)		
5.1		The **Data Analyst** approves the ML Dataset by clicking on the “Approve” Button	**Remember (1)**	1*1.17*0.7 = 0.82
6 – ELSE (30%)		If at least one Mobility Sessions Feature lacks a relevant interval of values (Figure 2 below)		
6.1	**Data Analyst**	The **Data Analyst** clicks on “Reject Dataset” Button	**Remember (1)**	1*1.17*0.3 = 0.35
6.2	**System**	The **System** send a Configuration Message to the Message System		
Total				**7.02**

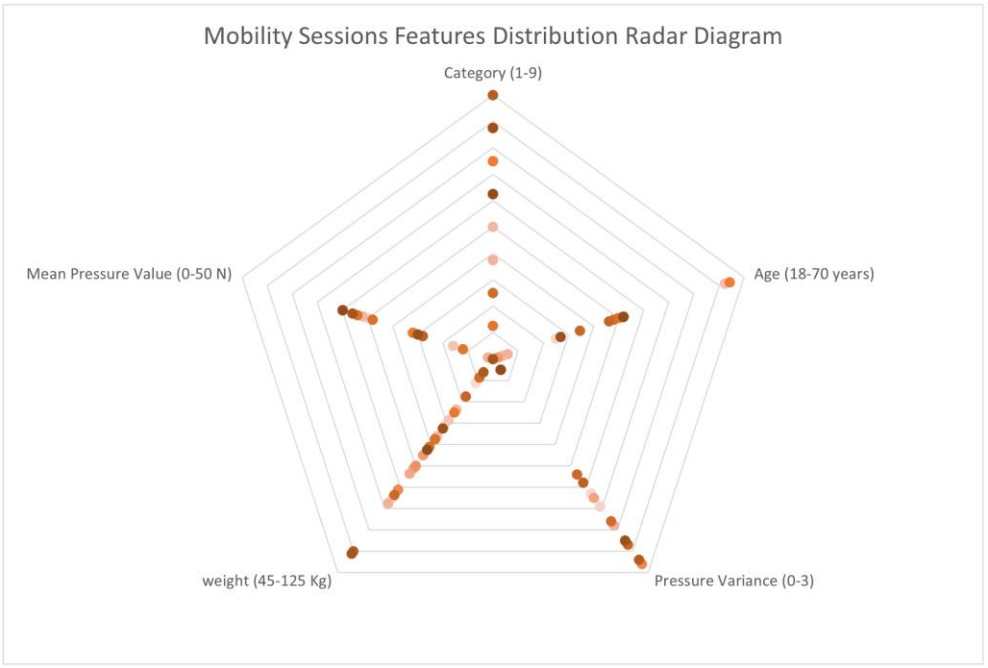
Sessions Quality Report Panel



Reject Dataset

Approve Dataset

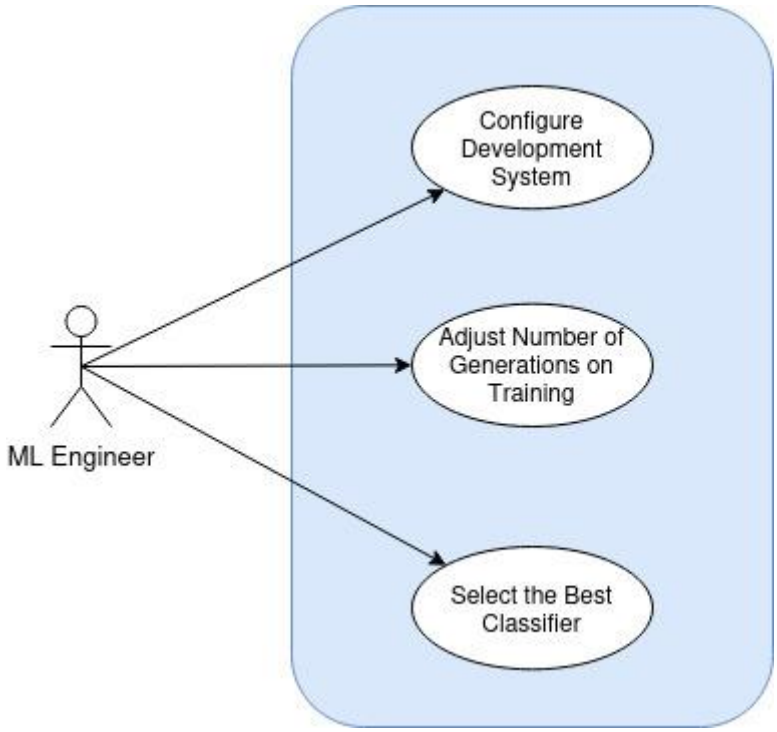
Sessions Quality Report Panel



Reject Dataset

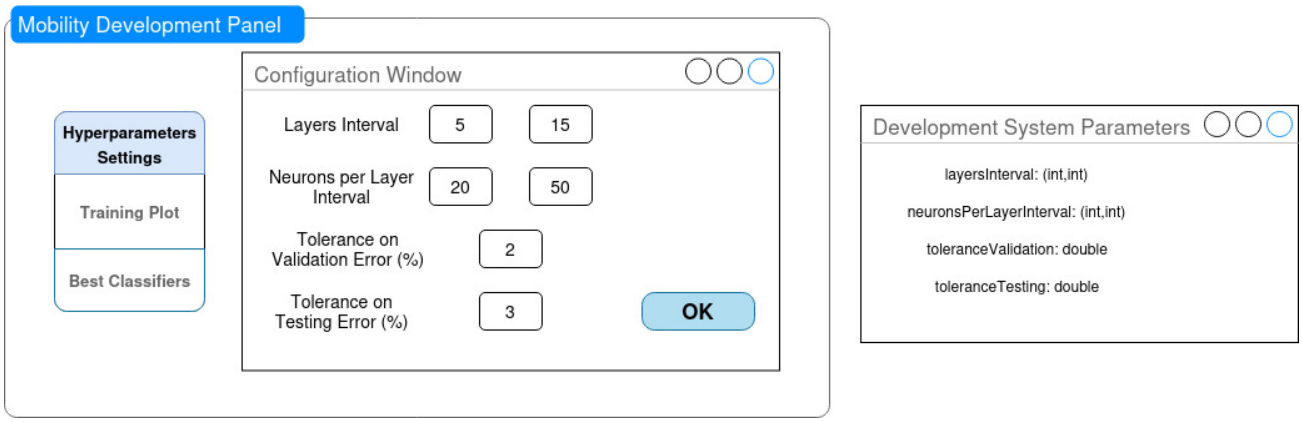
Approve Dataset

3.5. DEVELOPMENT MOBILITY CLASSIFIER – Use cases



3.5.1. CONFIGURE DEVELOPMENT SYSTEM - Use Case (Francesco Marabotto)

SUB-TASK	ACTOR	ACTION	COGNITIVE EFFORT	COST
1	**System**	The system notifies that a configuration request has arrived		
2	**ML Engineer**	Clicks on “Hyperparameters Settings” button on the “Mobility Development Panel”	**Remember (1)**	1.83*1 = 1.83
3	**ML Engineer**	Configures the “Layers Interval” by experience	**Analyze (4)**	1.83*4 = 7.32
4	**ML Engineer**	Configures the “Neurons per Layer Interval” by experience	**Analyze (4)**	1.83*4=7.32
5	**ML Engineer**	Configures the “Tolerance on Validation Error (%)” by experience	**Analyze (4)**	1.83*4=7.32
6	**ML Engineer**	Configures the “Tolerance on Testing Error (%)” by experience	**Analyze (4)**	1.83*4=7.32
7	**ML Engineer**	Clicks on “OK” button	**Remember (1)**	1.83*1=1.83
Total Configure Development System				**32.94**

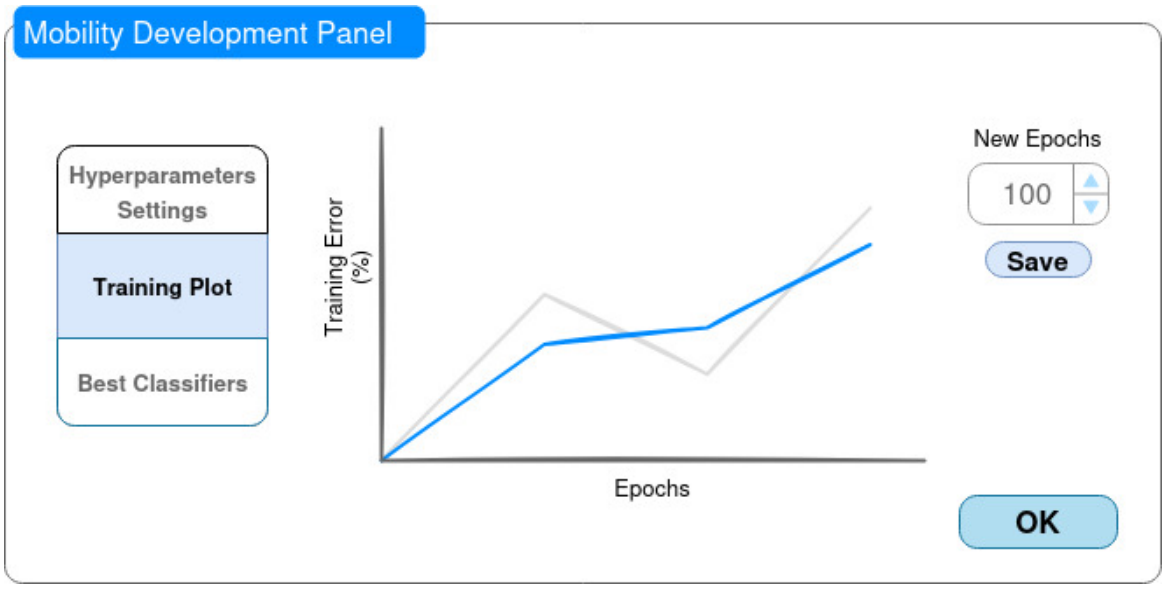


Hyperparameters Configuration

3.5.2. ADJUST NUMBER OF GENERATIONS ON TRAINING – Use Case (Francesco Marabotto)

SUB-TASK	ACTOR	ACTION	COGNITIVE EFFORT	COST
1	**ML Engineer**	Clicks on "Training Plot" button	**Remember (1)**	1.83*1 = 1.83
2	**System**	Opens "Training Plot" panel		
3	**ML Engineer**	Analyze the training error plot	**Apply (3)**	1.83*3 = 5.49
4 – IF (25%)	**System**	If the number of epochs is adequate		
4.1	**ML Engineer**	Clicks on "OK" button	**Remember (1)**	0.25*1.83*1 = 0.46
5 – ELSE (75%)	**System**	ELSE		
5.2	**ML Engineer**	Adjust the new number of epochs by experience	**Analyze (4)**	0.75*1.83*4 = 5.49
5.3	**ML Engineer**	Set the new number of epochs in the field	**Remember (1)**	0.75*1.83*1= 1.37
5.4	**ML Engineer**	Clicks on "Save" button	**Remember (1)**	0.75*1.83*1 = 1.37
5.4	**ML Engineer**	Clicks on "OK" button	**Remember (1)**	0.75*1.83*1 = 1.37
6	**System**	Shows a message informing the ML Engineer that the new number of epochs has been successfully saved		

Total *"Adjust Number of Generations on Training"* **17.38**

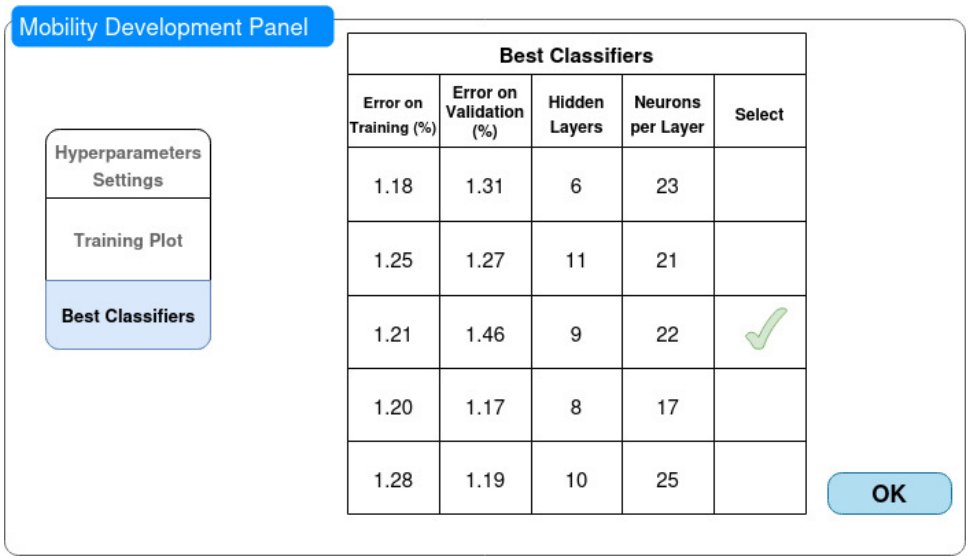


Training Error Plot

3.5.3. SELECT THE BEST CLASSIFIER – Use Case (Francesco Marabotto)

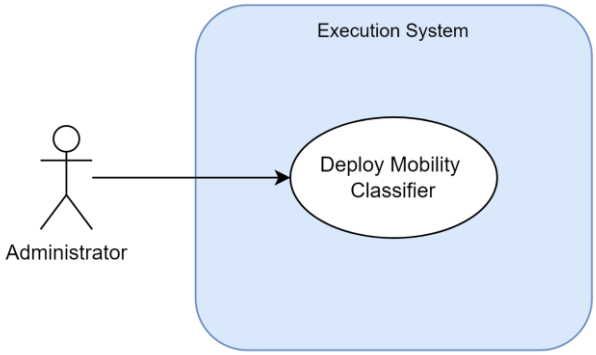
SUB-TASK	ACTOR	ACTION	COGNITIVE EFFORT	COST
1	ML Engineer	Clicks on “Best Classifiers” button	Remember (1)	1.83*1 = 1.83
2	System	Opens “Best Classifiers” panel		
3	ML Engineer	Analyze the classifiers table by experience	Analyze (4)	1.83*4 = 7.32
4	ML Engineer	Select, by clicking on the green tick in the classifier table, the best classifier	Remember (1)	1.83*1 = 1.83
5	ML ENGINEER	Clicks on “OK” button	Remember (1)	1.83*1 = 1.83
6	System	Shows a message informing the ML Engineer that the classifier has been correctly chosen		

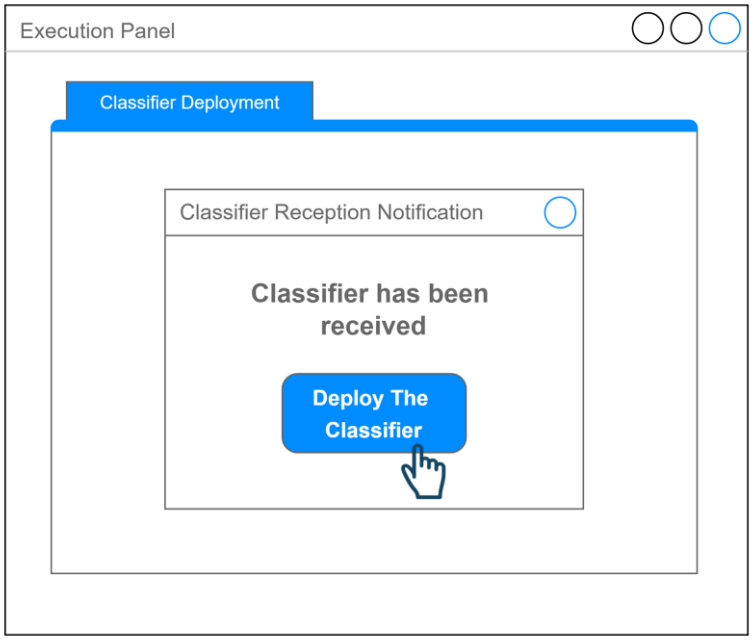
Total “Select the Best Classifier” **12.81**



Best Classifiers

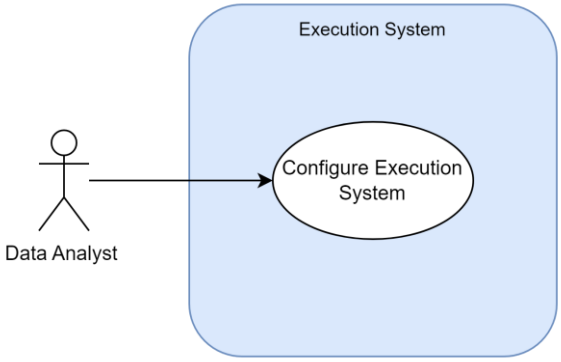
3.5.4. DEPLOY MOBILITY CLASSIFIER - Use Case (Ludovica Cocchella)





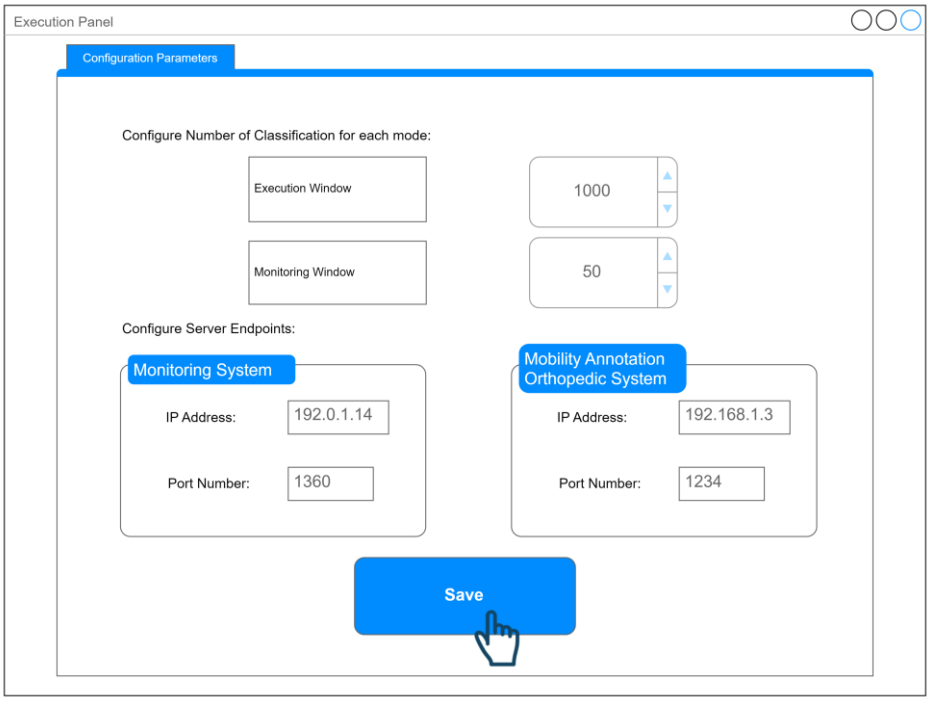
SUB-TASK	ACTOR	ACTION	COGNITIVE EFFORT	COST
1	System	Notifies the Data Analyst that a new Classifier has been received.		
2	Administrator (1)	Click “Classifier Deployment” button	Remember (1)	1*1 = 1
3	System	Show Classifier Reception Notification		
2	Administrator (1)	Clicks “Deploy the classifier” button	Remember (1)	1*1 = 1
3	System	Deploys the classifier		
Total “Deploy the Classifier”				**2**

3.6. CLASSIFY MOBILITY SESSION – Use Cases



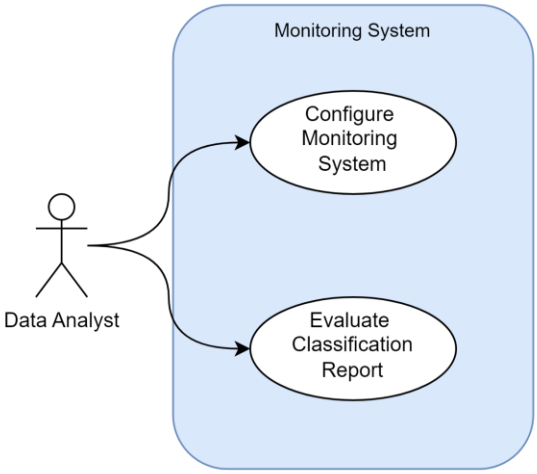
3.6.1. CONFIGURE EXECUTION SYSTEM - Use Case (Ludovica Cocchella)

Set number of classifications to do before switching from execution to monitoring mode and vice versa



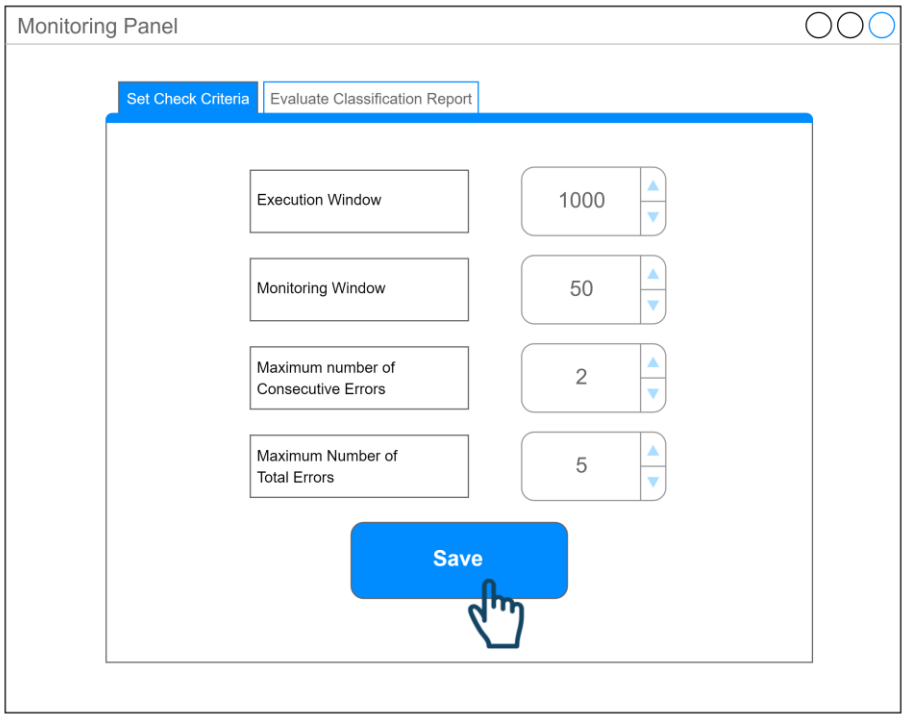
SUB-TASK	ACTOR	ACTION	COGNITIVE EFFORT	COST
1	**Data Analyst (1.17)**	Clicks on “Configuration Parameters” button	**Remember (1)**	1.17*1 = 1.17
2	**System**	Shows the Configuration Parameters” interface		
3	**Data Analyst (1.17)**	Set the Execution Window value obtained from the message system	**Remember (1)**	1.17*1 = 1.17
4	**Data Analyst (1.17)**	Set the Monitoring Window value obtained from the message system	**Remember (1)**	1.17*1 = 1.17
5	**Data Analyst (1.17)**	Set the Endpoint IP address of the Monitoring System obtained from the message system	**Remember (1)**	1.17*1 = 1.17
6	**Data Analyst (1.17)**	Set the Port Number of the Monitoring System obtained from the message system	**Remember (1)**	1.17*1 = 1.17
7	**Data Analyst (1.17)**	Set the Endpoint IP address of the Mobility Annotation Orthopedic System obtained from the message system	**Remember (1)**	1.17*1 = 1.17
8	**Data Analyst (1.17)**	Set the Port Number of the Mobility Annotation Orthopedic System obtained from the message system	**Remember (1)**	1.17*1 = 1.17
9	**Data Analyst (1.17)**	Clicks on the “Save” button	**Remember (1)**	1.17*1 = 1.17
10	**System**	Saves the check criteria		
Total “Configure Execution System”				**9,36**

3.7. CHECK MOBILITY CLASSIFIER – Use Cases



3.7.1. CONFIGURE MONITORING SYSTEM - Use Case (Ludovica Cocchella)

Set check criteria for evaluating the classifier

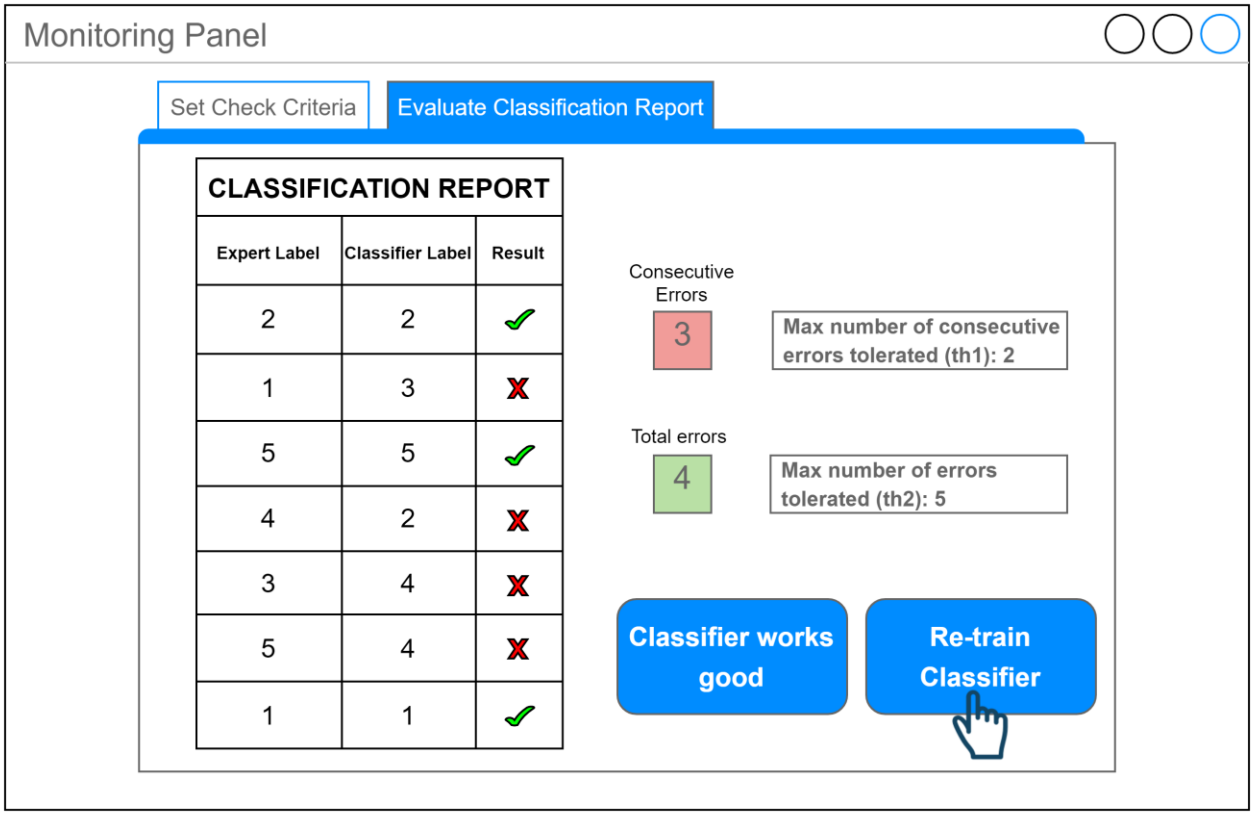


SUB-TASK	ACTOR	ACTION	COGNITIVE EFFORT	COST
1	**Data Analyst (1.17)**	Clicks on the "Set Check Criteria Interface" button on Monitoring Panel	**Remember (1)**	1.17*1 = 1.17
2	**System**	Show Set Check Criteria Interface		
3	**Data Analyst (1.17)**	Set the Execution Window value by experience	**Analyze (4)**	1.17*4 = 4.68
4	**Data Analyst (1.17)**	Set the Monitoring Window value by experience	**Analyze (4)**	1.17*4 = 4.68
5	**Data Analyst (1.17)**	Set the check criteria Maximum number of Consecutive Errors by experience	**Analyze (4)**	1.17*4 = 4.68
6	**Data Analyst (1.17)**	Set the check criteria for Maximum number of total errors by experience	**Analyze (4)**	1.17*4 = 4.68
7	**Data Analyst (1.17)**	Clicks on the "Save" button	**Remember (1)**	1.17*1 = 1.17
8	**System**	Saves the check criteria		

Total "Set Check Criteria for Evaluating the Classifier" **21.06**

3.7.2. EVALUATE CLASSIFICATION REPORT - Use Case (Ludovica Cocchella)

Check Mobility Classifier Performance



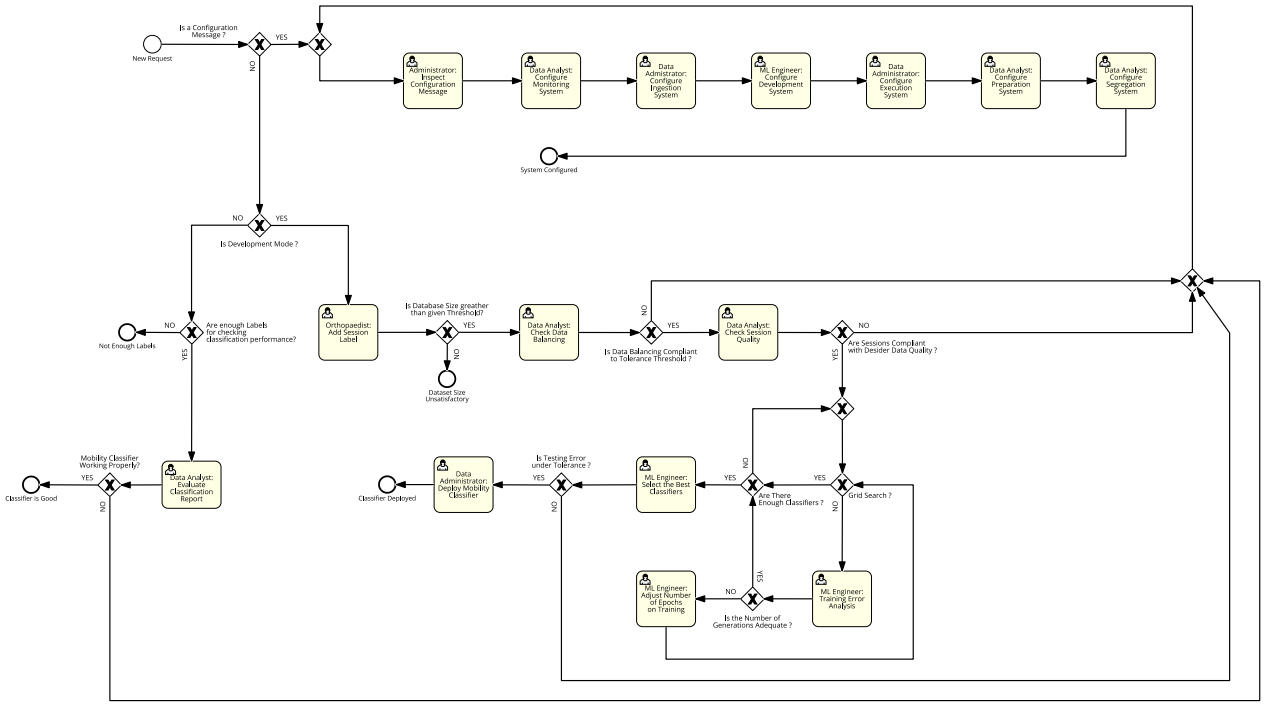
1	**Data Analyst (1.17)**	Clicks the button "Evaluate Classification Report" on the Monitoring Panel	**Remember (1)**	1*1.17 = 1.17
2	**System**	Show the classification report interface		
3	**System**	Checks for each label pair if a mismatch has occurred		
4 – IF (10%)		If a mismatch has occurred		
4.1	**System**	Increase the consecutive errors count		
4.2	**System**	Increase the total error count		
5 – IF (5%)		If both consecutive errors and total errors are greater than their respective thresholds		
5.1	**System**	Sets the box color corresponding to consecutive errors and total errors to RED		
6– ELSE IF (5%)		If only one of the two counters are greater than their respective thresholds		
6.1	**System**	Sets the box color corresponding to the counter which overcome the threshold to RED and the other box color to GREEN		
7 – ELSE (90%)		ELSE		

7.1	**System**	Sets both boxes color to GREEN		
8	**Data Analyst (1.17)**	Checks the color of the two boxes	**Understand (2)**	2*1.17 = 2.34
9 – IF (10%)		If at least one of the two boxes color is RED		
9.1	**Data Analyst (1.17)**	Clicks on “re-train Classifier” button	**Remember (1)**	1*1.17*0.1 = 0.117
10 – ELSE (90%)		If both colors of the boxes are GREEN		
10.1	**Data Analyst (1.17)**	Clicks on “Classifier works good” button	**Remember (1)**	1*1.17*0.9 = 1.053

Total “Evaluation Classification Performance” **4.68**

4. SIMULATION MODELS

4.1. AS-IS Model



The assumptions we make in order to simulate the process are the following:

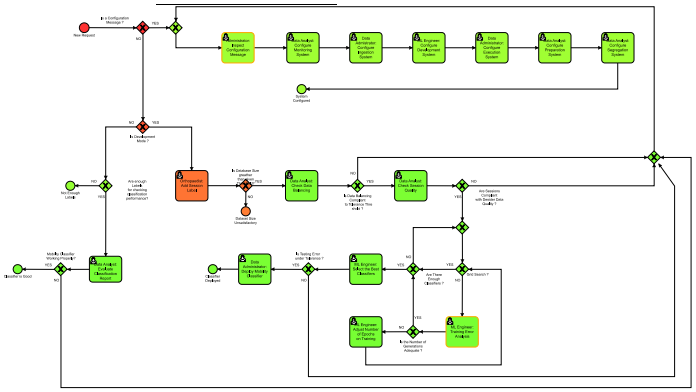
- Suppose 10,000 tokens are enough to emulate the behavior of a company in its operational phase. out of these 10000 tokens, about 10% are reconfiguration messages, the rest are new sessions. So, we have 9000 sessions and 1000 reconfiguration messages
- of the 9000 sessions, we assume that 90% are related to classifier development, while the remaining 10% are monitoring sessions.
- Of these 8100 tokens, they partially represent the sessions that are used to build the dataset necessary for the subsequent training of the classifier. To build the dataset, some situations have been considered:
    1) it is large enough (10% yes and 90% no)
    2) the dataset obtained is balanced according to the data imbalance tolerance threshold (33% yes and 67% no)
    3) the quality of the dataset complies with the desired data quality (60% yes and 40% no).
- the remaining tokens, which we want at least 500, are used to be consumed by the human tasks related to classifier training.
- We expect that the task related to the use of the classifier is reached by a few tokens (at most 20 times). In order to reflect the behavior of a company in its operational phase.

GATEWAY	TYPE	ACTION	PERCENTAGE
Are Sessions Compliant with Desider Data Quality?	XOR	YES	60%
		NO	40%
Are There Enough Classifiers?	XOR	YES	85%
		NO	15%
	XOR	YES	3%

Are enough Labels for checking classification performance?		NO	97%
Grid Search?	XOR	YES	80%
		NO	20%
Is Data Balancing Compliant to Tolerance Threshold?	XOR	YES	33%
		NO	67%
Is Database Size greater than given Threshold?	XOR	YES	3%
		NO	97%
Is Development Mode?	XOR	YES	40%
		NO	60%
Is Testing Error under Tolerance?	XOR	YES	90%
		NO	10%
Is a Configuration Message?	XOR	YES	10%
		NO	90%
Is a Registration Request?	XOR	YES	90%
		NO	10%
Is the Number of Generations Adequate?	XOR	YES	40%
		NO	60%
Mobility Classifier Working Properly?	XOR	YES	85%
		NO	15%

Heatmap

Heatmap based on Counts

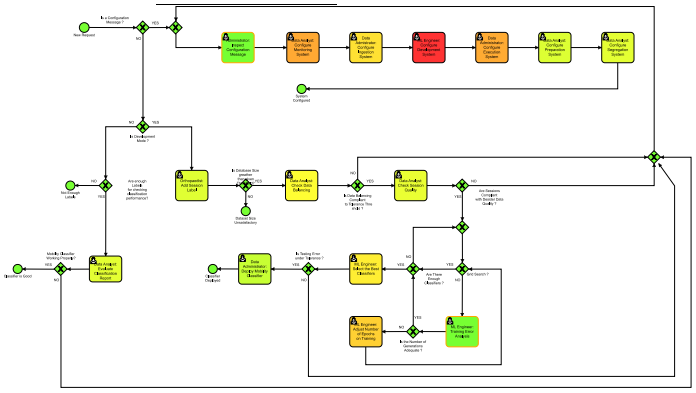


Legend

Color	Value
	0
	1111
	2222
	3333
	4444
	5556
	6667
	7778
	8889
	10000

Heatmap

Heatmap based on Durations



Legend

Color	Value
	0 s
	4 s
	7 s
	11 s
	15 s
	18 s
	22 s
	26 s
	29 s
	33 s

Simulation Results

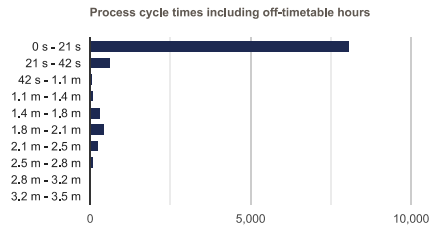
General information

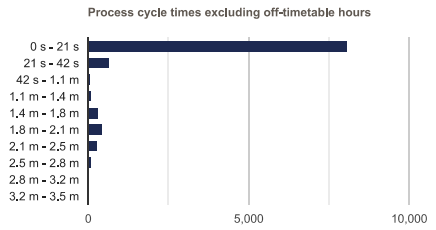
Completed process instances 10000

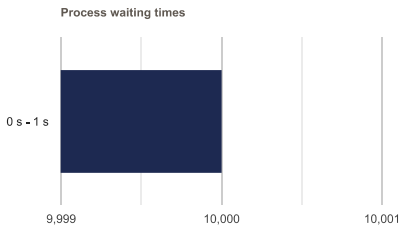
Total cost 0 USD

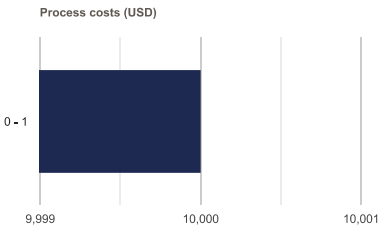
Total simulation time 41,4 weeks

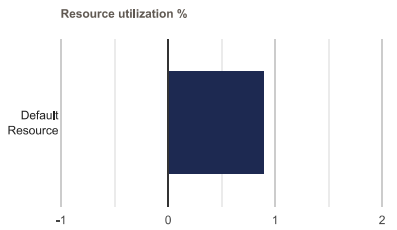
Charts











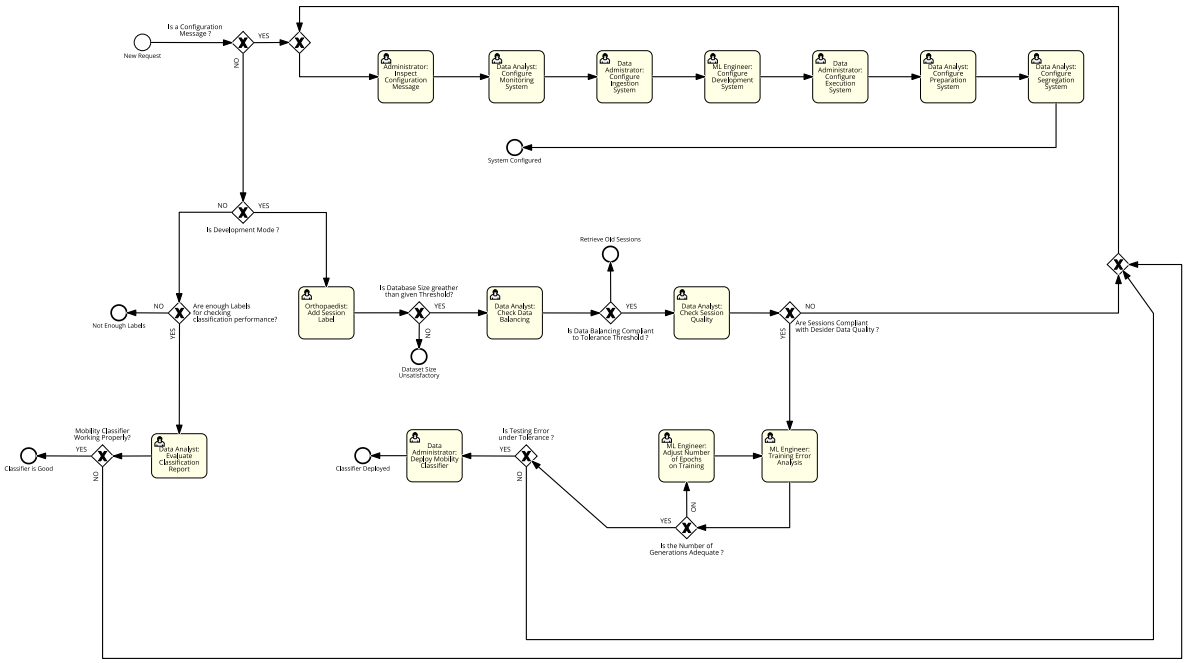
Scenario Statistics

	Minimum	Maximum	Average
Process instance cycle times including off-timetable hours	0 seconds	3.4 minutes	22.6 seconds
Process instance cycle times excluding off-timetable hours	0 seconds	3.4 minutes	22.6 seconds
Process instance costs	0 USD	0 USD	0 USD

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
Data Analyst: Evaluate Classification Report	23	0 s	0 s	0 s	0.4 s	10.6 s	27.6 s	0 s	0 s	0 s	0	0	0	0	0	0
Data Administrator:
Deploy Mobility Classifier	84	0 s	0 s	0 s	0 s	7.8 s	22 s	0 s	0 s	0 s	0	0	0	0	0	0
Data Administrator: Configure Execution System	1206	0 s	0 s	0 s	0.8 s	21.4 s	51.7 s	0 s	0 s	0 s	0	0	0	0	0	0
Data Admistrator: Configure Ingestion System	1206	0 s	0 s	0 s	0.1 s	16.1 s	45.1 s	0 s	0 s	0 s	0	0	0	0	0	0
Data Analyst: Check Data Balancing	245	0 s	0 s	0 s	0 s	13.3 s	47.6 s	0 s	0 s	0 s	0	0	0	0	0	0
Data Analyst: Check Session Quality	170	0 s	0 s	0 s	0.1 s	11.2 s	40 s	0 s	0 s	0 s	0	0	0	0	0	0
Data Analyst: Configure Monitoring System	1206	0 s	0 s	0 s	0.2 s	21.5 s	49.6 s	0 s	0 s	0 s	0	0	0	0	0	0
Data Analyst: Configure Preparation System	1206	0 s	0 s	0 s	0 s	9.9 s	39.3 s	0 s	0 s	0 s	0	0	0	0	0	0
Data Analyst: Configure Segregation System	1206	0 s	0 s	0 s	0 s	10.6 s	39 s	0 s	0 s	0 s	0	0	0	0	0	0
ML Engineer:
Adjust Number
of Epochs
on Training	21	0 s	0 s	0 s	0.1 s	17.8 s	30.3 s	0 s	0 s	0 s	0	0	0	0	0	0
ML Engineer:
Select the Best Classifiers	93	0 s	0 s	0 s	0.2 s	14.8 s	45.2 s	0 s	0 s	0 s	0	0	0	0	0	0
ML Engineer: Configure Development System	1206	0 s	0 s	0 s	1.2 s	32.9 s	1.2 m	0 s	0 s	0 s	0	0	0	0	0	0
Orthopaedist: Add Session Label	8086	0 s	0 s	0 s	0 s	10.2 s	41.1 s	0 s	0 s	0 s	0	0	0	0	0	0

4.2. TO-BE Model



The TO-BE model was created starting from the AS-IS model adding three improvements to the processes:

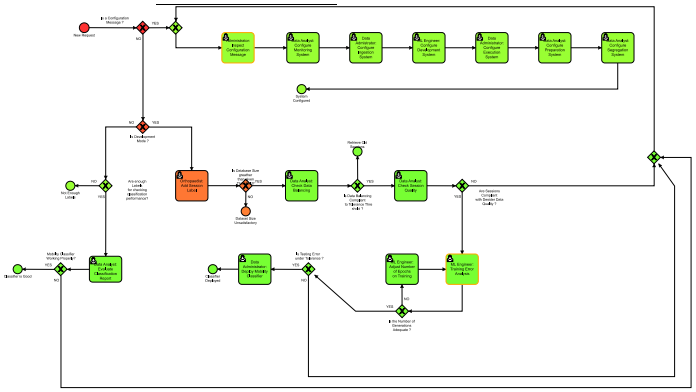
1) at handoff level: balance classes by taking minor data from old sessions database.

2) at service level: Since the Best hyperparameters are retrieved on the history of already developed classifiers, once the number of training epochs is defined, a single classifier has been trained and no more grid search and selection of the best classifier must be performed.

3) at task level: the total cognitive effort of the task “adjust number of epochs on training” has been reduce from 17.38 to 16.01.

Heatmap

Heatmap based on Counts

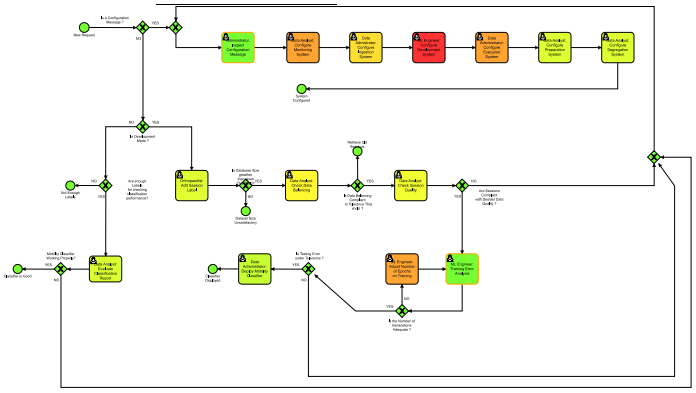


Legend

Color	Value
	0
	1111
	2222
	3333
	4444
	5556
	6667
	7778
	8889
	10000

Heatmap

Heatmap based on Durations



Legend

Color	Value
	0 s
	4 s
	7 s
	11 s
	15 s
	18 s
	22 s
	25 s
	29 s
	33 s

Simulation Results

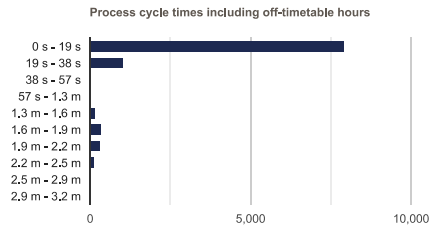
General information

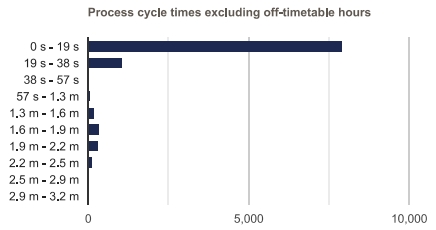
Completed process instances 10000

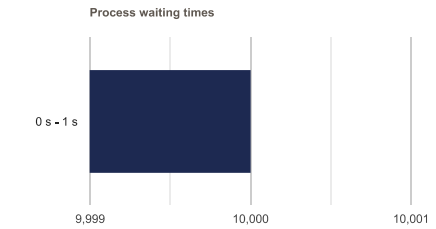
Total cost 0 USD

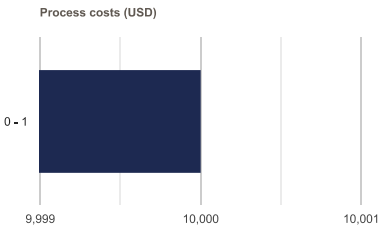
Total simulation time 41,4 weeks

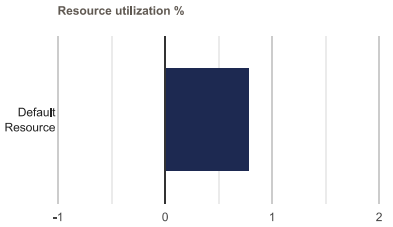
Charts











Scenario Statistics

	Minimum	Maximum	Average
Process instance cycle times including off-timetable hours	0 seconds	3 minutes	19.9 seconds
Process instance cycle times excluding off-timetable hours	0 seconds	3 minutes	19.9 seconds
Process instance costs	0 USD	0 USD	0 USD

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
Data Analyst: Evaluate Classification Report	29	0 s	0 s	0 s	0.6 s	9.6 s	27.5 s	0 s	0 s	0 s	0	0	0	0	0	0
Data Administrator:
Deploy Mobility Classifier	4	0 s	0 s	0 s	3.1 s	7.8 s	12.5 s	0 s	0 s	0 s	0	0	0	0	0	0
Data Administrator: Configure Execution System	1004	0 s	0 s	0 s	0.2 s	21.9 s	1.1 m	0 s	0 s	0 s	0	0	0	0	0	0
Data Admistrator: Configure Ingestion System	1004	0 s	0 s	0 s	0 s	16.3 s	50.3 s	0 s	0 s	0 s	0	0	0	0	0	0
Data Analyst: Check Data Balancing	234	0 s	0 s	0 s	0.2 s	13.8 s	35.1 s	0 s	0 s	0 s	0	0	0	0	0	0
Data Analyst: Check Session Quality	4	0 s	0 s	0 s	2.2 s	11.8 s	32.3 s	0 s	0 s	0 s	0	0	0	0	0	0
Data Analyst: Configure Monitoring System	1004	0 s	0 s	0 s	0 s	21.9 s	54 s	0 s	0 s	0 s	0	0	0	0	0	0
Data Analyst: Configure Preparation System	1004	0 s	0 s	0 s	0.1 s	10 s	36.1 s	0 s	0 s	0 s	0	0	0	0	0	0
Data Analyst: Configure Segregation System	1004	0 s	0 s	0 s	0.1 s	10.2 s	44.4 s	0 s	0 s	0 s	0	0	0	0	0	0
ML Engineer:
Adjust Number
of Epochs
on Training	4	0 s	0 s	0 s	14 s	21.8 s	30 s	0 s	0 s	0 s	0	0	0	0	0	0
ML Engineer: Configure Development System	1004	0 s	0 s	0 s	2.9 s	32.8 s	1 m	0 s	0 s	0 s	0	0	0	0	0	0
Orthopaedist: Add Session Label	8019	0 s	0 s	0 s	0 s	10.2 s	47.7 s	0 s	0 s	0 s	0	0	0	0	0	0

4.3. BIMP Conclusion

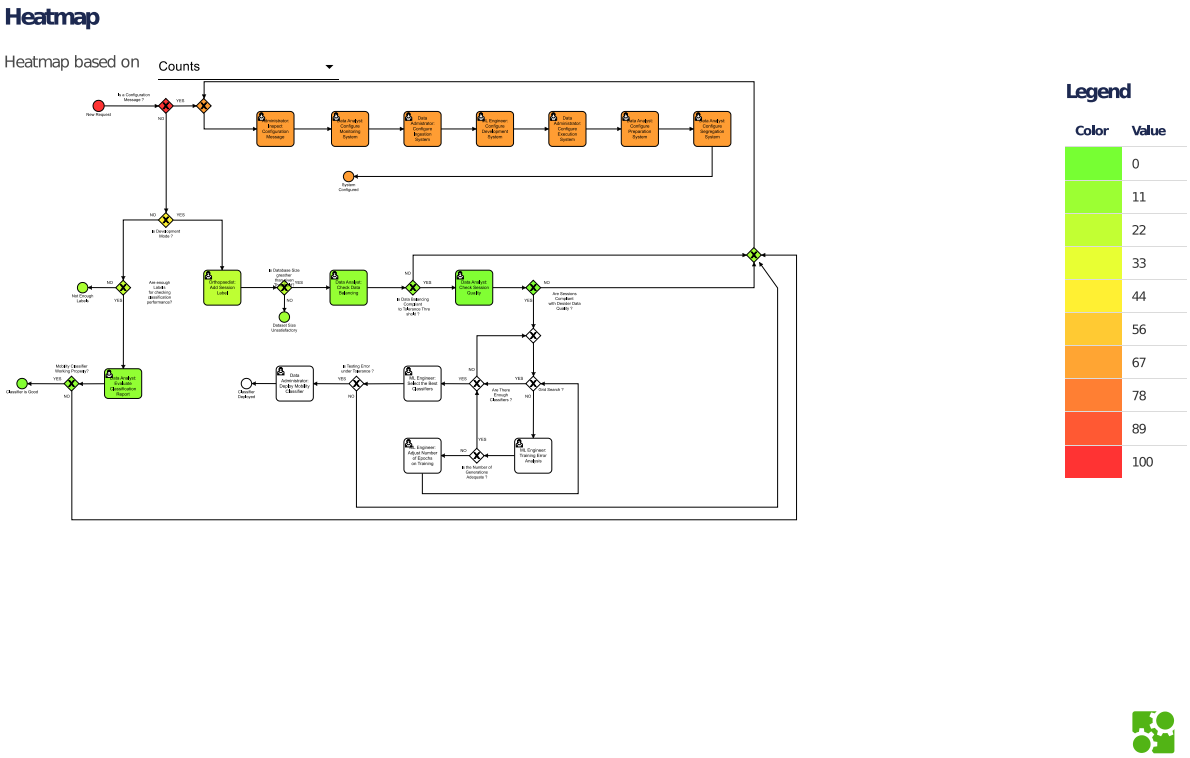
The simulation results of the TO-BE model demonstrate that the three performed improvements have reduced the maximum process instance time by **11%**, from 3.4m to 3m; while for the average process instance time the improvement is **12**%, from 22.6s to 19.9s.

5. PROCESS MINING

5.1. Normative Process Mining and Evaluation

5.1.1 Bimp Normative Log Generation for Mining

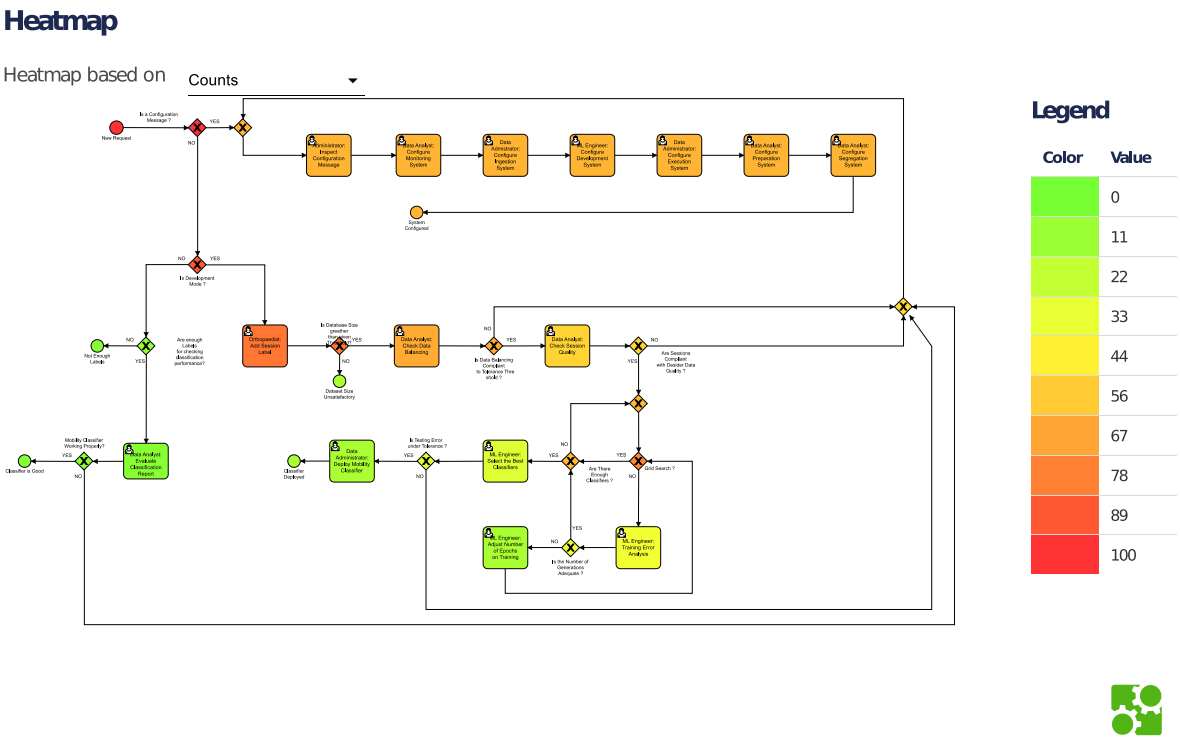
We calculated the heatmap and simulation log using the value of cost, resources and gateway percentage suggested on the vision document and we obtained the following result:



From the previous image we can notice that using these percentage split values for the gateways some scenario never happens, e.g. the human task “Deploy Mobility Classifier”. Since we want to distribute tokens to every task, we decide to modify percentage value of some gateways:

- “Is a configuration Message?” -> YES: 10% and NO: 90%
- “Is a Development Mode?” -> YES: 15% and NO: 85%
- “Is Dataset greater than given threshold?” -> YES: 10% and NO: 90%
- “Is Data Balancing Compliant to Tolerance Threshold?” -> YES: 90% and NO: 10%

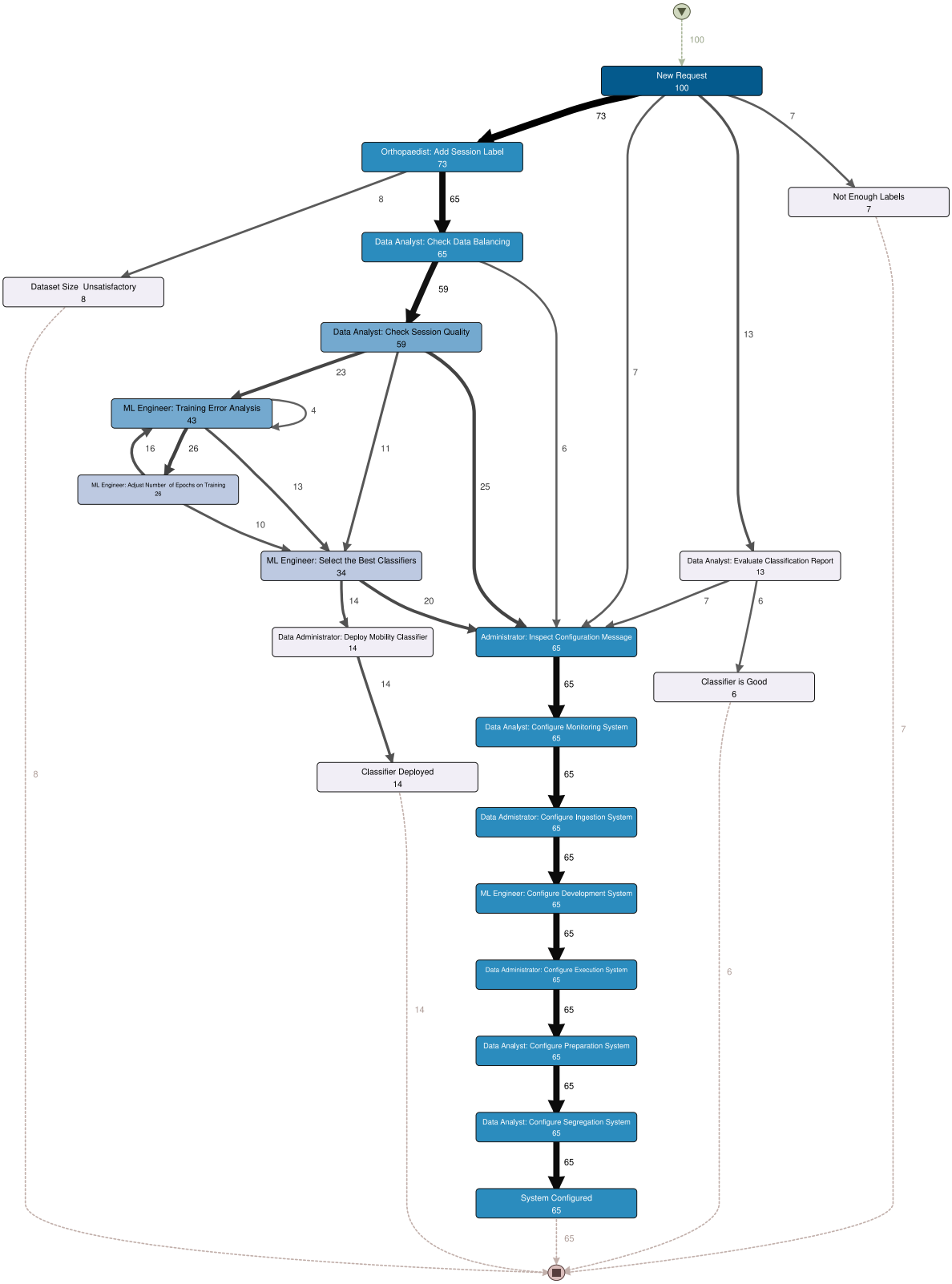
After doing these modifications we obtained the following result:



We observed now that every task has received at least one token.

5.1.2 Disco Normative Model

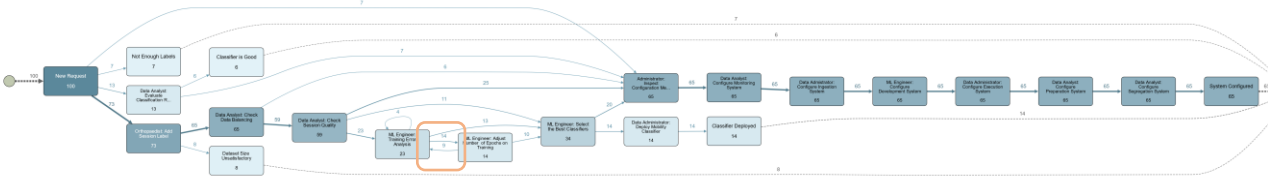
We imported on Disco the simulation log retrieved on BIMP and we obtained the following transition map:



Then we exported the log in CSV format adding endpoints.

We can observe that the transition map extracted from DISCO complies with the log generated by the simulation on the AS-IS model on BIMP.

5.1.3 Apromore Normative Model

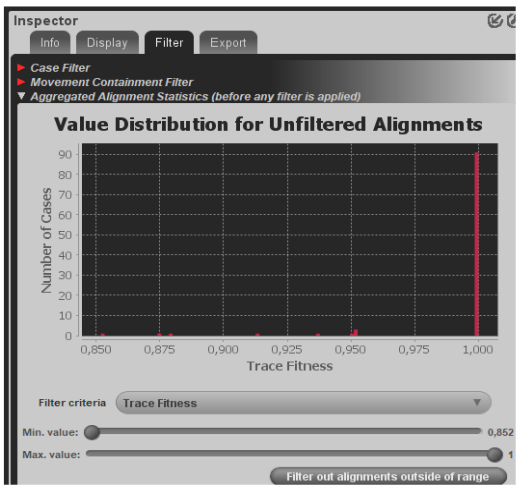


5.1.4 Comparison between Disco Transition Map and Apromore Process Map

Although there is a small difference in occurrences in the 2 Maps between the reciprocal workflows between the tasks “Training Error Analysis” and “Adjust Number of Epochs on Training” (orange box), attributable to the different mining algorithms used by the two software, the two models perfectly match.

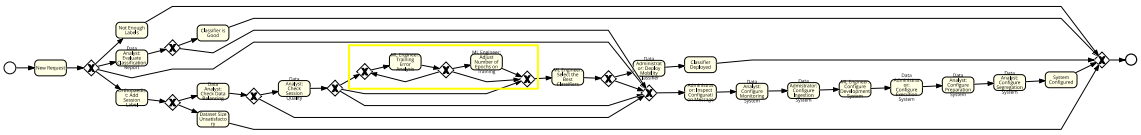
5.1.4.1 Conformance Checking on Apromore BPMN Model

Starting from the logs obtained through the simulation on the AS-IS model on BIMP, we have, via DISCO, converted the format from MXML to XES and, on Apromore, we have generated a BPMN model (figure below), in addition to the transition map shown above. We imported this BPMN model obtained from Apromore and the logs obtained from BIMP into ProM. On ProM we converted the BPMN diagram into Petri-Net and we performed the conformance checking. The results of the conformance checking are shown in the following figures.



Being the fitness equal to 0.99, we can observe that the log is practically successfully replayed.

Apromore BPMN Diagram



Case	Relevant Trace
18	Training Errors Analysis -> Adjust Number of Epochs on Training -> Training Error Analysis
56	Training Error Analysis -> Adjust Number of Epochs on Training -> Training Error Analysis -> Adjust Number of Epochs on Training -> Training Error Analysis -> Training Error Analysis -> Adjust Number of Epochs on Training -> Training Error Analysis -> Adjust Number of Epochs on Training -> Training Error Analysis -> Select the Best Classifier
73	Training Error Analysis -> Adjust Number of Epochs on Training -> Training Error Analysis -> Adjust Number of Epochs on Training -> Training Error Analysis -> Adjust Number of Epochs on Training -> Training Error Analysis -> Select the Best Classifier
46	Training Error Analysis -> Adjust Number of Epochs on Training -> Training Error Analysis -> Adjust Number of Epochs on Training -> Training Error Analysis
29 – 11	Training Error Analysis -> Adjust Number of Epochs on Training -> Training Error Analysis
72	Adjust Number of Epochs on Training -> Training Error Analysis -> Select the Best Classifier

All of those cases are considered Violations because does not exist the direct workflow from “Adjust Number of Epochs on Training” to “Training Error Analysis”.

5.1.5 ProM Normative Model

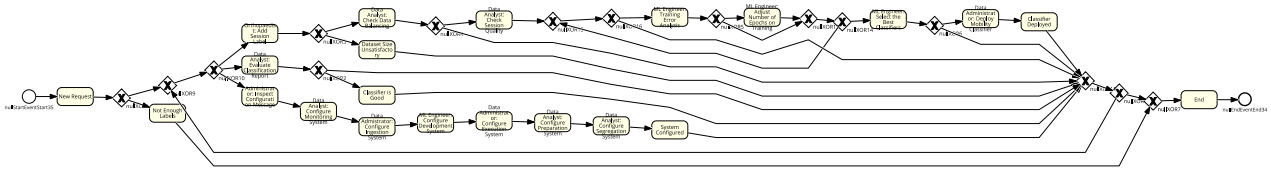
Starting from the BIMP mxml log file, we added the Endpoints (resulting in the addition of the only End Event) while converting the file to csv format on Disco.

We imported the CSV in ProM (with the related plugin) and converted it in XES by selecting only the time of process end; the resulting XES can be used as a log in ProM. Before doing the discovery, we apply on the XES file the following filters:

- Enhance Log: Remove all 'non-standard' attributes (In Place)
- Enhance Log: Repair Event Log (In Place)

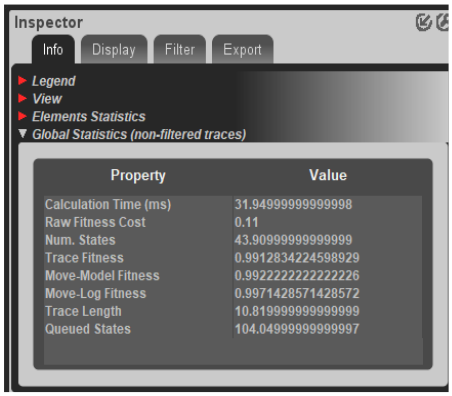
5.1.5.1 Discovery

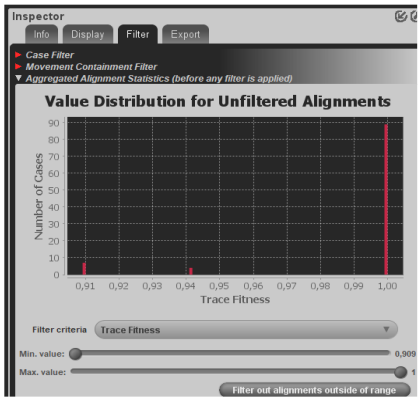
We applied on the XES file the **Inductive Miner Algorithm** and we obtained the following BPMN Diagram

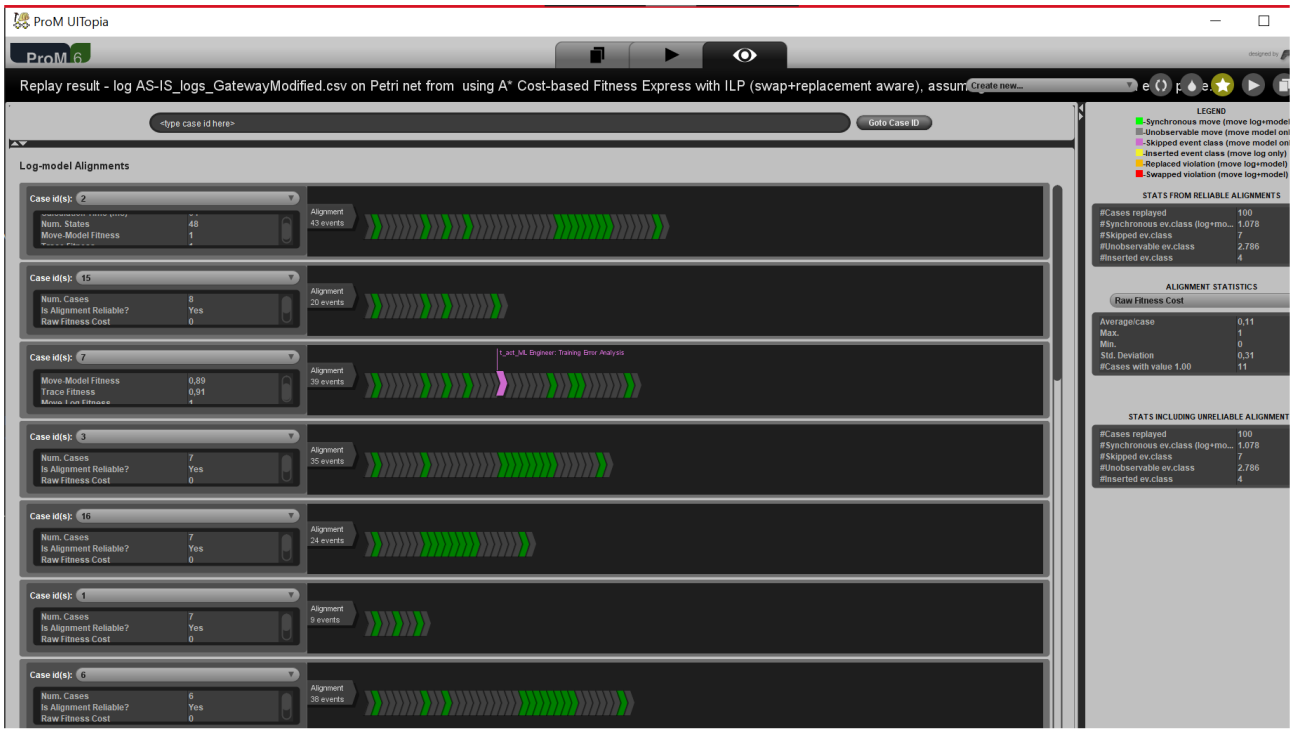


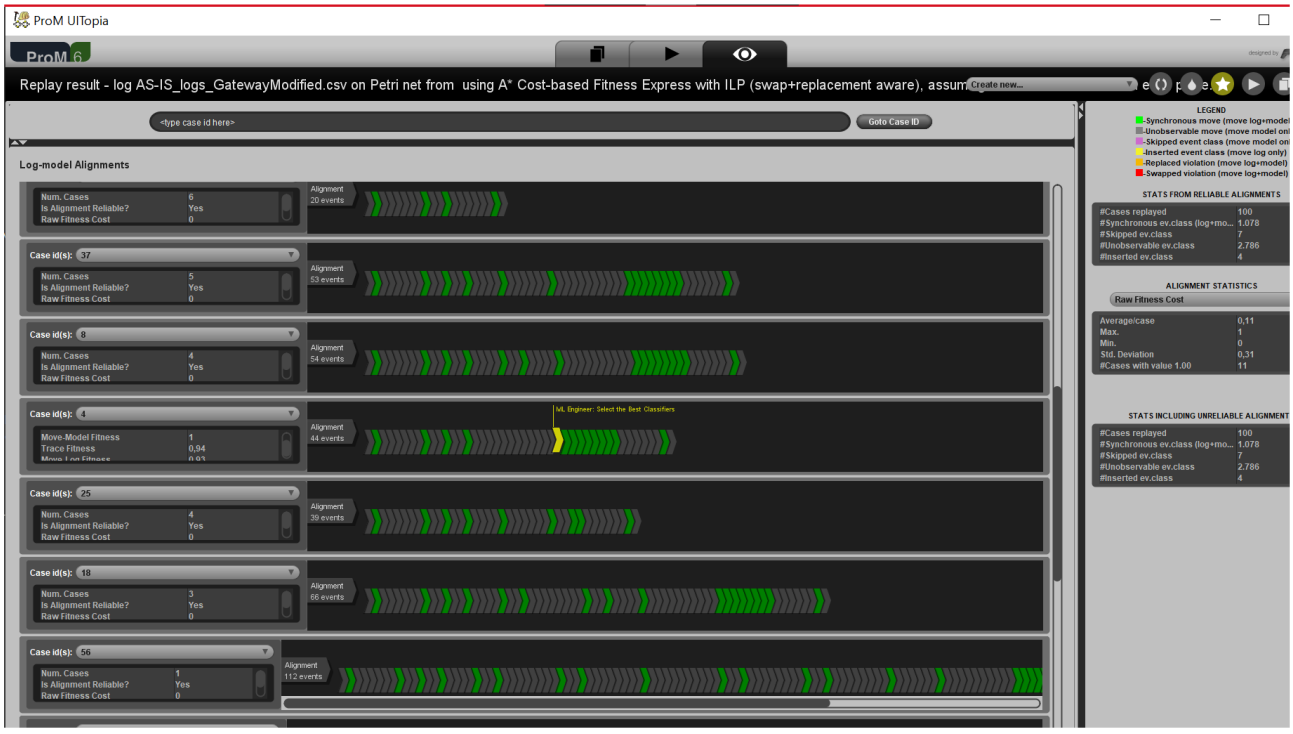
5.1.5.2 Conformance checking

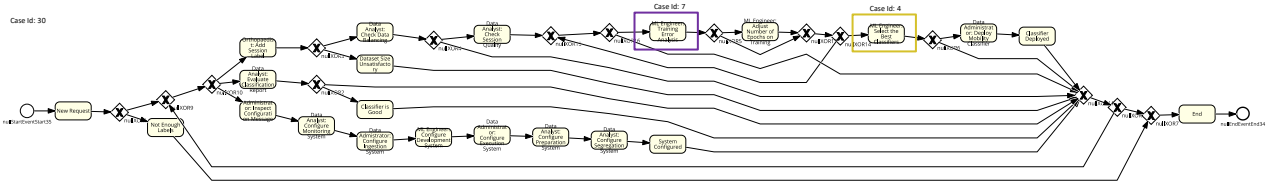
In order to do the conformance checking analysis we need to convert the BPMN Diagram into Petri Net Diagram. After that we executed "Replay a Log on Petri Net for Conform Analysis" using the Petri Net Diagram and the XES file as input and the results that are obtained are as follows:











Case	Relevant Trace	Color Legend
7	Check Session Quality -> Training Error Analysis -> Select Best Classifier	Skipped Event

4	Check Session Quality -> Select Best Classifier -> Inspect Configuration message	Inserted event

The two violations can be attributed to the problem that ProM Mined Model did not represent the direct workflow from “Check Session Quality” to “Select Best Classifier”.

Indeed, the first violation ‘skipped event’ would have fit the model if such direct workflow was present.

The same is true for the second violation because, missing such direct workflow, the ProM software chose the shortest path from “Check Session Quality” to “Inspect Configuration Message” and inserted the middle event as a violation.

5.1.5.3. Quality Dimension Analysis

The quality dimensions are four and beyond the fitness value there are other dimensions of quality which are the following:

86 workflows, 15 gateways

- **Simplicity (#gateways + #sequence flows + #activities) = 86**
- **Precision = 0,70633**
- **Generalization = 0,99851**

5.2. Normative Process Evaluation against Log Violations

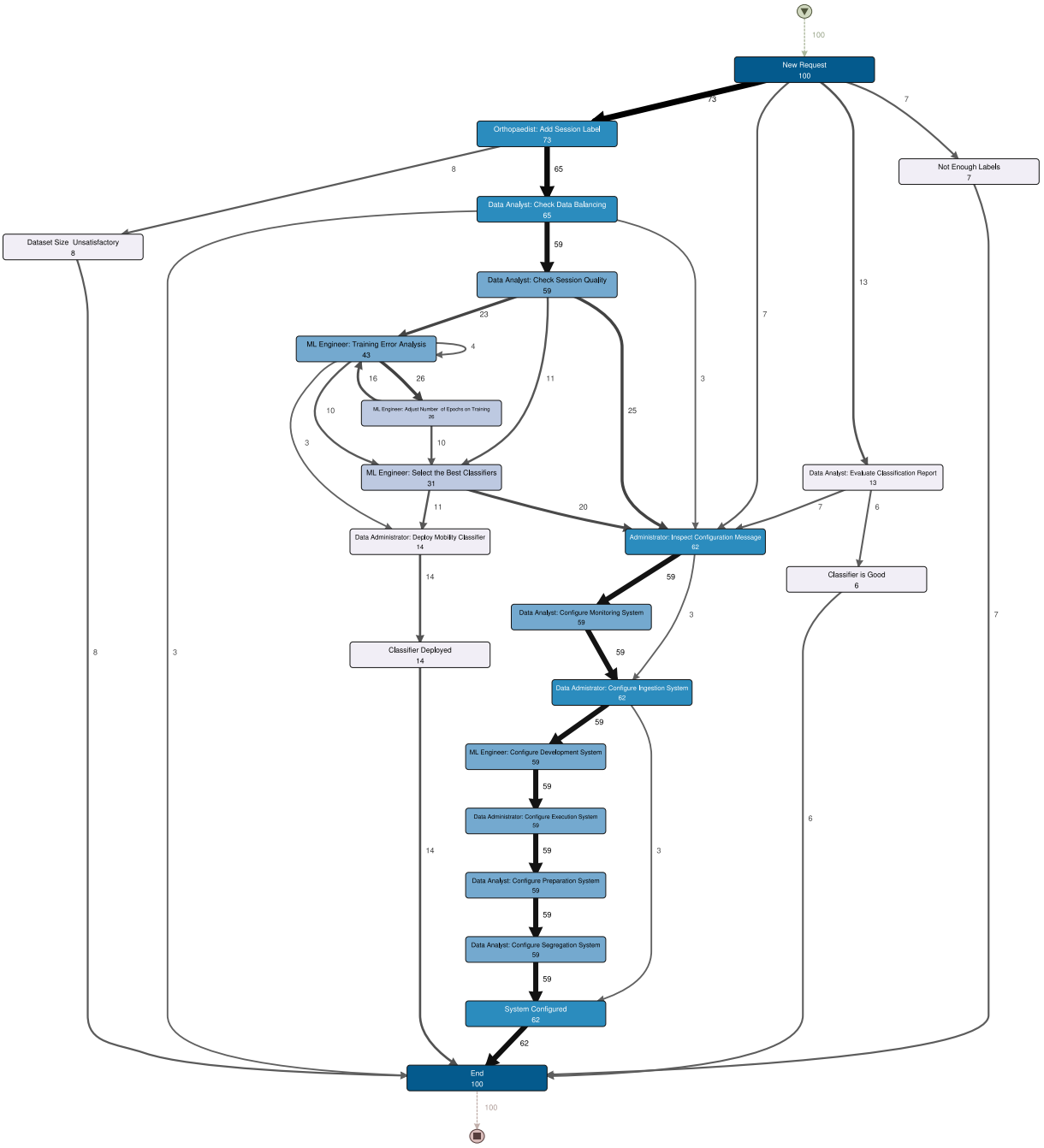
5.2.1. Introducing Log Violations

The log has been modified introducing three types of violations:

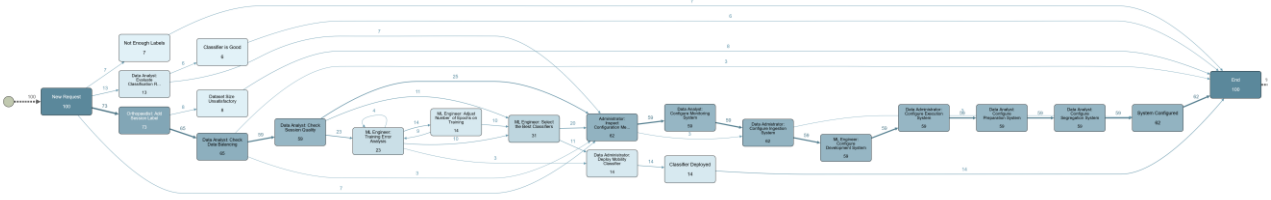
- The first emulates the behavior of a data analyst who, in case of unbalanced dataset, solves the issue by taking minor data from old sessions database stored in the segregation system instead of requesting new data by sending a message to the configuration system.
- The second emulates the behavior of a Machine Learning Engineer who trains the classifier choosing as hyperparameters the ones retrieved on the history of already developed classifiers, such that no grid search and selection of the best classifier must be performed.
- The third concerns configuration messages. There are situations where it is not necessary to configure all systems but only the ingestion system.

It is fair to remember that the violations introduced are legal and do not compromise the quality of service offered by the company.

5.2.2 Disco Model with Violations



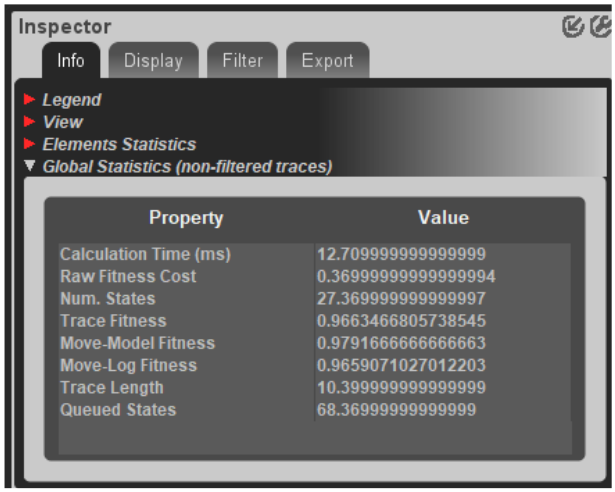
5.2.3. Apromore Model with Violations

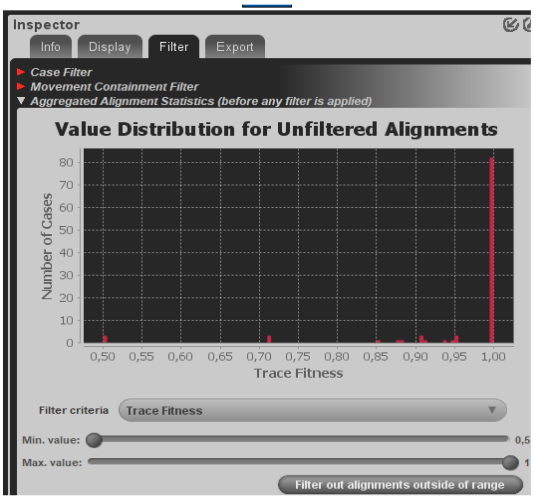


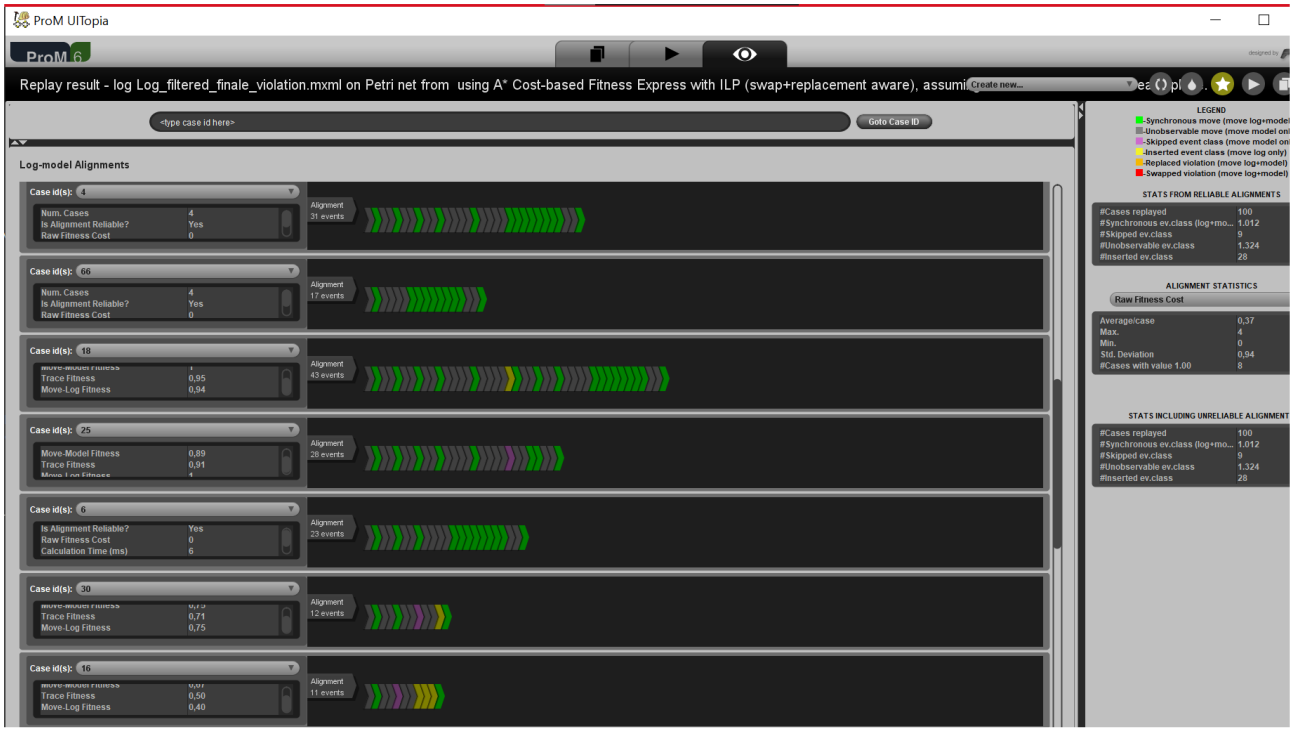
5.2.4. Comparison between Disco and Apromore Models with Violations

Although there is a small difference in occurrences in the 2 Maps between the reciprocal workflows between the tasks “Training Error Analysis” and “Adjust Number of Epochs on Training”, attributable to the different algorithms used by the two software, the two models perfectly match.

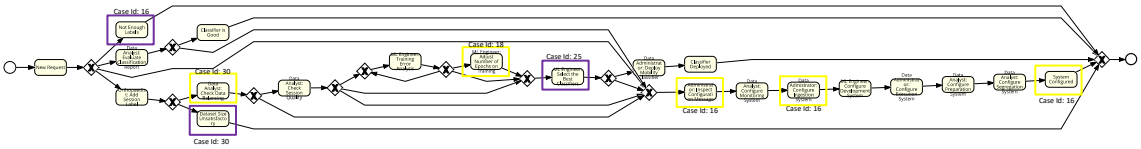
5.2.5. Conformance Checking Apromore Normative Model against Violations







Apromore BPMN Diagram

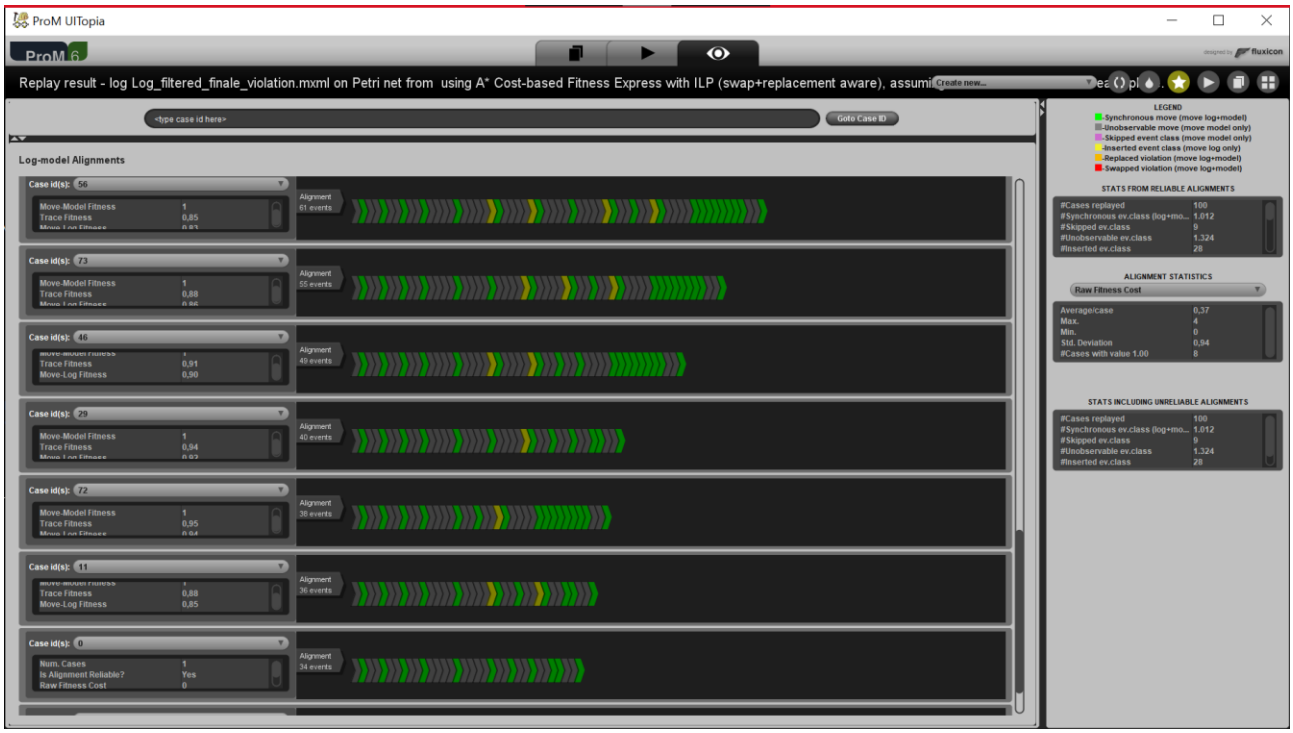


Case	Relevant Trace
18	Training Errors Analysis -> Adjust Number of Epochs on Training -> Training Error Analysis
25	Training Errors Analysis -> Select the best Classifier -> Deploy Mobility Classifier
16	New Request -> Not Enough Labels -> Inspection Configuration Message -> Configure Ingestion System -> System Configured -> End
30	New Request -> Add Session Label ->Dataset Size Unsatisfactory -> Check Data Balancing -> End

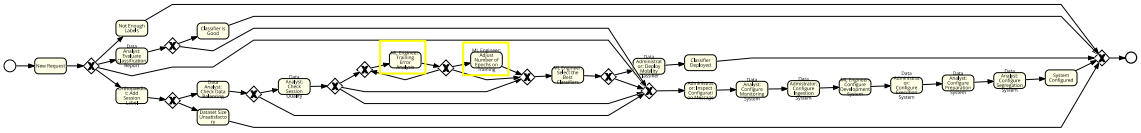
The Log Case 16 was modified to introduce the violation in the configuration process by skipping 5 configuration tasks. Indeed, it was modified to follow the following path: “New Request” -> “Inspect Configuration Message” -> “Configure Ingestion System” -> “System Configured” -> “End”.

We suppose that the problem regarding the event “Not Enough Labels” recognized as ‘skipped event’ (violet) and the events “Inspection Configuration Message”, “Configure Ingestion System” and “System Configured” recognized as ‘inserted events missing in the model’ (yellow) can be attributed to the intrinsicf functioning of the ProM Software itself.

Indeed, analyzing the path that Log case ID 16 would fit the most, once the ProM algorithm recognized the first (“new Request”) and last (“End”) events within the Model, it probably choses among the possible path connecting those events the one with the lowest number of Violations (and not the one with the highest number of matching). Since there exists the path with only one event (“Not Enough Labels”) it was chosen as the path of the Log Cases 16, thus labelling incorrectly the events as above.



Apromore BPMN Diagram



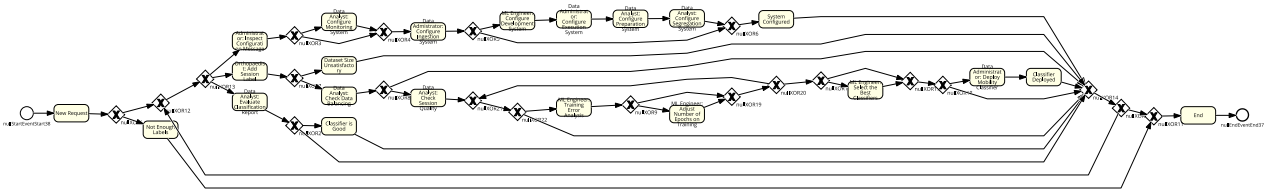
Case	Relevant Trace
56	Training Error Analysis -> Adjust Number of Epochs on Training -> Training Error Analysis -> Adjust Number of Epochs on Training -> Training Error Analysis -> Training Error Analysis -> Adjust Number of Epochs on Training -> Training Error Analysis -> Adjust Number of Epochs on Training -> Training Error Analysis -> Select the Best Classifier
73	: Training Error Analysis -> Adjust Number of Epochs on Training -> Training Error Analysis -> Adjust Number of Epochs on Training -> Training Error Analysis -> Adjust Number of Epochs on Training -> Training Error Analysis -> Select the Best Classifier
46	Training Error Analysis -> Adjust Number of Epochs on Training -> Training Error Analysis -> Adjust Number of Epochs on Training -> Training Error Analysis
29 – 11	Training Error Analysis -> Adjust Number of Epochs on Training -> Training Error Analysis
72	Adjust Number of Epochs on Training -> Training Error Analysis -> Select the Best Classifier

Most of those cases are considered Violations because does not exist the direct workflow from “Adjust Number of Epochs on Training” to “Training Error Analysis”.

5.2.5. ProM Model against Violations

5.2.5.1 Discovery

We applied on the XES file the Inductive Miner Algorithm and we obtained the following BPMN Diagram



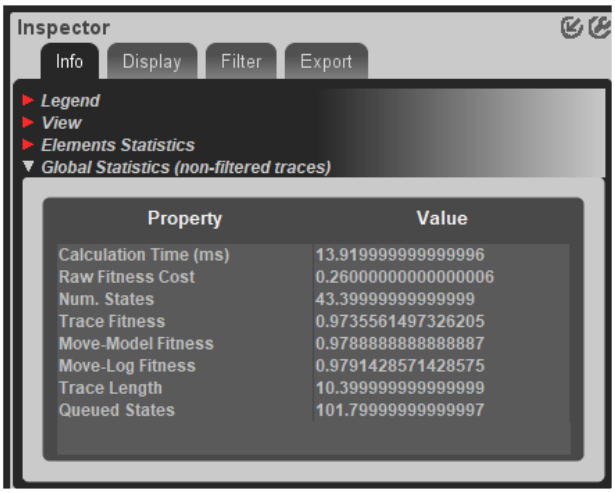
5.2.5.2. Conformance Checking

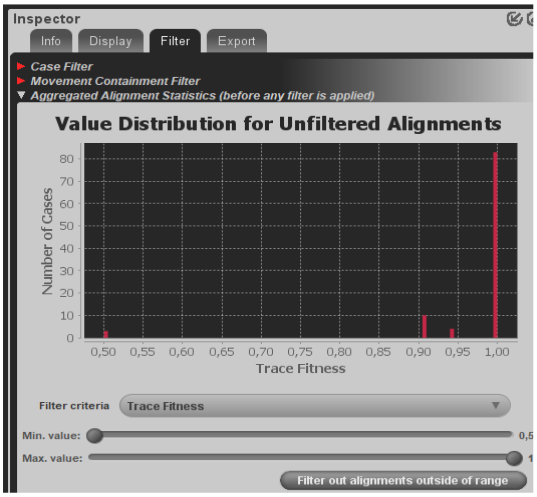
After modifying manually, the csv for introducing the violation, we performed conformance checking using the log with violations and the results we got are as follows:

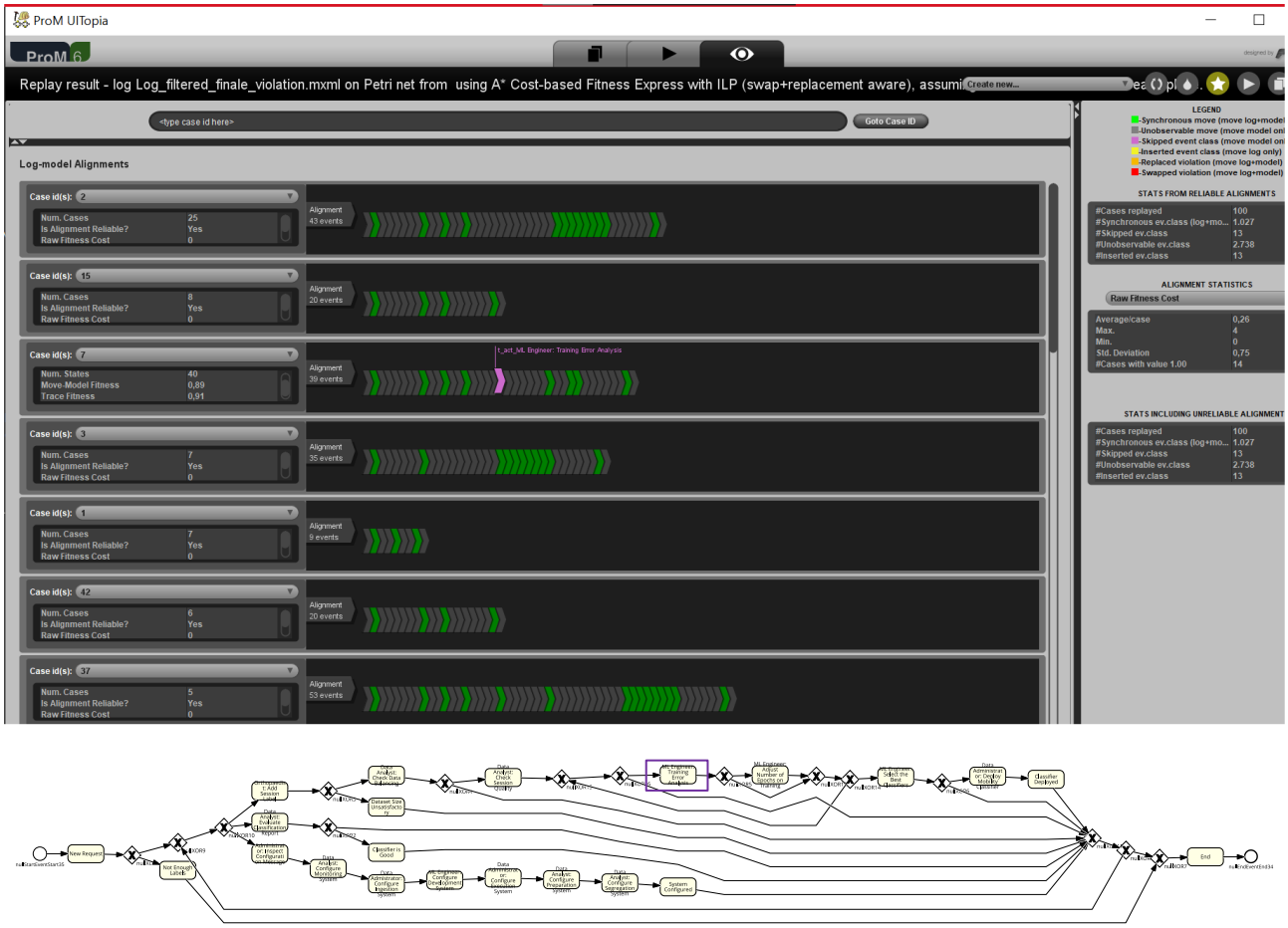
From the panel we can see that some violations have been revealed.

All violations were found where the classifier is used without choosing the best one. We can see the yellow arrow indicating the select best classifier event was expected in the diagram.

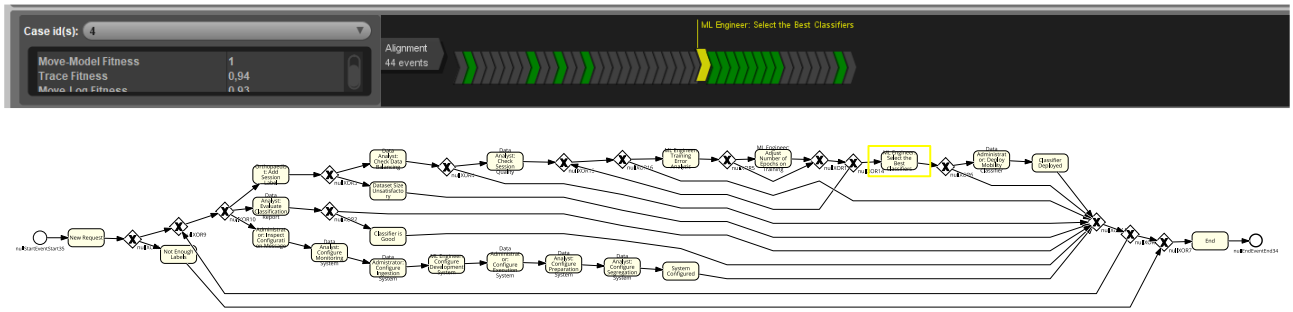
The violations related to the "check data balancing" task have not been identified (red square). From the panel we notice that the arrows are green despite this behavior of the token was not accepted in the as-is model. We can find the answer by looking at the BPMN diagram obtained on ProM. The diagram has introduced many XOR leading to generalize and therefore accept any unwanted behavior of the company's users. This can be dangerous because a high generalization of the model could hide dangerous user's violations. The yellow arrow we see in the previous picture shows the violations introduced regarding the configuration of the various systems.



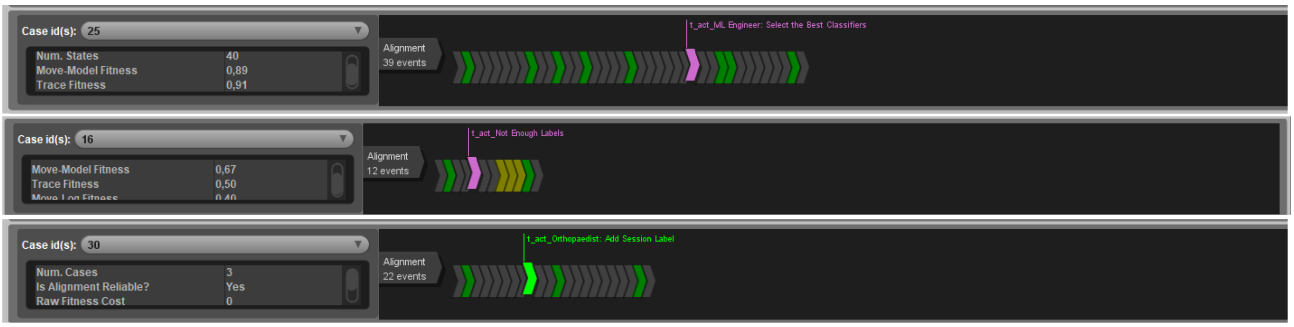


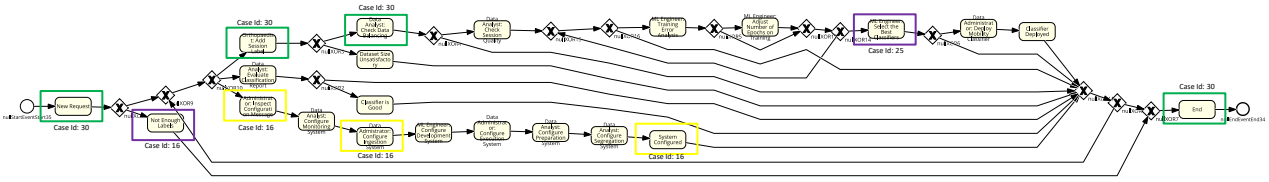


Case	Relevant Trace
7	Check Session Quality -> Training Error Analysis -> Select Best Classifier



Case	Relevant Trace
4	Check Session Quality -> Select Best Classifier -> Inspect Configuration message





Case	Relevant Trace
25	Training Errors Analysis -> Select the best Classifier -> Deploy Mobility Classifier
16	New Request -> Not Enough Labels -> Inspection Configuration Message -> Configure Ingestion System -> System Configured -> End
30	New Request -> Add Session Label -> Check Data Balancing -> End

Same explanation of the paragraph 5.2.5.

5.2.5.3. Quality Dimension Analysis

The quality dimensions are four and beyond the fitness value there are other dimensions of quality which are the following:

- **Simplicity = 101**
- **Precision = 0,71000**
- **Generalization = 0,99831**

21 gateways, 58 workflows

5.3 Evaluate Models Quality Dimensions

In this paragraph, the values of the four quality dimensions of the mined models have been reported.

The first section we compared the apromore and ProM (inductive miner) models using the log obtained from the as-is model.

The second section concerns a comparison for the same models seen in the previous section but using the log with some violations introduced.

The last section is about a comparison of apromore (inductive miner) models.

5.3.1. Comparison between Apromore and ProM Models Quality Dimensions

	Simplicity	**Precision**	**Generalization**	**Fitness**
ProM Normative Model	86	0.7063	0.99851	0.9912
Apromore Normative Model	74	0.9274	0.99862	0.9926

Looking at the table we can see that the Apromore model is less complex and more precise than the ProM mined model. This means that the apromore model would avoid behavior not seen by the event logs.

While for Generalization and Fitness the values are more or less equal. So, we can say that the model given by the Apromore software results to be better than the ProM model.

5.3.2. Comparison between Apromore and ProM Models Quality Dimensions with Modified Log

The models are the same as in the previous section, but the logs are different. Violations have been introduced in the logs in order to observe the variation of the quality quantities of the models. The conformance analysis of the models takes these modified logs as input.

	Simplicity	**Precision**	**Generalization**	**Fitness**
ProM Normative Model	86	0.71	0.99831	0.9735
Apromore Normative Model	74	0.9282	0.99853	0.9663

Although the fitness decreased for both models, but the Apromore model continues to have higher accuracy than the ProM model.

5.3.3. Comparison between ProM with and without Violations Models Quality Dimensions

	Simplicity	**Precision**	**Generalization**	**Fitness**
ProM Normative Model	86	0.71	0,99831	0.9735
ProM Improved Normative Model	101	0.59635	0.99808	0.9912

5.3.4. Comparison between Apromore with and without Violations Models Quality Dimensions

	Simplicity	**Precision**	**Generalization**	**Fitness**
Apromore Normative Model	74	0.9282	0.99853	0.9663
Apromore Improved Normative Model	92	0.73416	0.99810	0.9926