UNIVERSITY OF PISA
Master's Degree in Computer Engineering

Anomaly Detection In Manufacturing

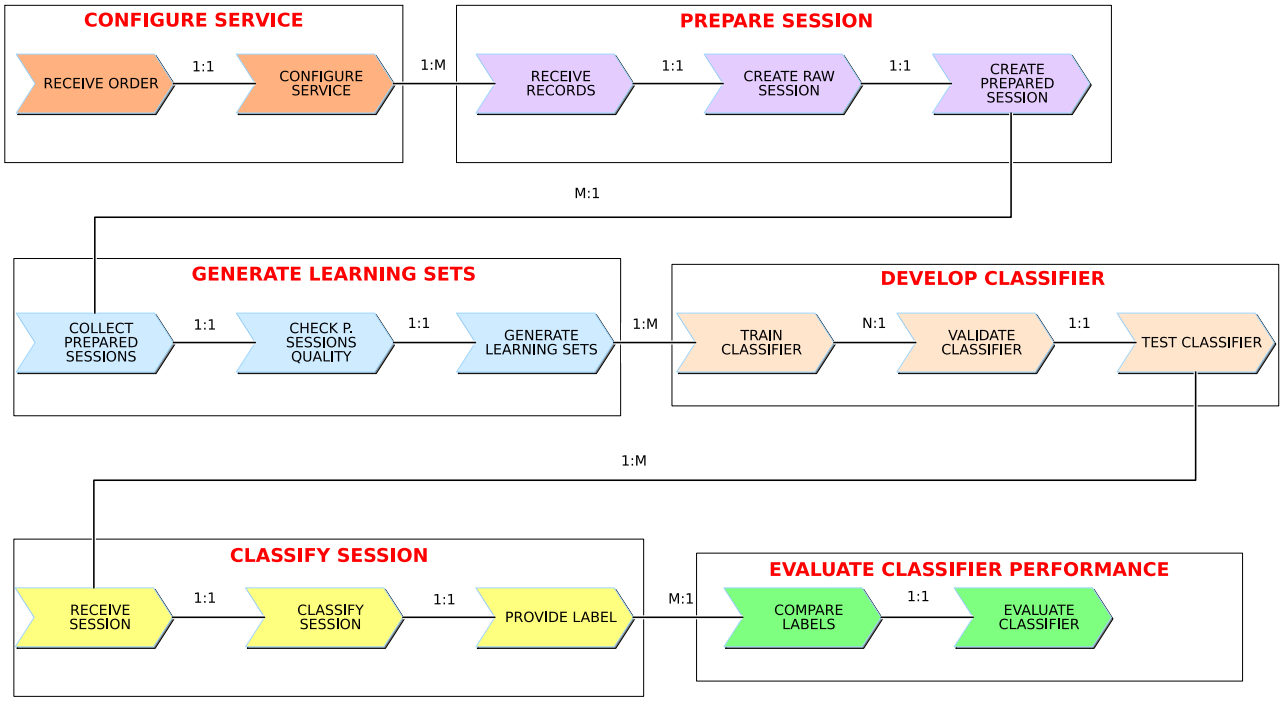
Software Systems Engineering

BARBIERI GIOVANNI
CAVEDONI FEDERICO
DI RICCO ALESSIO
SALTI NICOLÒ
TRASACCO ANDREA

A.Y. 2023-24

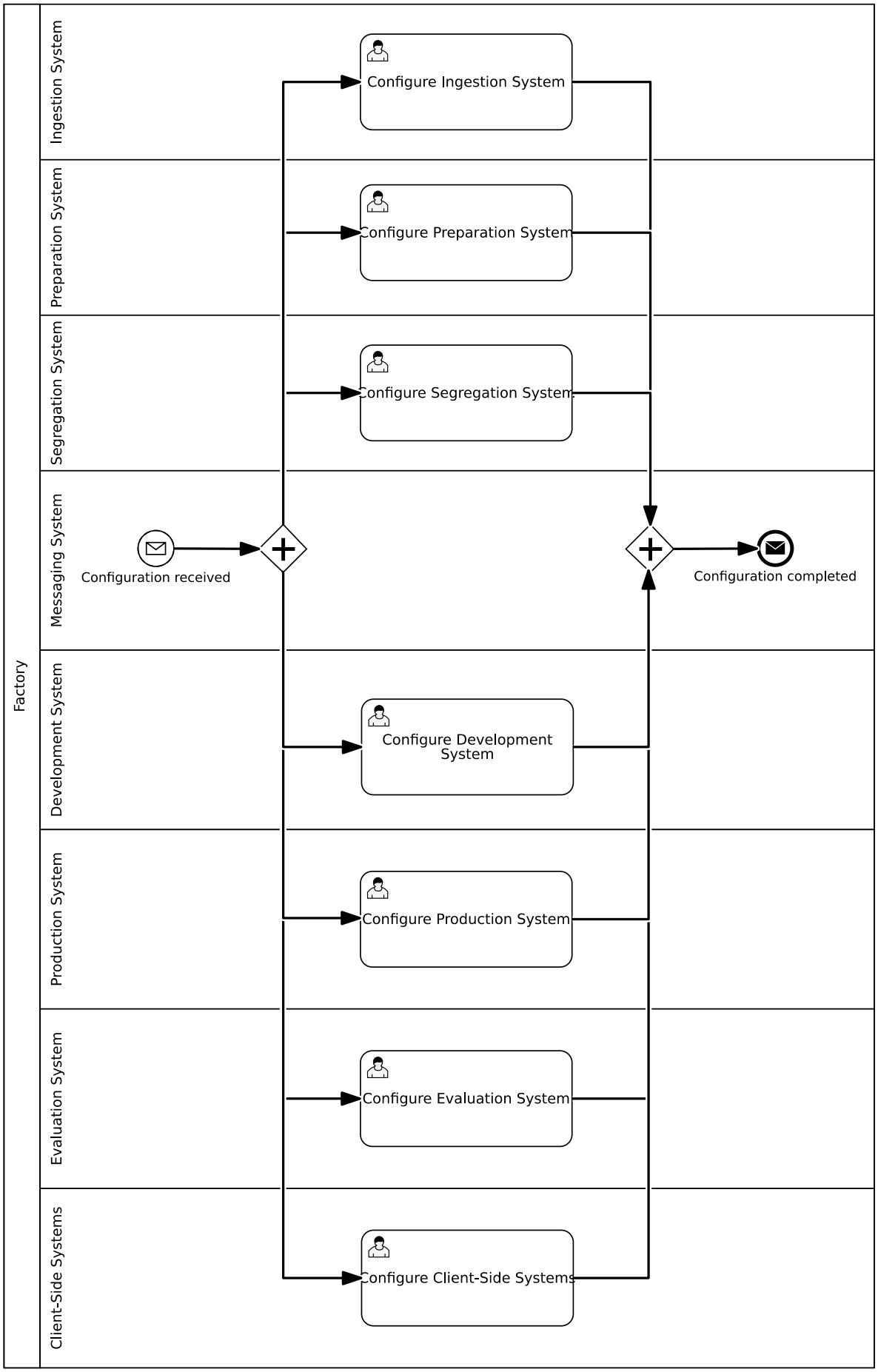
Sommario

PROCESS LANDSCAPE (Barbieri, Cavedoni, Di Ricco, Salti, Trasacco)

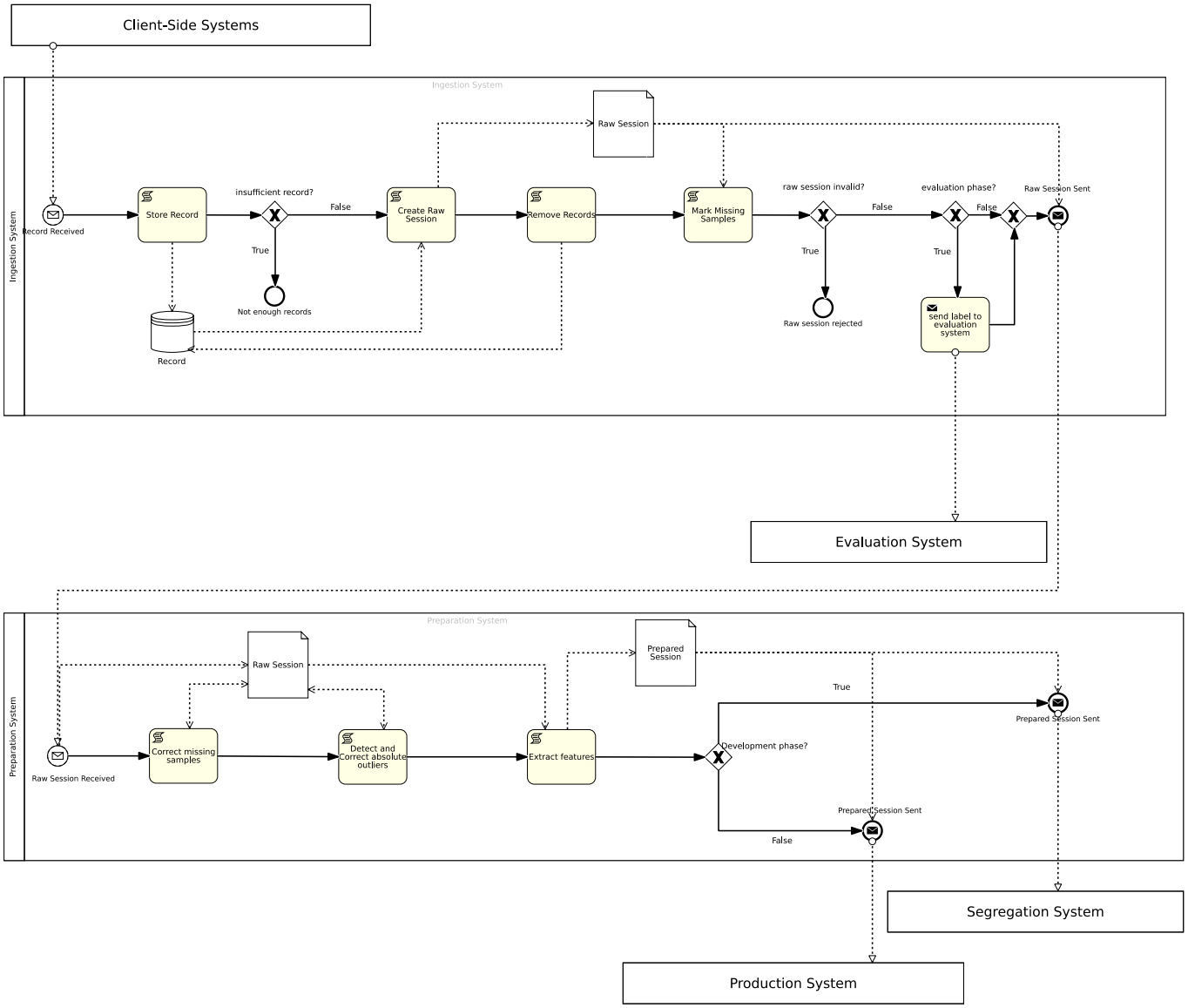


BPMN Modeling

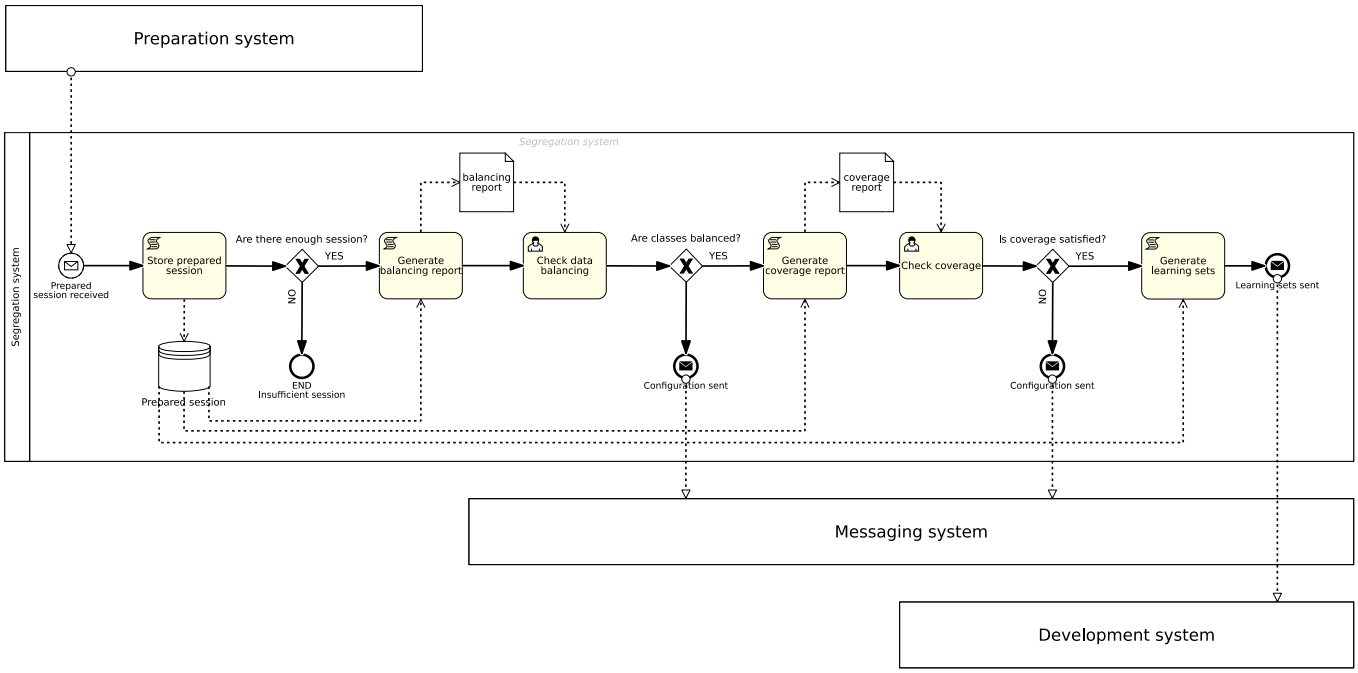
Process CONFIGURE SYSTEMS (Barbieri, Cavedoni, Di Ricco, Salti, Trasacco)



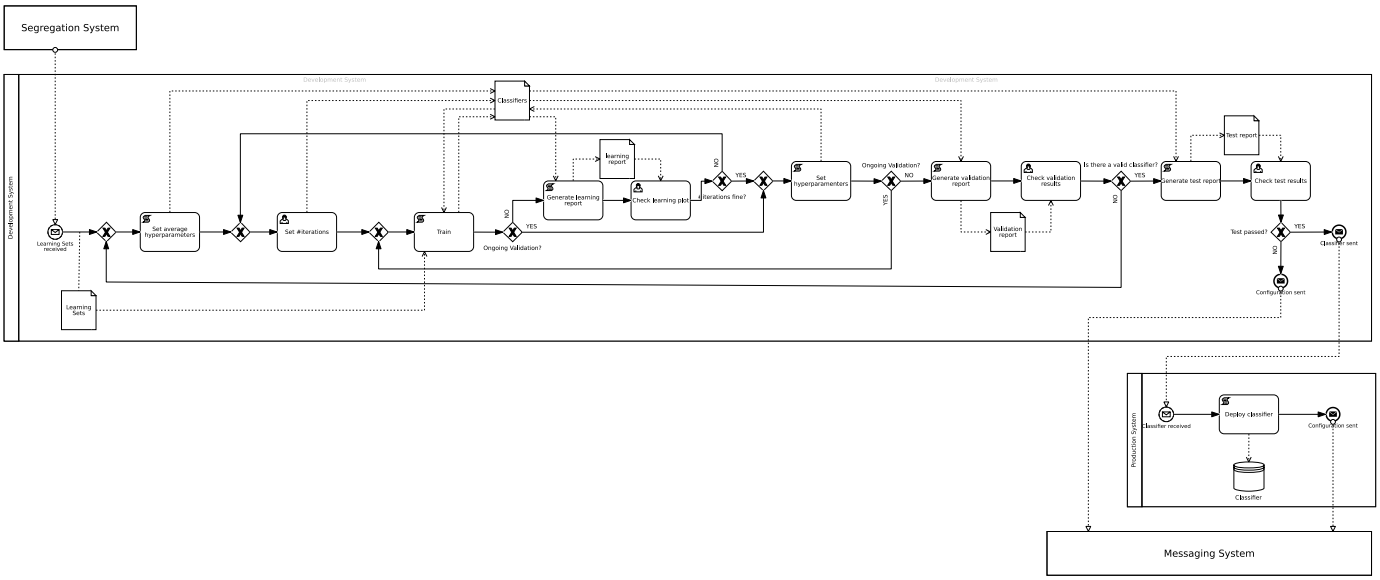
Process PREPARE SESSION (Barbieri)



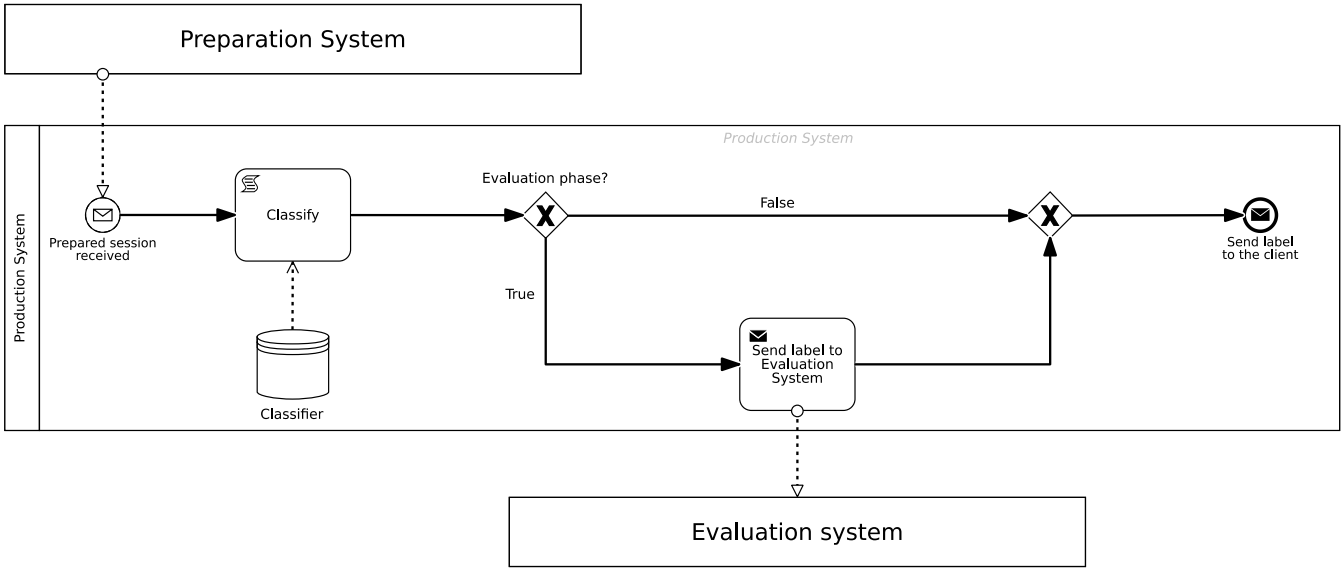
Process GENERATE LEARNING SETS (Di Ricco)



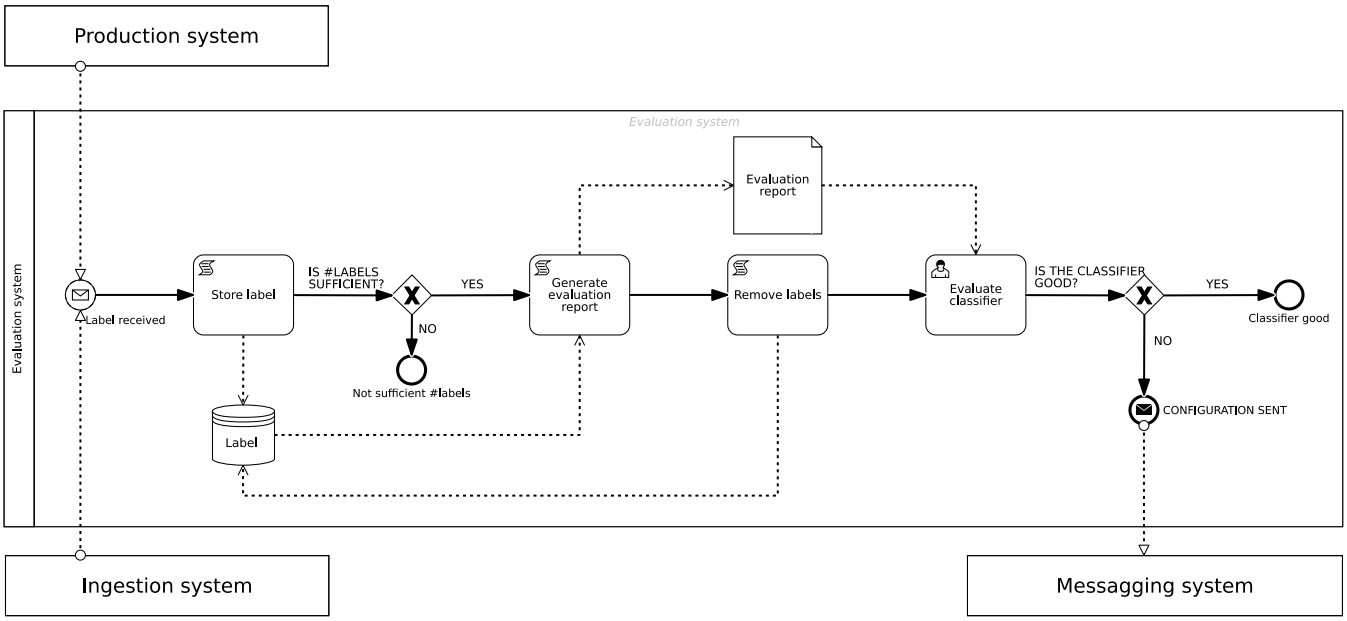
Process DEVELOP CLASSIFIER (Trasacco)



Process CLASSIFY SESSION (Cavedoni)



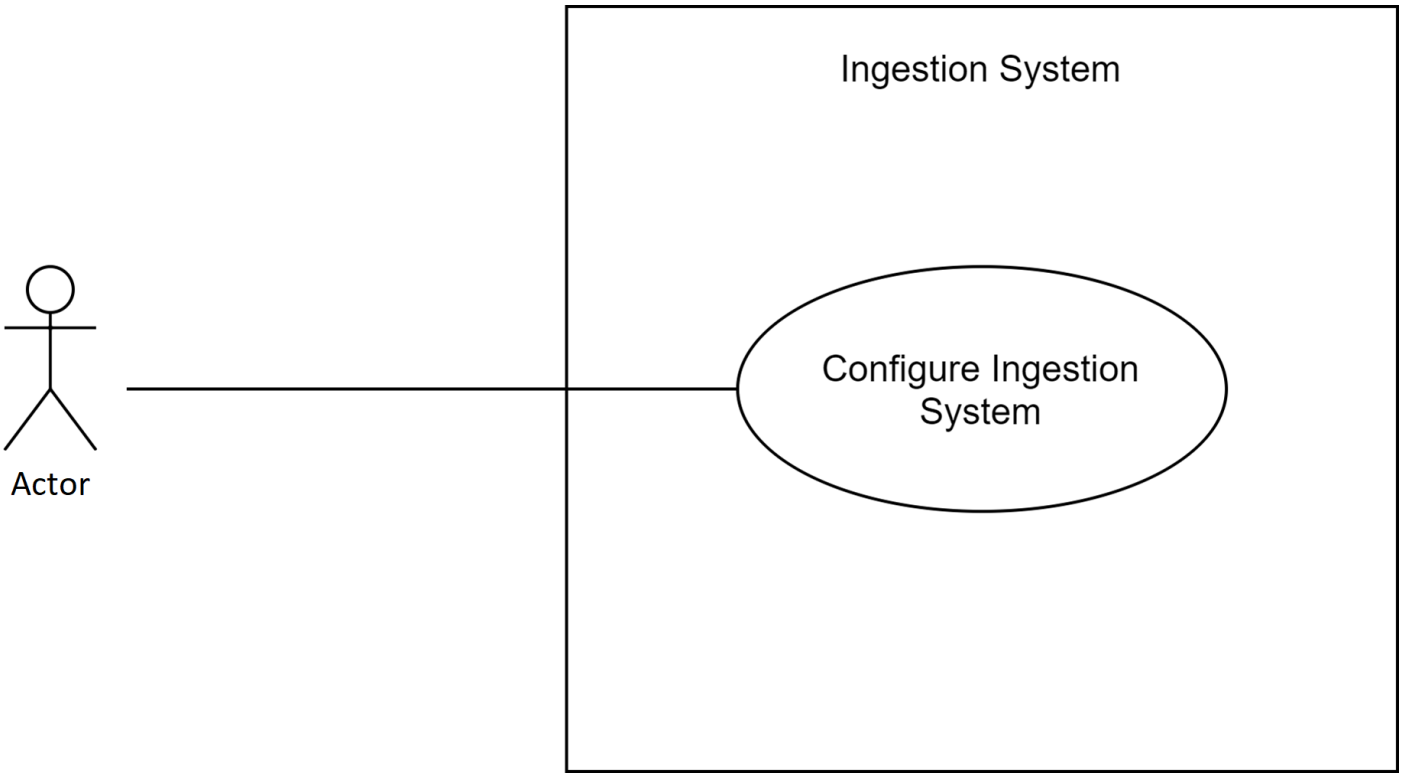
Process EVALUATION CLASSIFIER PERFORMANCE (Salti)



ANALYSIS – Human Tasks – UML Use Case Diagrams

Ingestion system

Human Task: Configure Ingestion System (Barbieri)

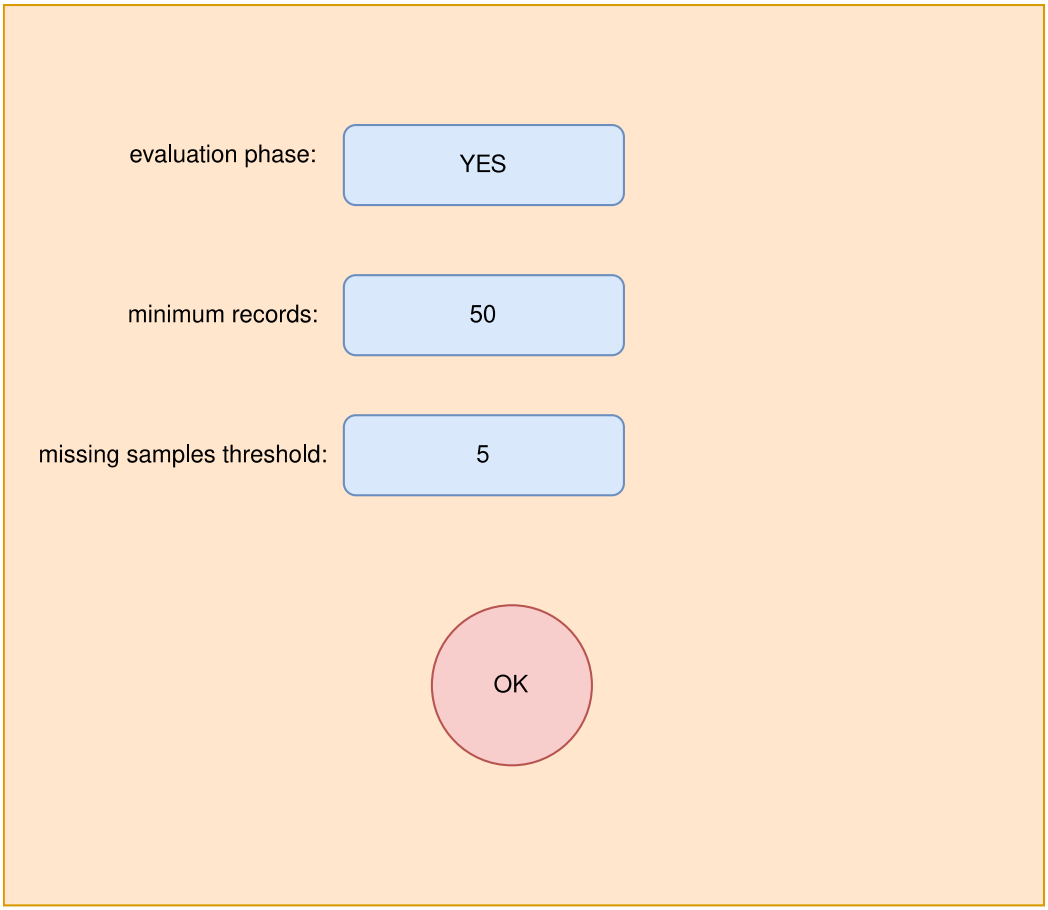


Scenario

1. The use case starts when Actor opens the interface to configure the ingestion system
2. for each parameter the Actor digits the value in the specific input field
3. The Actor press the OK button

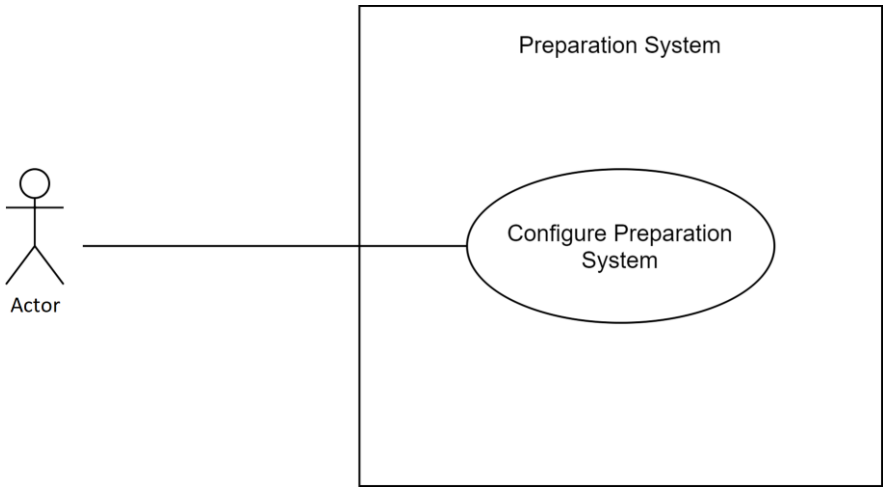
Details

Name	Value
Postconditions	Ingestion System has the parameters updated



Preparation system

Human Task: Configure Preparation System (Barbieri)

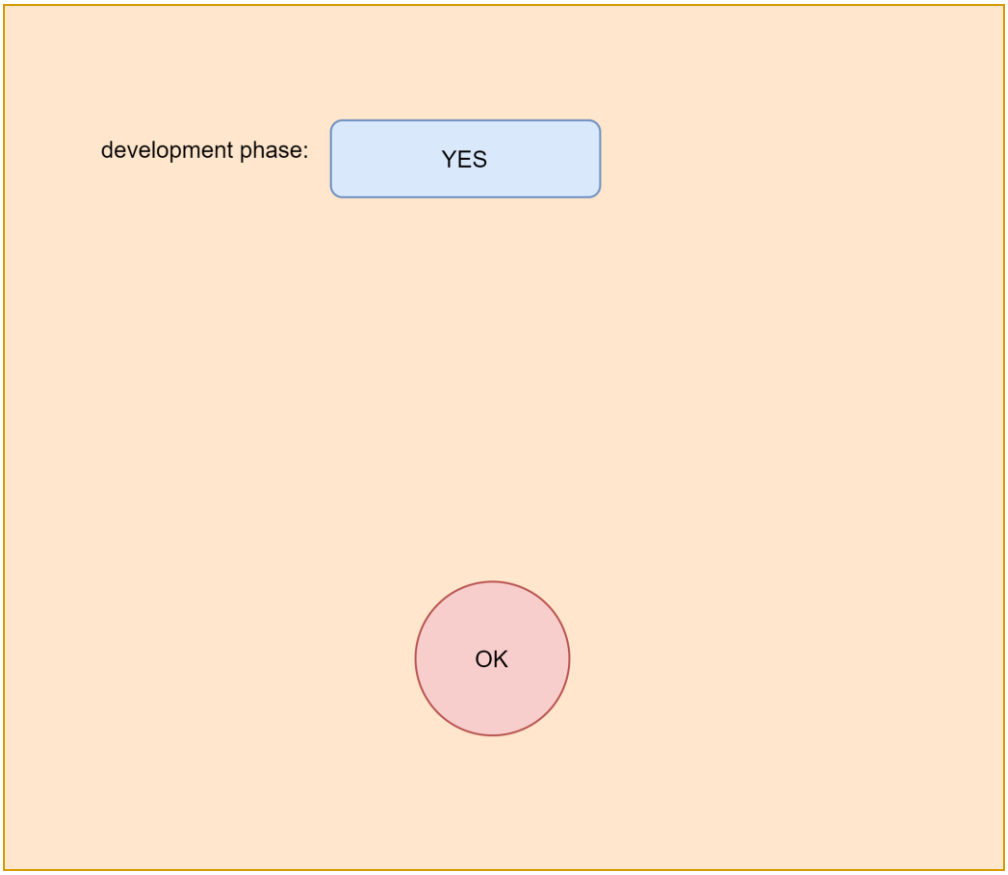


Scenario

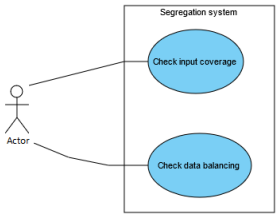
1. The use case starts when the Actor opens the interface to configure Preparation System
2. for each parameter the Actor digits the value in the specific input field
3. The Actor press the OK button

Details

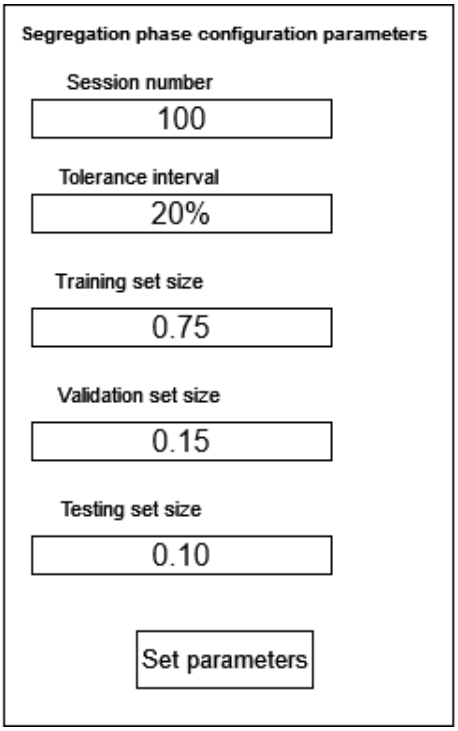
Name	Value
Postconditions	Preparation System has the parameters updated



Segregation system



Human Task: Configure Segregation System (Di Ricco)

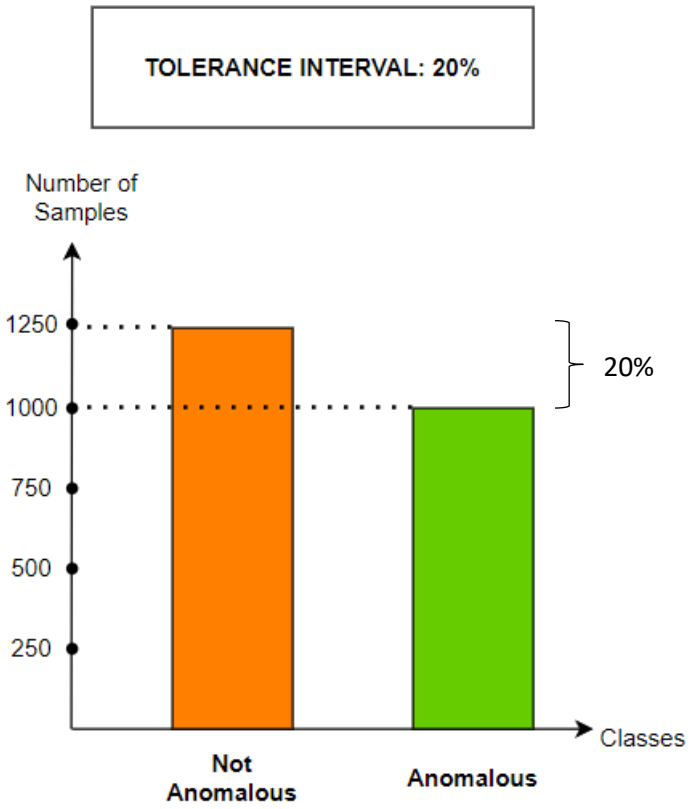


Scenario

1.The use case starts when the system opens the configuration panel
2. The Actor sets the number of prepared session that the system will receive
3. The Actor sets the tolerance interval
4. The Actor sets the training set size
5. The Actor sets the validation set size
6. The Actor sets the testing set size
7. The Actor press the button “Set parameters”

Details

Name	Value
Post-conditions	The Segregation System has been configured

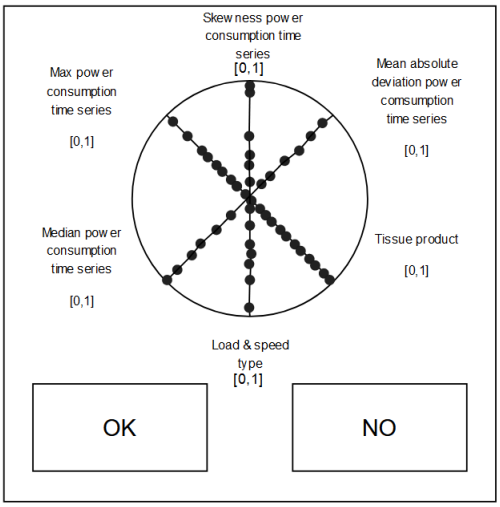


Scenario

1. The Actor launches the img viewer
2. The Actor opens the plot that display the data balancing
3. The Actor evaluate if the data are well balanced between the classes
4. The Actor close the plot
5. if the data are balanced
5.1 The Actor enter “balanced” in the file “balancingEvaluation.json”
6. else
6.1 The Actor enter “not balanced” in the file “balancingEvaluation.json”
6.2 A new configuration is requested
7.end if

Details

Name	Value
Preconditions	The application has produced a plot to represent the data distribution
Preconditions	The application has saved the plot in a png file called "data_balancing.png"
Preconditions	The application stops waiting for user input



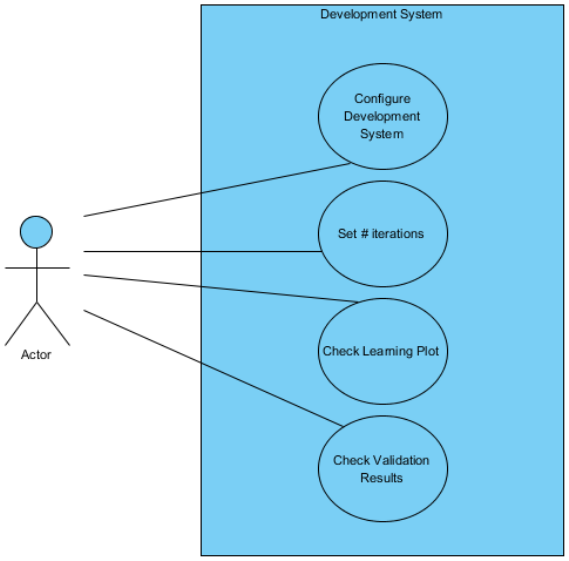
Scenario

1.The actor launch the application with the balancing report
2.The actor analyze the report and check whether the input space is uniformly covered
3.FOR EACH features
3.1.IF the distribution is not uniform
3.1.1.The actor click NO button
3.1.2.The Actor request a new Data Configuration
3.1.3.The Actor close the balancing report
4. end for each
5.The Actor click OK button

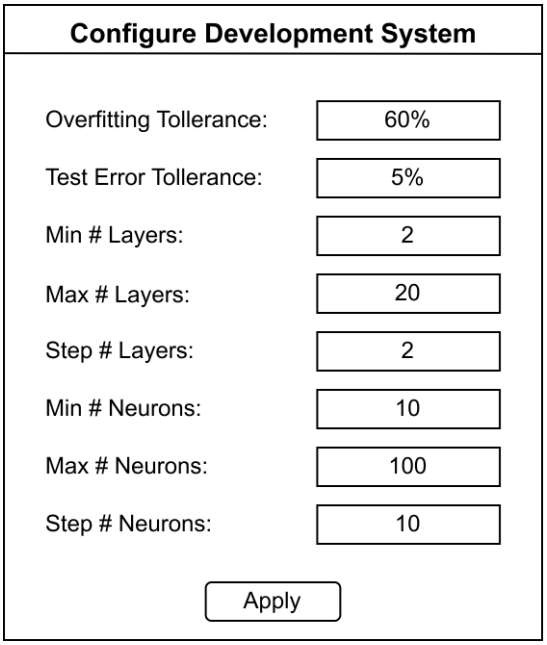
Details

Name	Value
Preconditions	Application has produced a balancing report Application has stopped waiting for user input command
Post-conditions	Learning sets are generated.

Development system



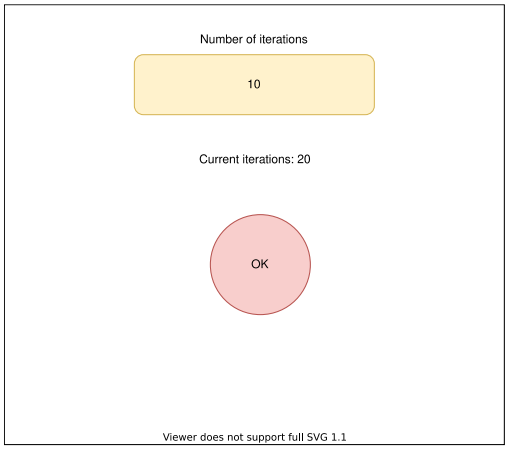
Human Task: Configure Development System (Trasacco)



Scenario

1.The use case starts when the System shows the panel for the configuration.
2. for each parameter to be configured
2.1.The System shows the name of the parameter and an input field for its value
end for each
3. for each parameter to be configured
3.1.The Actor inserts the value of the parameter in the correspondent input field.
end for each
4.The Actor clicks on the Apply button.
5.The System reads the values of the parameters and closes the window.

Human Task: Set #Iterations (Barbieri)



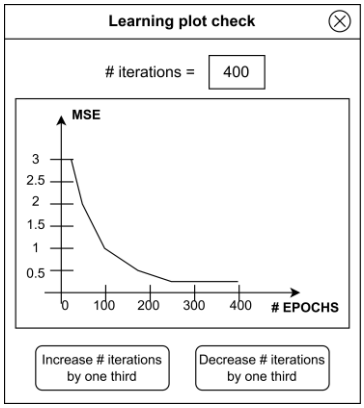
Scenario

1.The use case starts when The Actor opens the interface for setting the number of iterations
2.The Actor digits the number of iterations in the "Number of iterations" field
3.The Actor presses the "OK" button

Details

Name	Value
Post-conditions	The development system updates the number of iterations

Human task: Check Learning Plot (Trasacco)



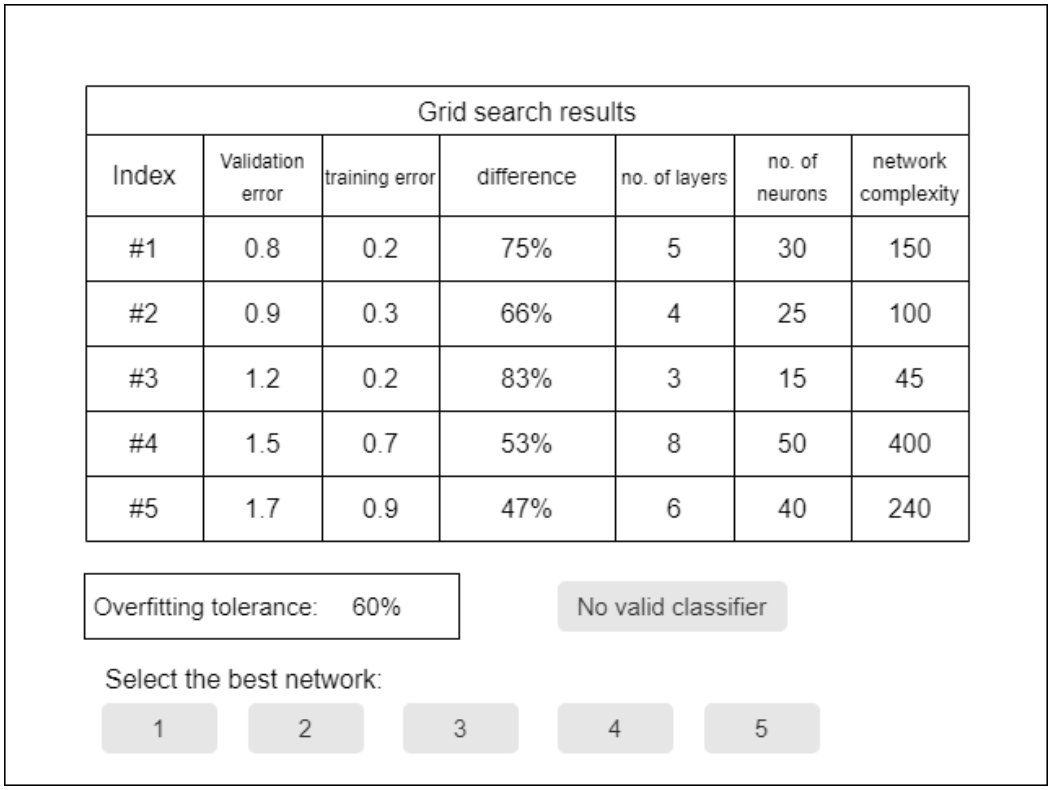
Scenario

1.The use case starts when the Actor clicks on "Check the learning plot".
2.The System shows a window with the learning error plot (MSE) against the number of iterations.
3. if the Actor sees that the loss is flat for at least the half of the iterations then
3.1.The Actor reduces by one third the number of iterations via the proper button.
3.2.The System updates the new value for the number of iterations.
3.3.The Actor closes the window using the "X" button.
4. else if the Actor sees that the loss is not flat at the end of the iterations then
4.1.The Actor enlarges by one third the number of iterations via the proper button.
4.2.The System updates the new value for the number of iterations.
4.3.The Actor closes the window using the "X" button.
5. else
5.1.The Actor closes the window using the "X" button.
end if

Details

Name	Value
Preconditions	The training completed. The learning report has been generated and it is visible.
Post-conditions	The number of iterations has been updated.

Human task: Check validation results (Salti, Trasacco)



Scenario

1.The use case starts when the System shows the Validation Report with the top 5 classifiers.
2. for each classifier in the report
2.1.The System displays validation error, training error, no. of layers, no. of neurons, network complexity
2.2.The System computes the difference between validation and training error.
end for each
3.The System displays the overfitting tolerance
4. for each classifier in the report
4.1.The Actor checks if the difference is under the overfitting tolerance
end for each
5. if The Actor notices that there are no classifiers which satisfy the tolerance
5.1.The Actor clicks on "No valid classifier"
6. else
6.1.The Actor chooses the best network among the selected ones based on the value of the difference, the no. of layers, the no. of neurons.
6.2.The Actor clicks on the button correspondent to the index of the chosen classifier.
end if

Details

Name	Value
Preconditions	The validation completed. The validation report has been computed.
Post-conditions	The best network is chosen, if any.

Generalization Tolerance: 20%

Winner Network	Validation Error	Test Error	Difference
Network 1	0.78	0.60	0.18

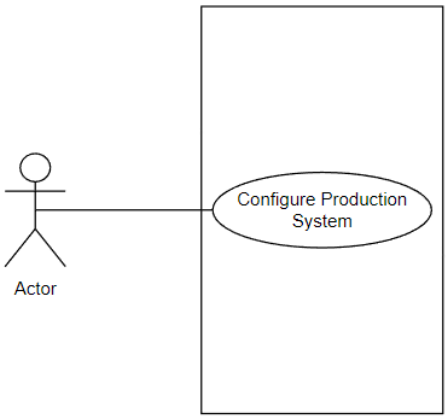
Scenario

1.The Actor launches the Image viewer
2.The Actor opens the file “bestClassifierReport.png”
3. The Actor evaluates the classifier using the data from the report file
4. The Actor close the Image viewer
5. if The difference between validation and testing error is less than the tolerance interval
5.1 The Actor updates the json file “classifierEvaluation.json” with the positive response
5.2.The System send the classifier to the production system
6. else
6.1 The Actor updates the json file “classifierEvaluation.json” with the negative response
6.2.The System request a new configuration

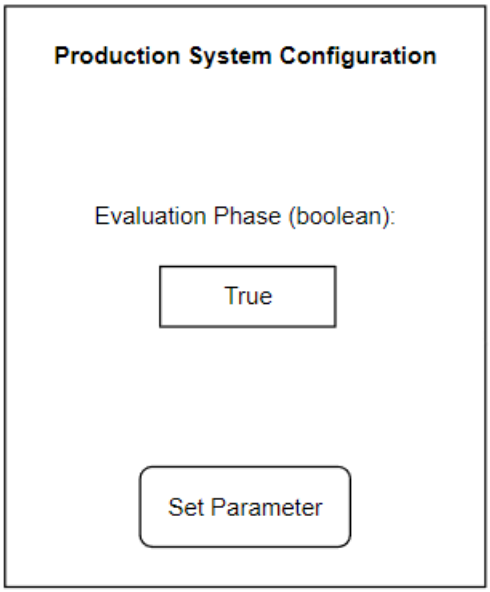
Details

Name	Value
Preconditions	The development system has generated the test report with the winner network

Production System



Human Task: Configure Production System (Cavedoni)



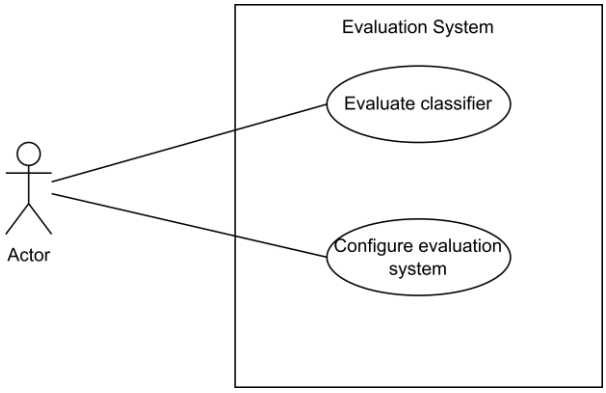
Scenario

1.The use case starts when the User must configure the production system
2. The Actor launches the configuration panel
3. The Actor enter the configuration parameter
5. The Actor press the button “set parameter”
6. The Actor closes the configuration panel

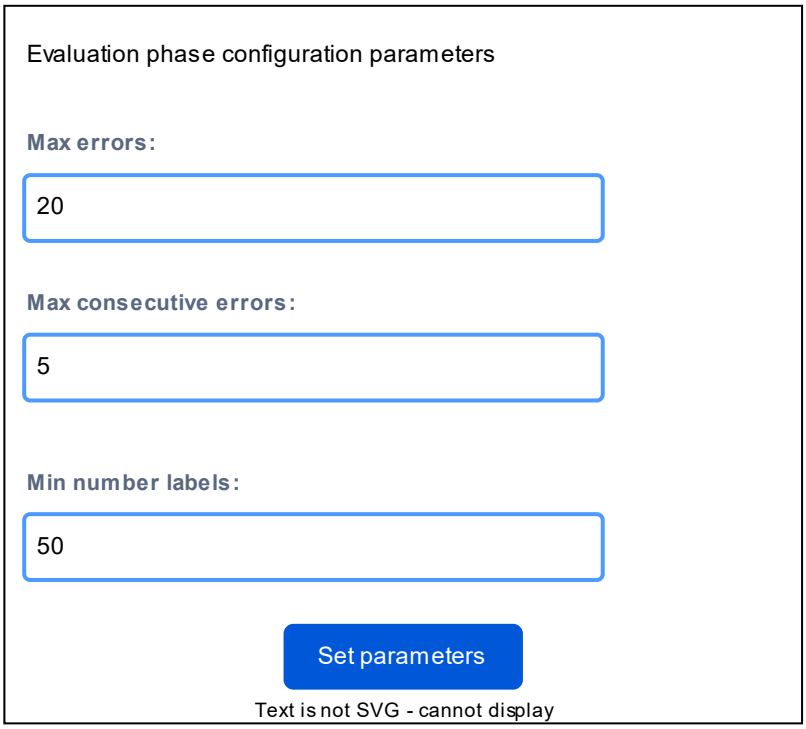
Details

Name	Value
Post-conditions	The Production System has been configured

Evaluation System



Human Task: Configure Evaluation System (Salti)



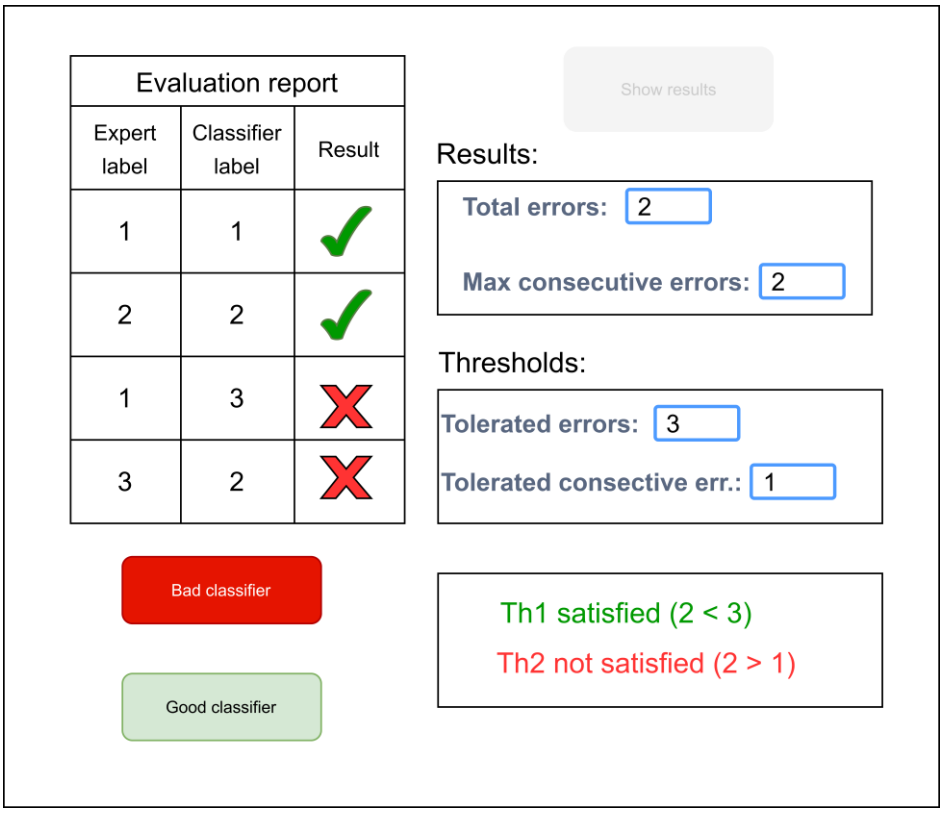
Scenario

1.The use case starts when the Actor opens the configuration panel
2. The Actor sets the maximum number of errors
3. The Actor sets the maximum number of consecutive errors
4. The Actor sets the minimum number of labels to start the evaluation phase
4. The Actor press the button “Set parameters”

Details

Name	Value
Post-conditions	The Evaluation System has been configured

Human task: Evaluate classifier (Salti)



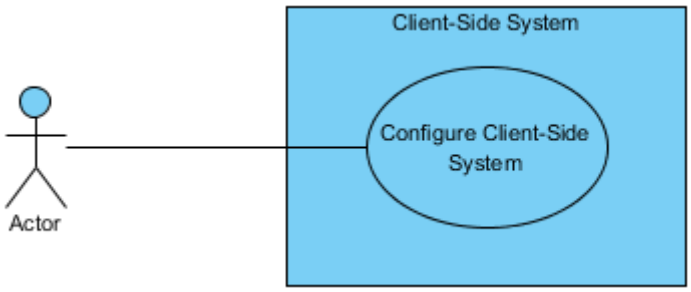
Scenario

1.The use case starts when the Actor clicks the button "show results"
2.The system shows a table having, for each session, a pair of labels (expert and classifier), the **max no. of consecutive errors,** the **total no. of errors**, and the thresholds related to the two quantities
3. if the Actor finds that both thresholds are satisfied
3.1.The Actor clicks the button "Good classifier"
4. else
4.1.The Actor clicks the button "Bad classifier"
4.2.The system sends the restart configuration to the Messagging system.
end if

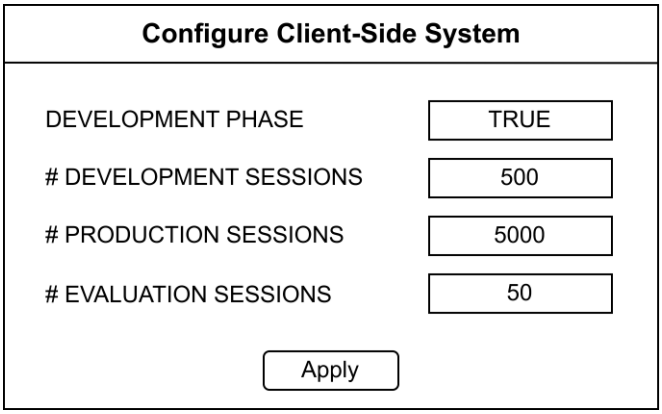
Details

Name	Value
Preconditions	The system must have a sufficient number of labels for each session.

Client-Side Systems (Trasacco)



Human Task: Configure Client-Side System



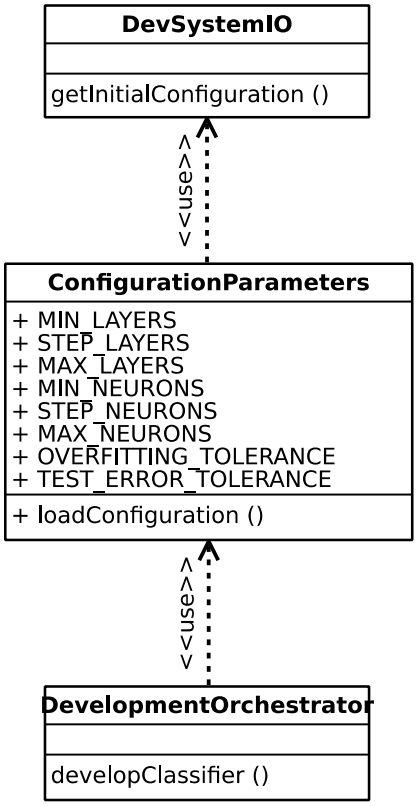
Flow of Events

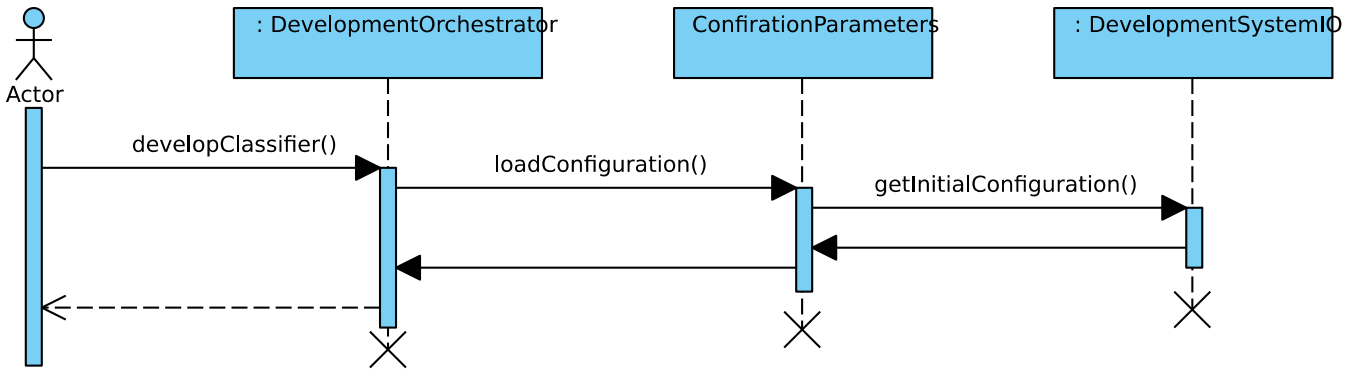
1.The use case starts when the System shows the panel for the configuration.
2. for each parameter to be configured
2.1.The System shows the name of the parameter and an input field for its value
end for each
3. for each parameter to be configured
3.1.The Actor inserts the value of the parameter in the correspondent input field.
end for each
4.The Actor clicks on the Apply button.
5.The System reads the values of the parameters and closes the window.

ANALYSIS – UML Class Diagrams and Sequence Diagrams

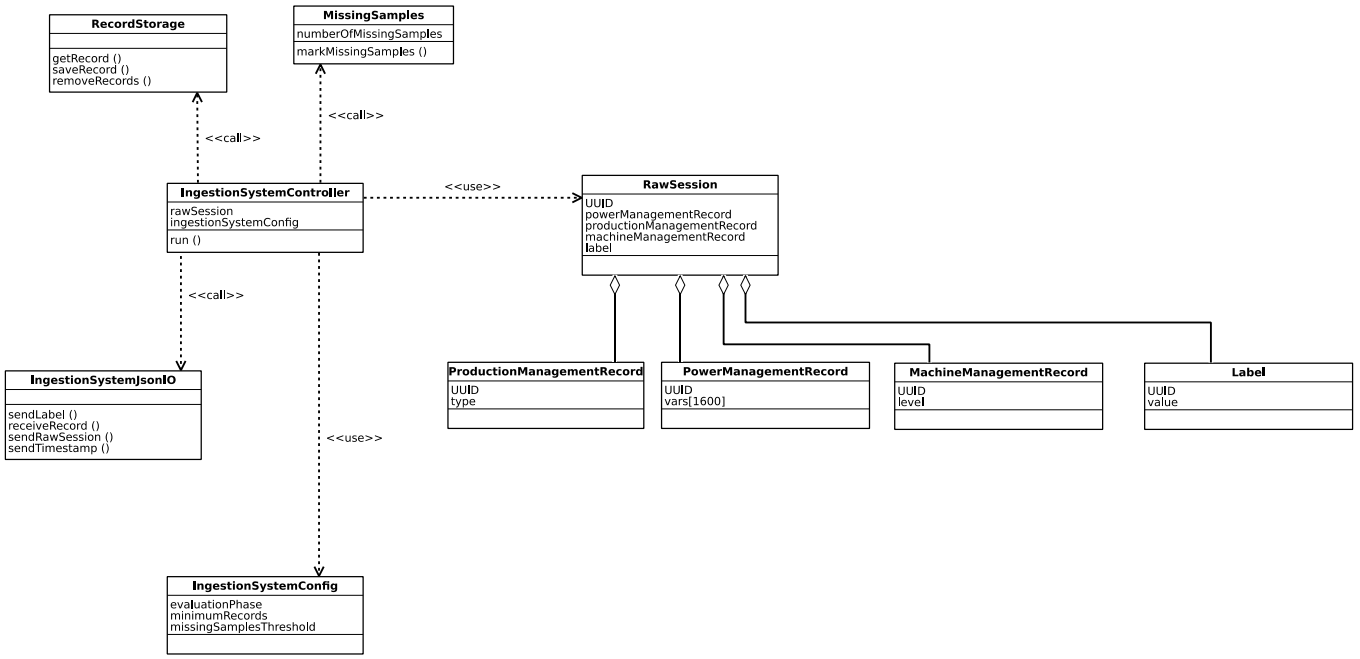
Process CONFIGURE SYSTEMS (Trasacco)

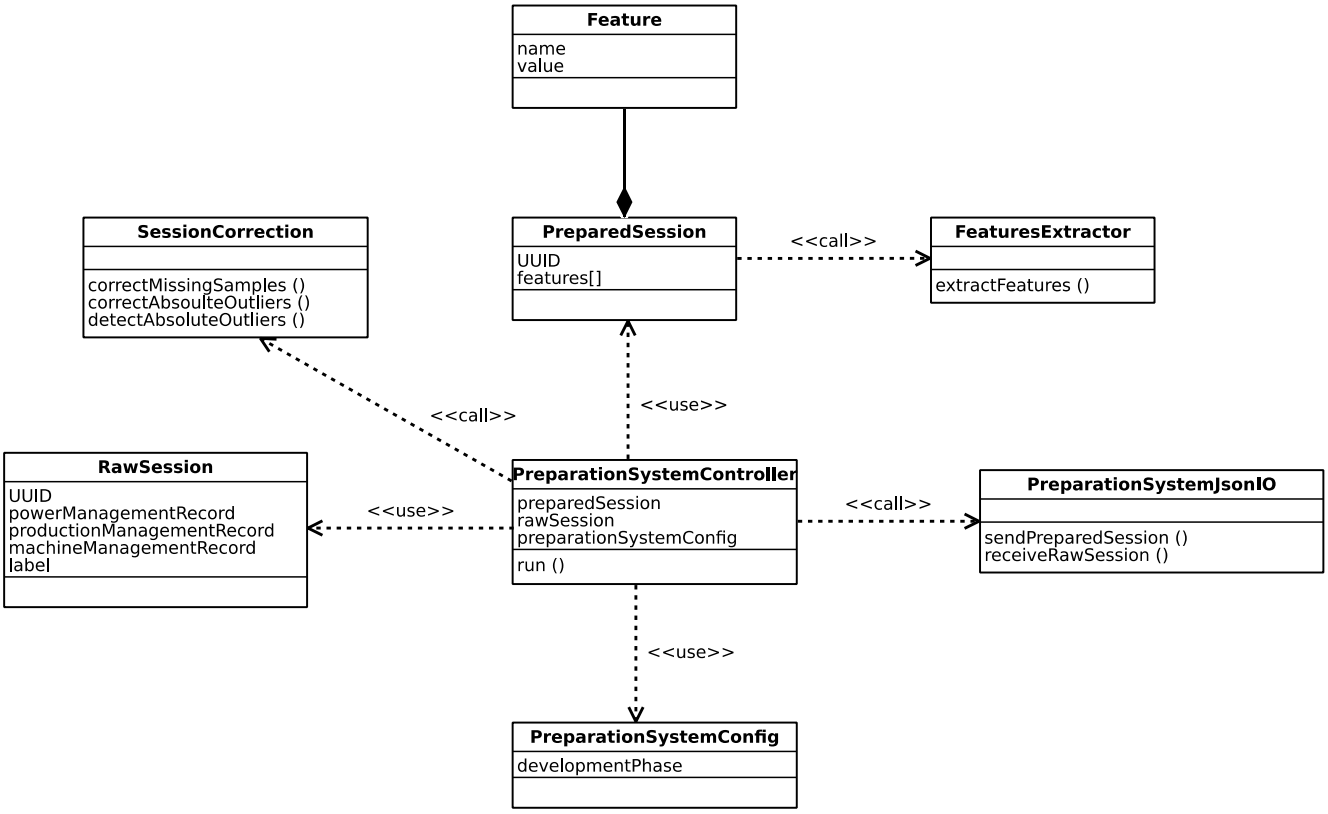
Per quanto riguarda le classi da modellare per questo processo, usiamo come esempio quanto fatto per il sistema di sviluppo del classificatore (e "replicato" anche per gli altri processi). Tutti i metodi e attributi della classe ConfigurationParameters sono statici, Signavio non mostra la sottolineatura.

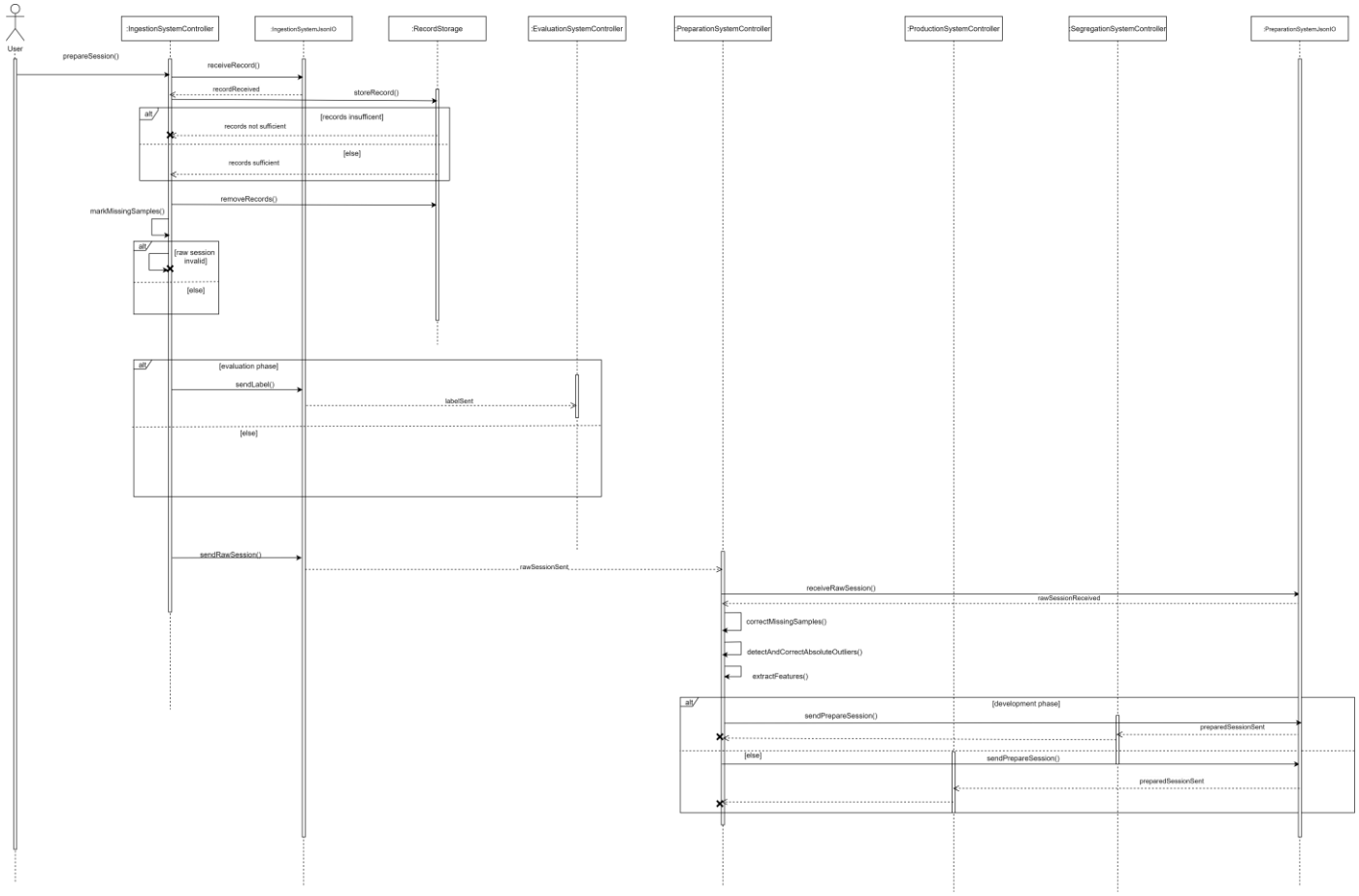




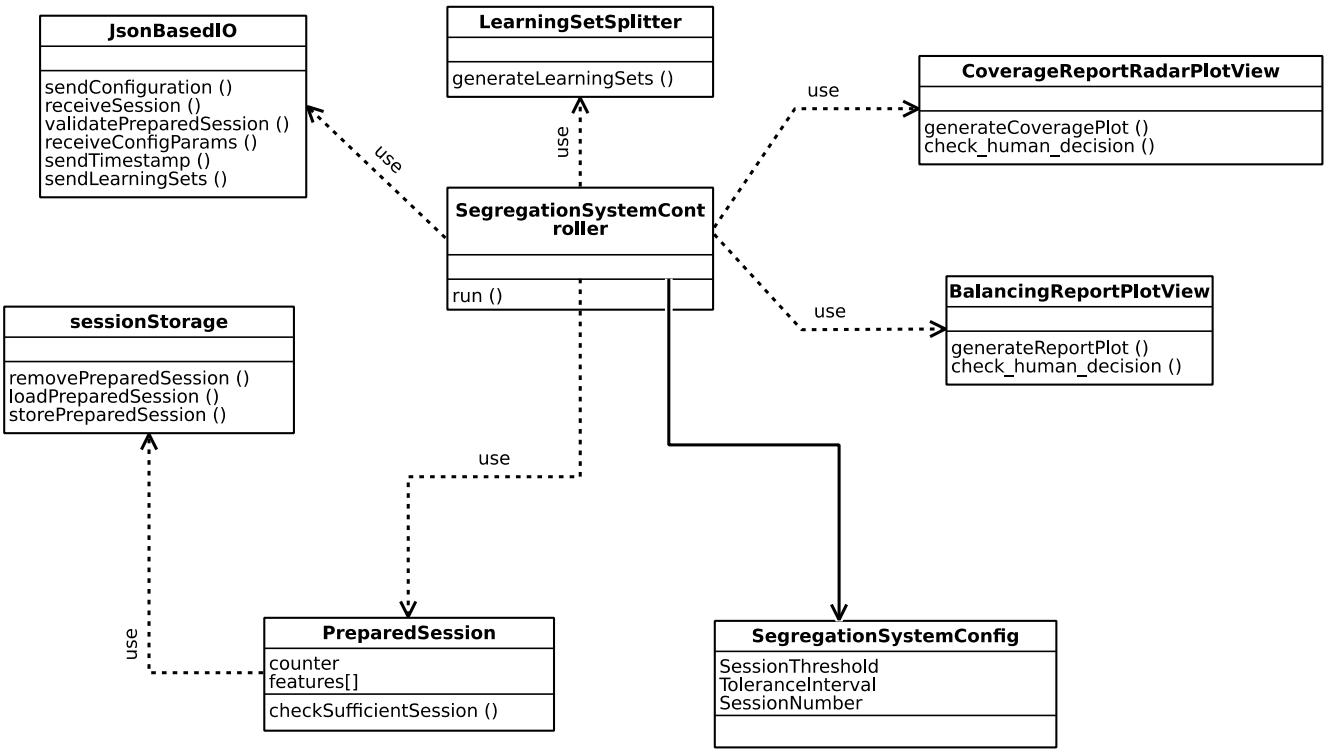
Process PREPARE SESSION (Barbieri)

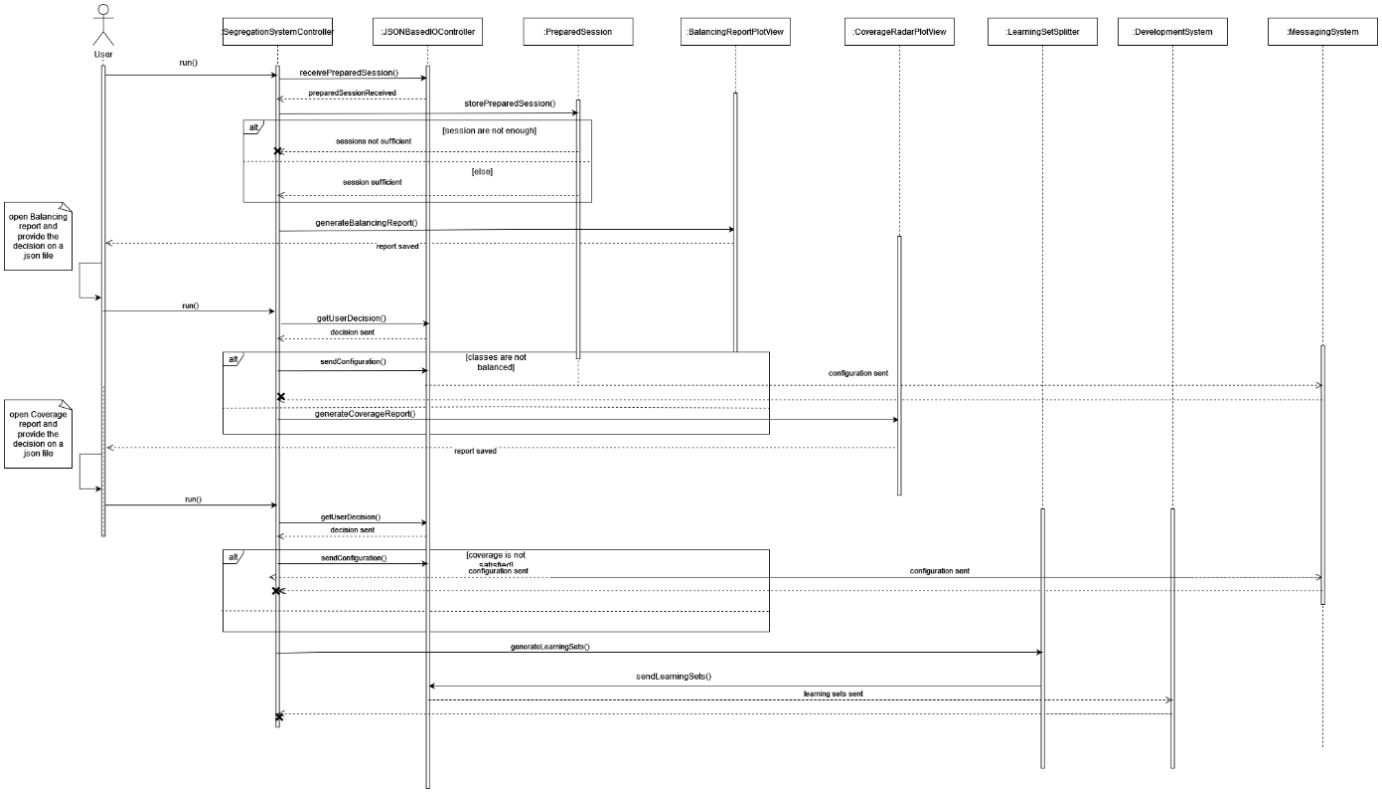




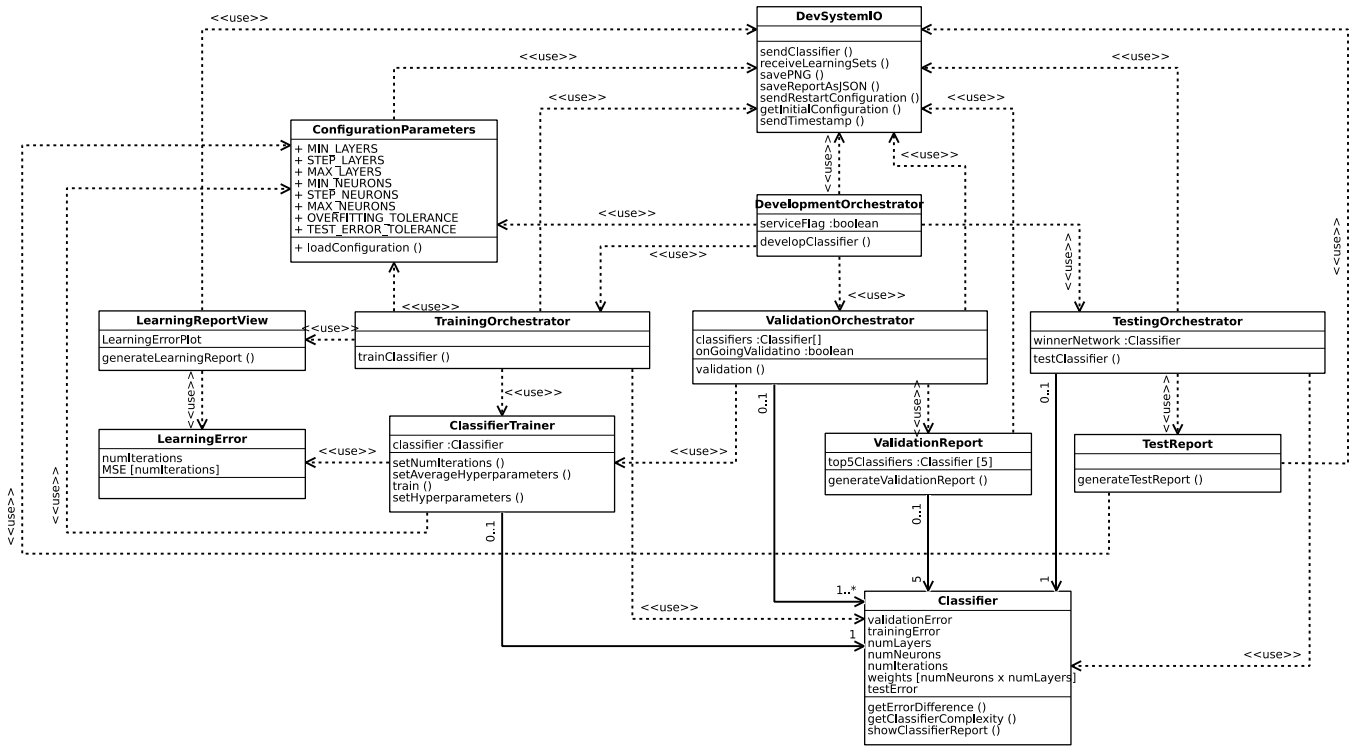

User
:IngestionSystemController
:IngestionSystemJsonIO
:RecordStorage
:EvaluationSystemController
:PreparationSystemController
:ProductionSystemController
:SegregationSystemController
:PreparationSystemJsonIO
prepareSession()
receiveRecord()
recordReceived
storeRecord()
alt
[records insufficent]
records not sufficient
[else]
records sufficient
removeRecords()
markMissingSamples()
alt
[raw session invalid]
[else]
alt
[evaluation phase]
sendLabel()
labelSent
[else]
sendRawSession()
rawSessionSent
receiveRawSession()
rawSessionReceived
correctMissingSamples()
detectAndCorrectAbsoluteOutliers()
extractFeatures()
alt
[development phase]
sendPrepareSession()
preparedSessionSent
[else]
sendPrepareSession()
preparedSessionSent

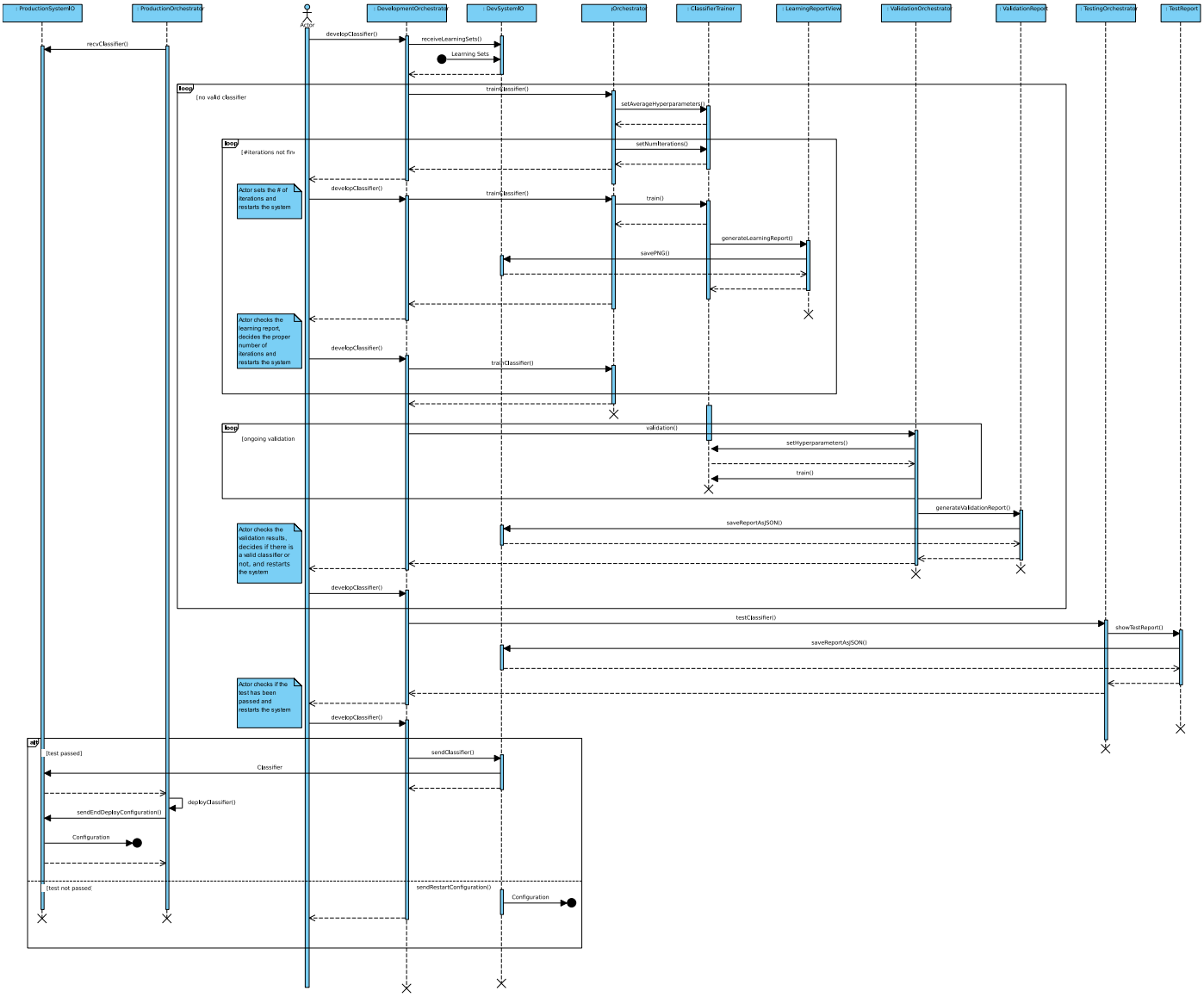
Process GENERATE LEARNING SETS (Di Ricco)



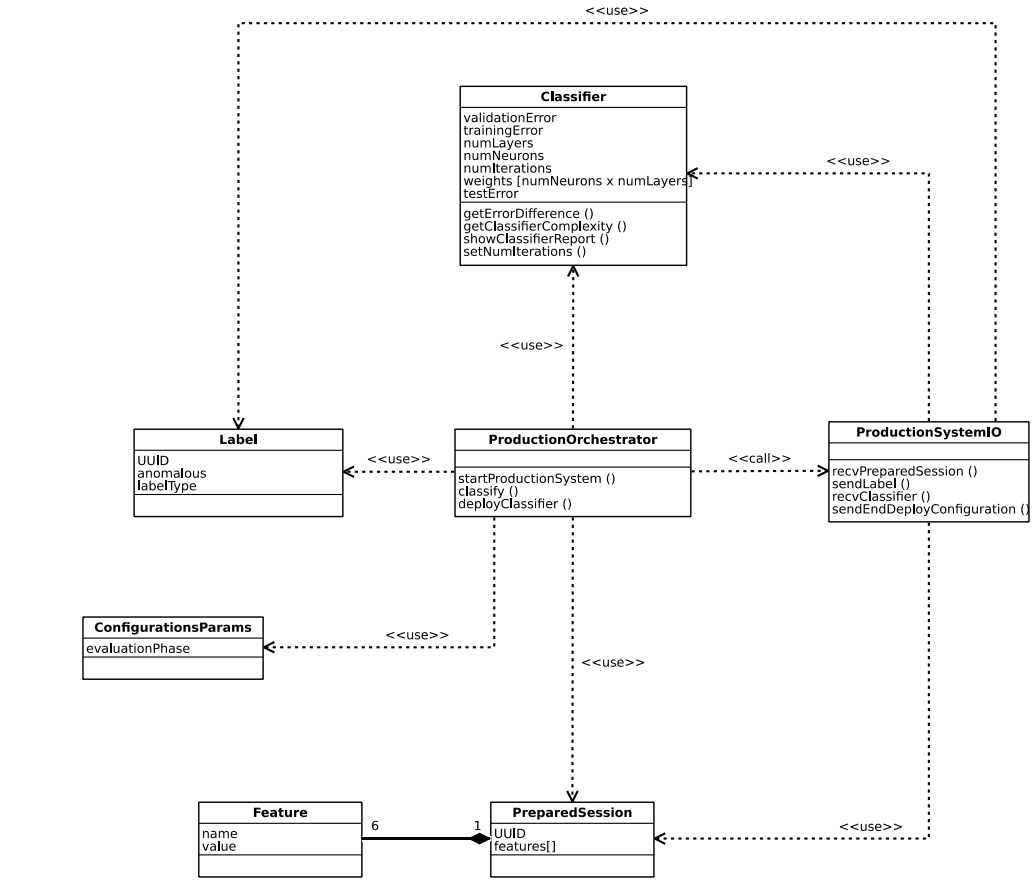


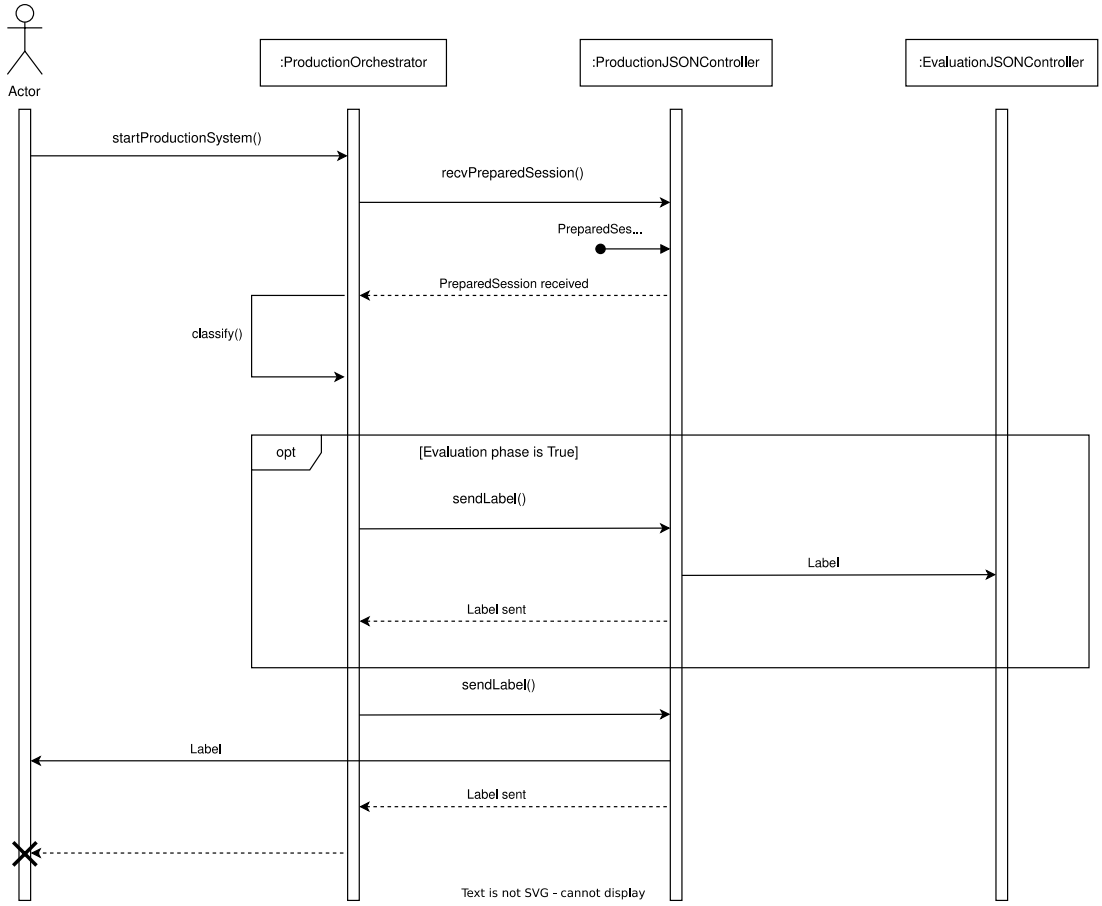
Process DEVELOP CLASSIFIER (Trasacco)



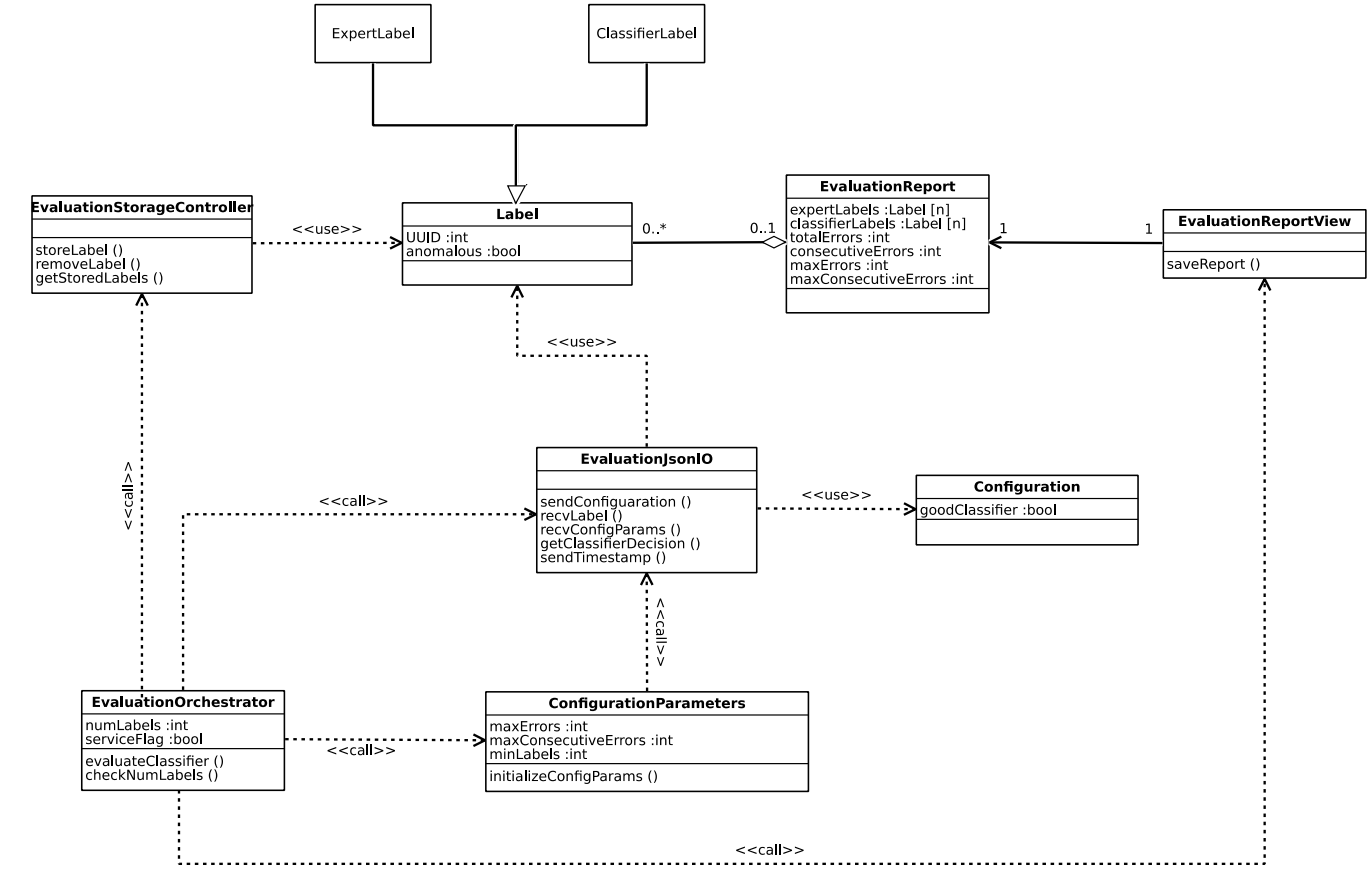


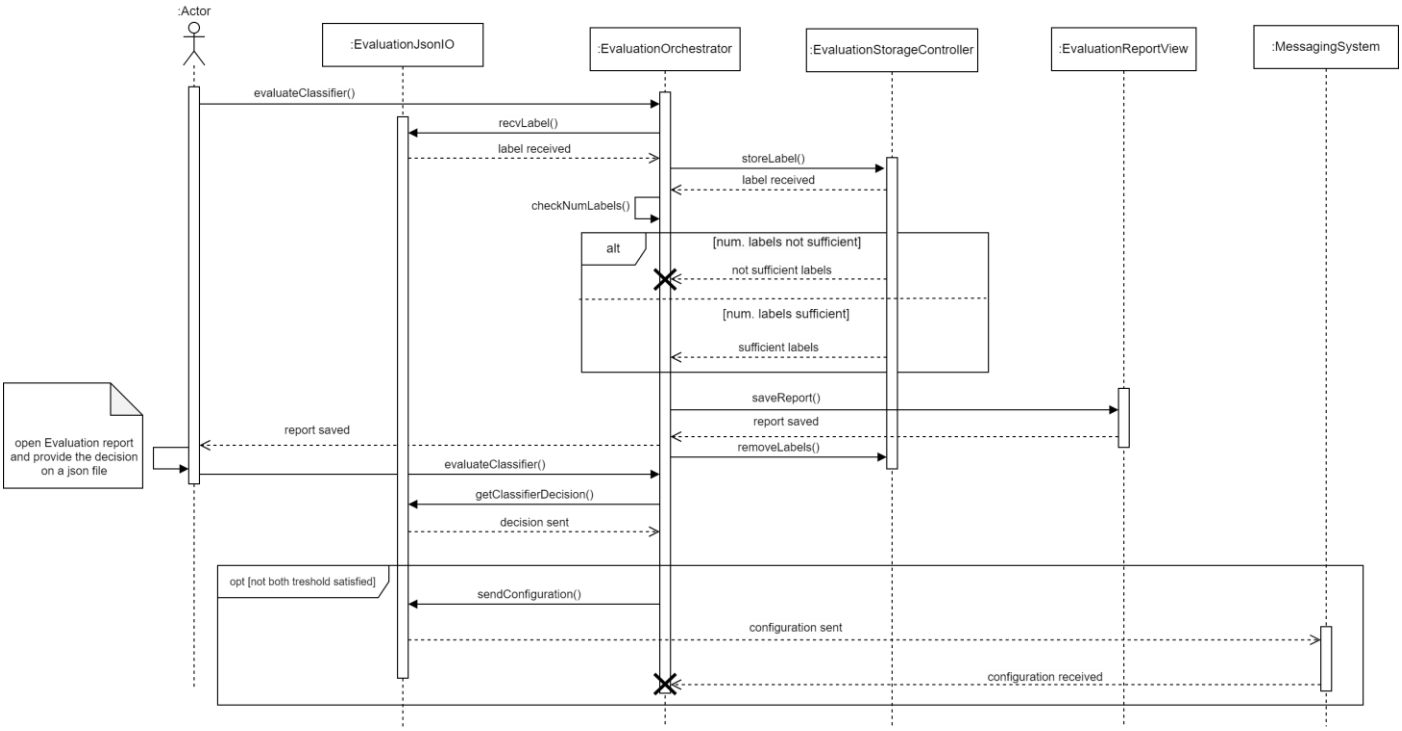
Process CLASSIFY SESSION (Cavedoni)





Process EVALUATION CLASSIFIER PERFORMANCE (Salti)

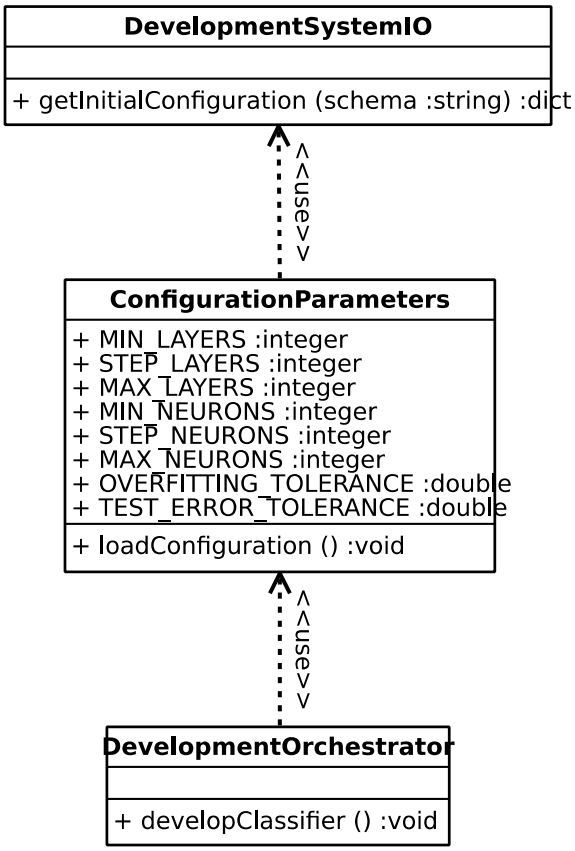


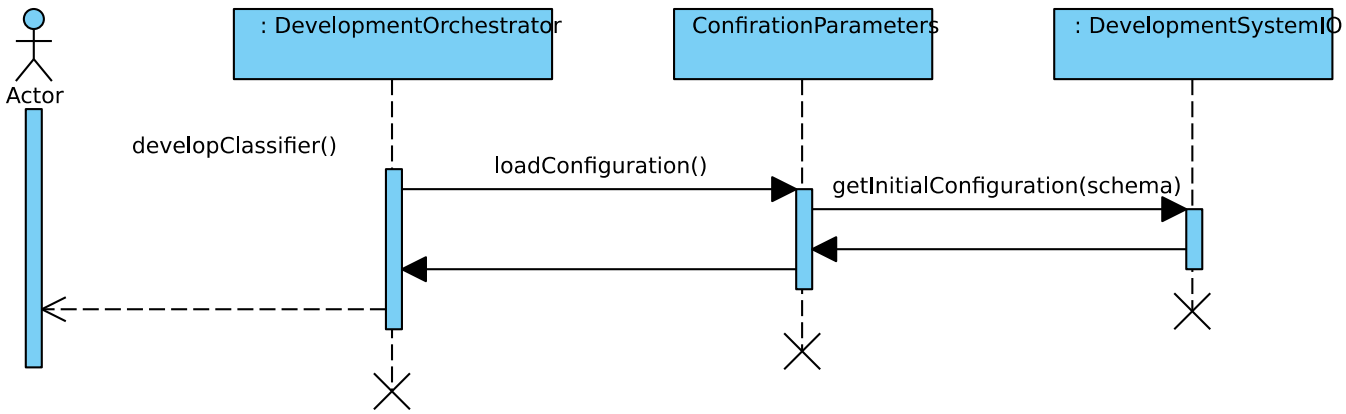


DESIGN – UML Class Diagrams and Sequence Diagrams

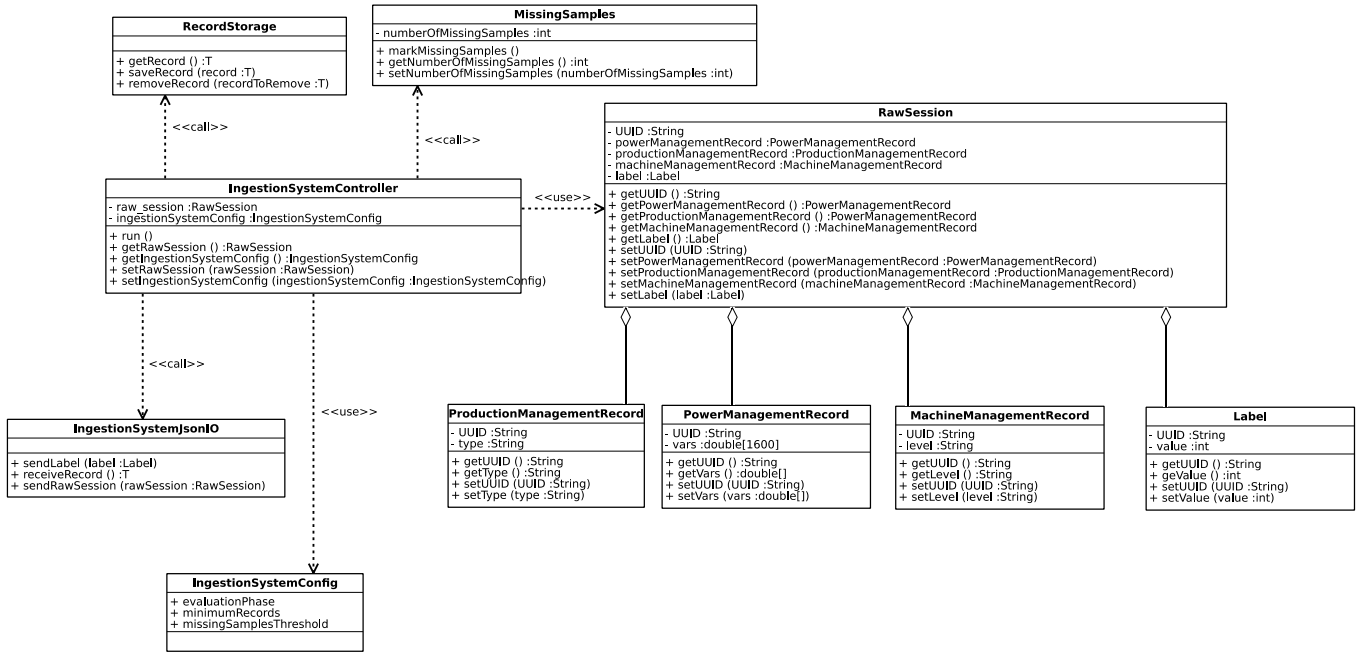
Process CONFIGURE SYSTEMS (Trasacco)

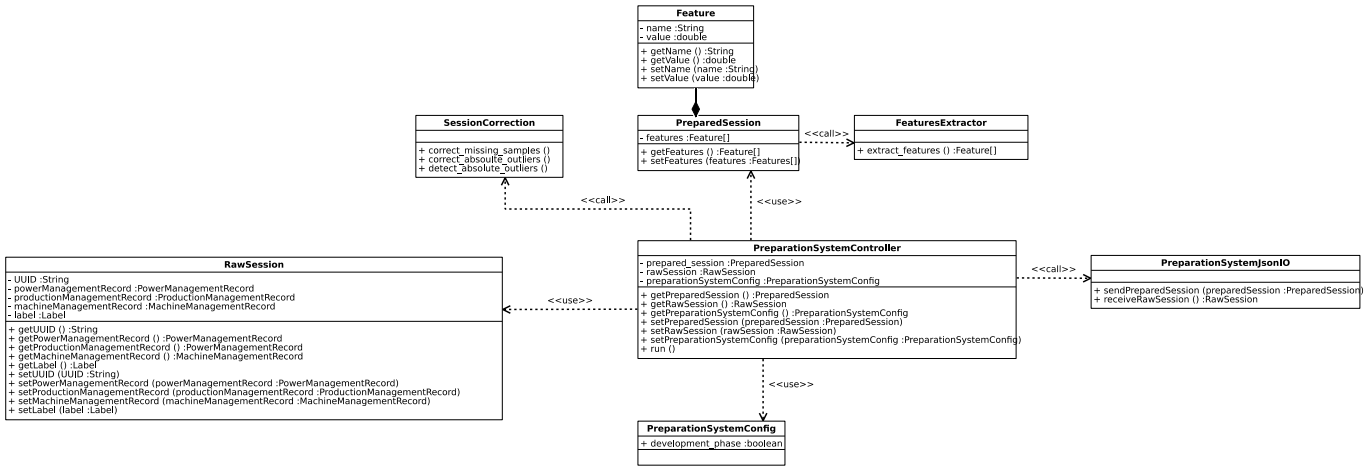
Come mostrato nella fase di Analisi, usiamo come esempio la parte di configurazione del sistema di sviluppo del classificatore (tutti i metodi e attributi della classe ConfigurationParameters sono statici, Signavio non mostra la sottolineatura).

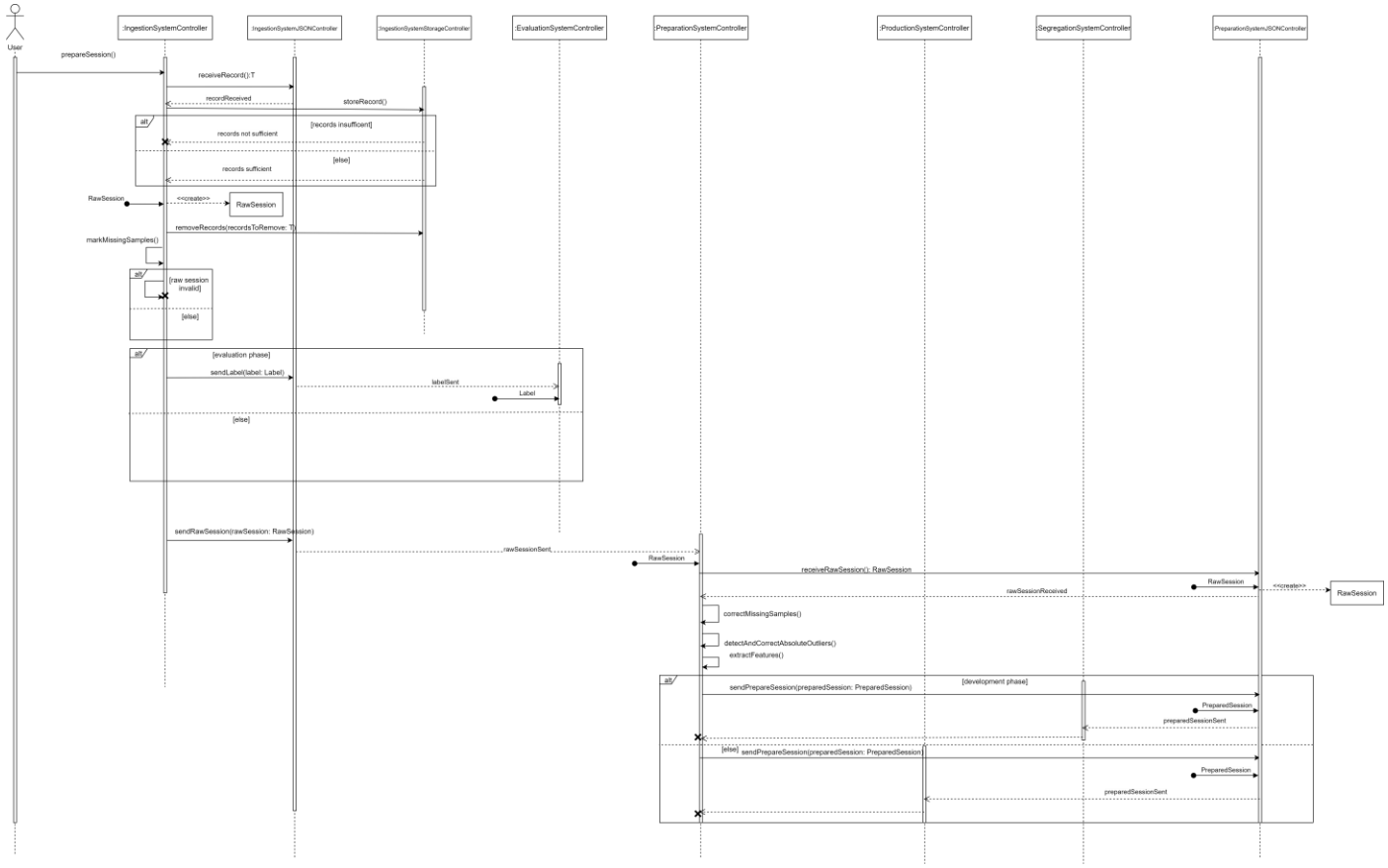




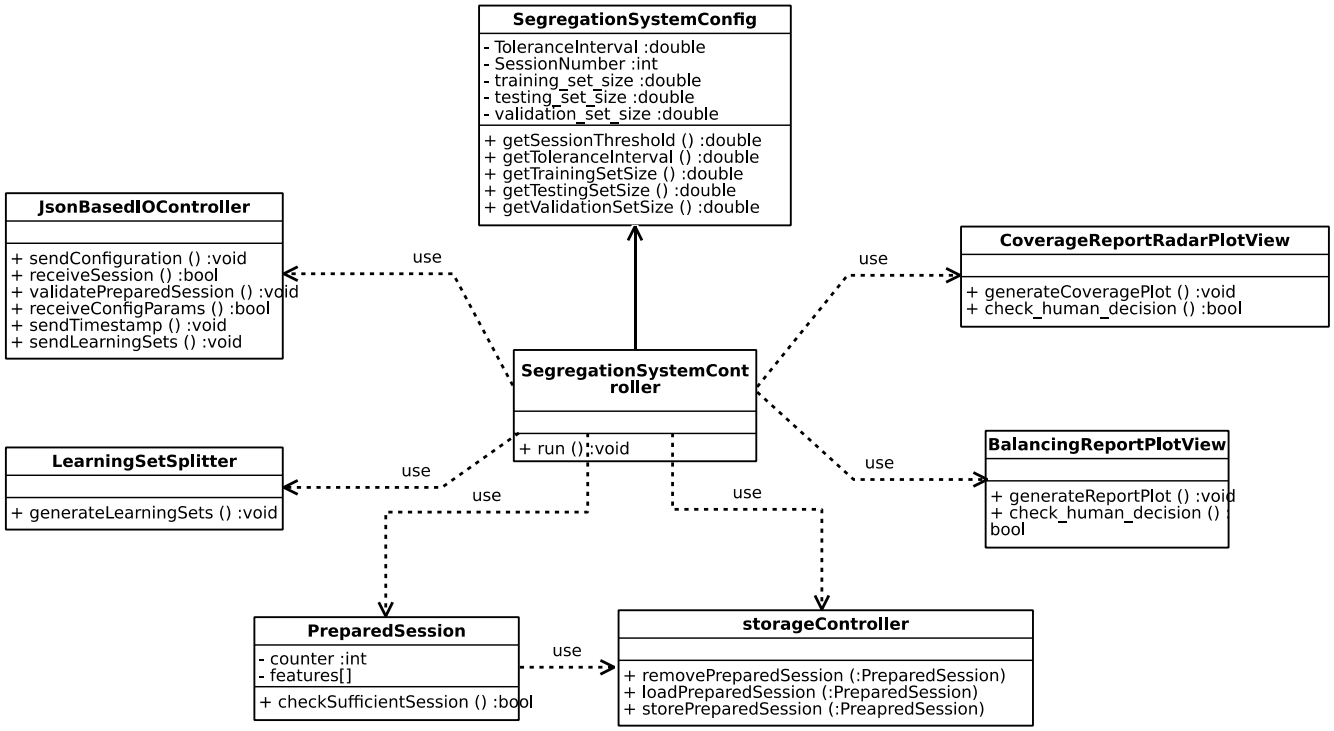
Process PREPARE SESSION (Barbieri)

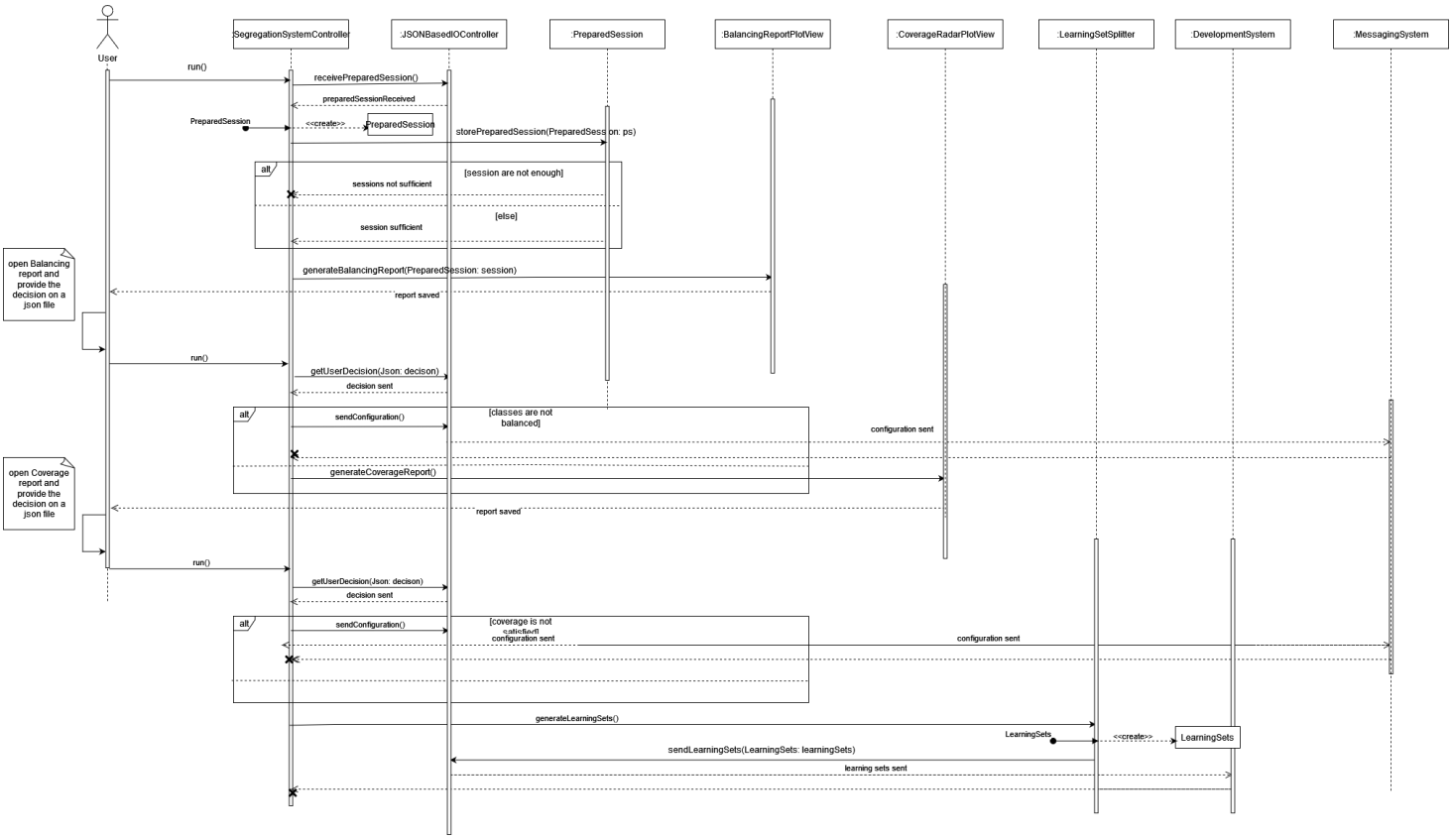




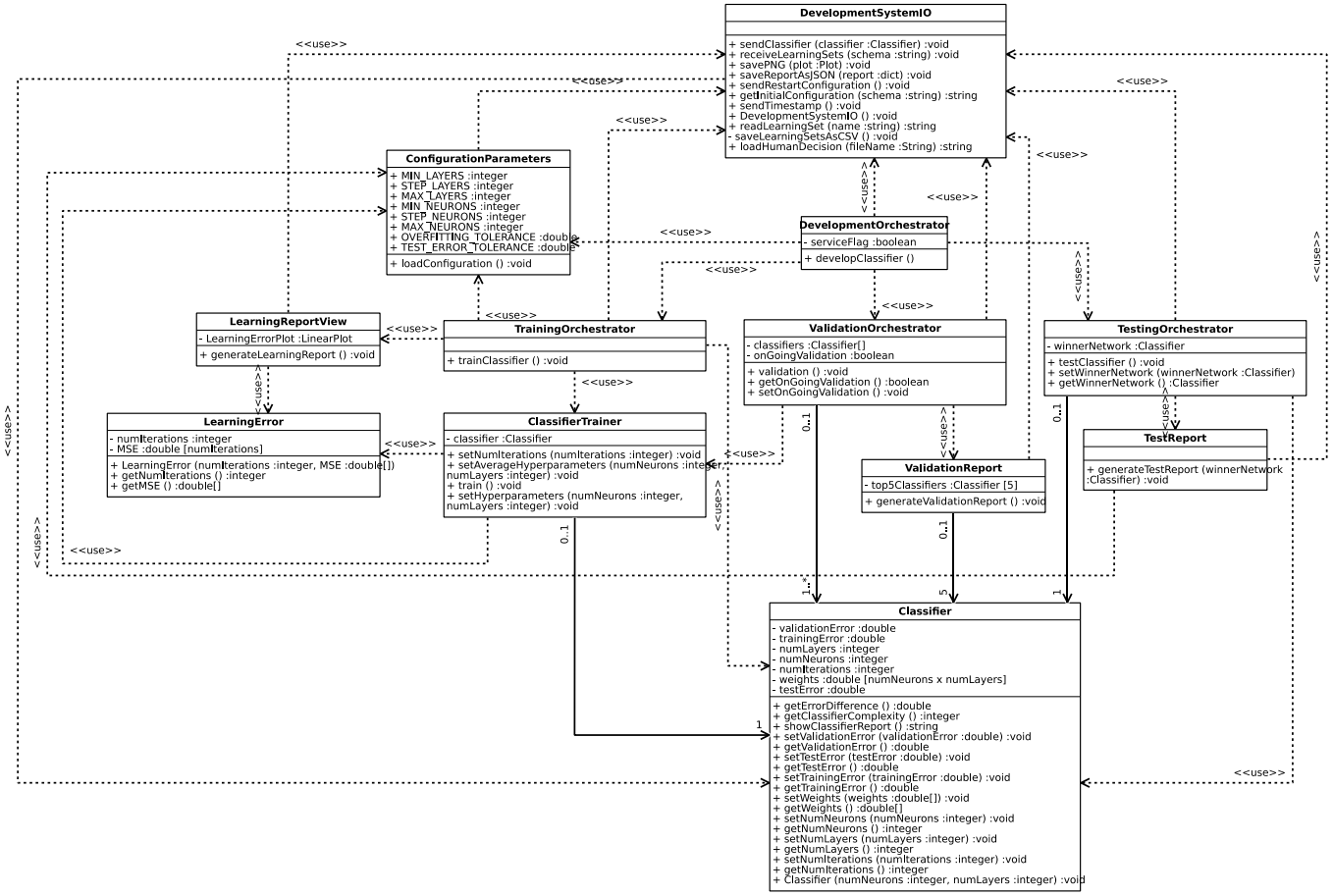

User
:IngestionSystemController
:IngestionSystemJSONController
:IngestionSystemStorageController
:EvaluationSystemController
:PreparationSystemController
:ProductionSystemController
:SegregationSystemController
:PreparationSystemJSONController
prepareSession()
receiveRecord():T
recordReceived
storeRecord()
alt
[records insufficient]
records not sufficient
[else]
records sufficient
RawSession
<<create>>
RawSession
removeRecords(recordsToRemove: T)
markMissingSamples()
alt
[raw session invalid]
[else]
alt
[evaluation phase]
sendLabel(label: Label)
labelSent
Label
[else]
sendRawSession(rawSession: RawSession)
rawSessionSent
RawSession
receiveRawSession(): RawSession
RawSession
<<create>>
RawSession
rawSessionReceived
correctMissingSamples()
detectAndCorrectAbsoluteOutliers()
extractFeatures()
alt
[development phase]
sendPrepareSession(preparedSession: PreparedSession)
PreparedSession
preparedSessionSent
[else]
sendPrepareSession(preparedSession: PreparedSession)
PreparedSession
preparedSessionSent

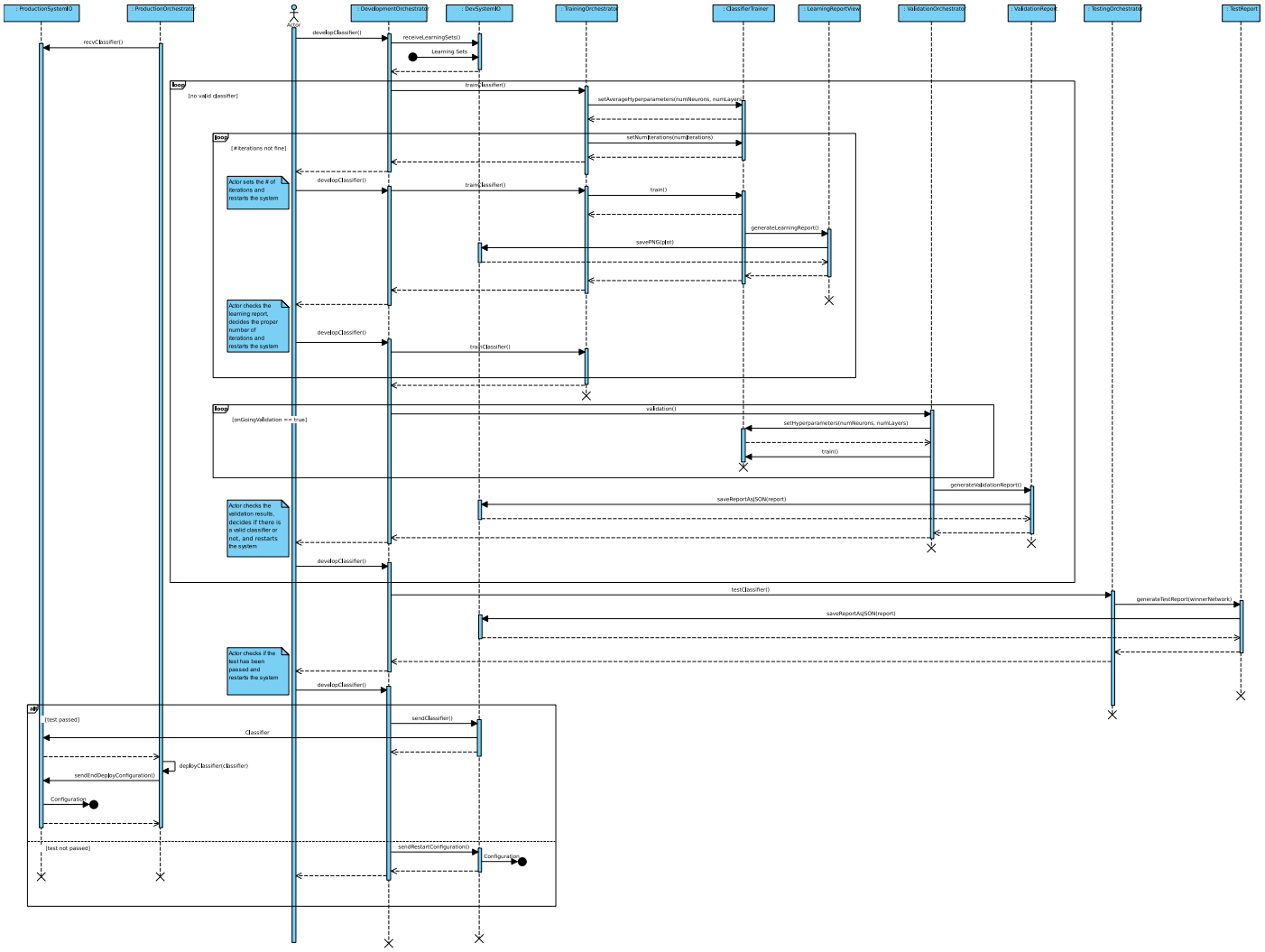
Process GENERATE LEARNING SETS (Di Ricco)



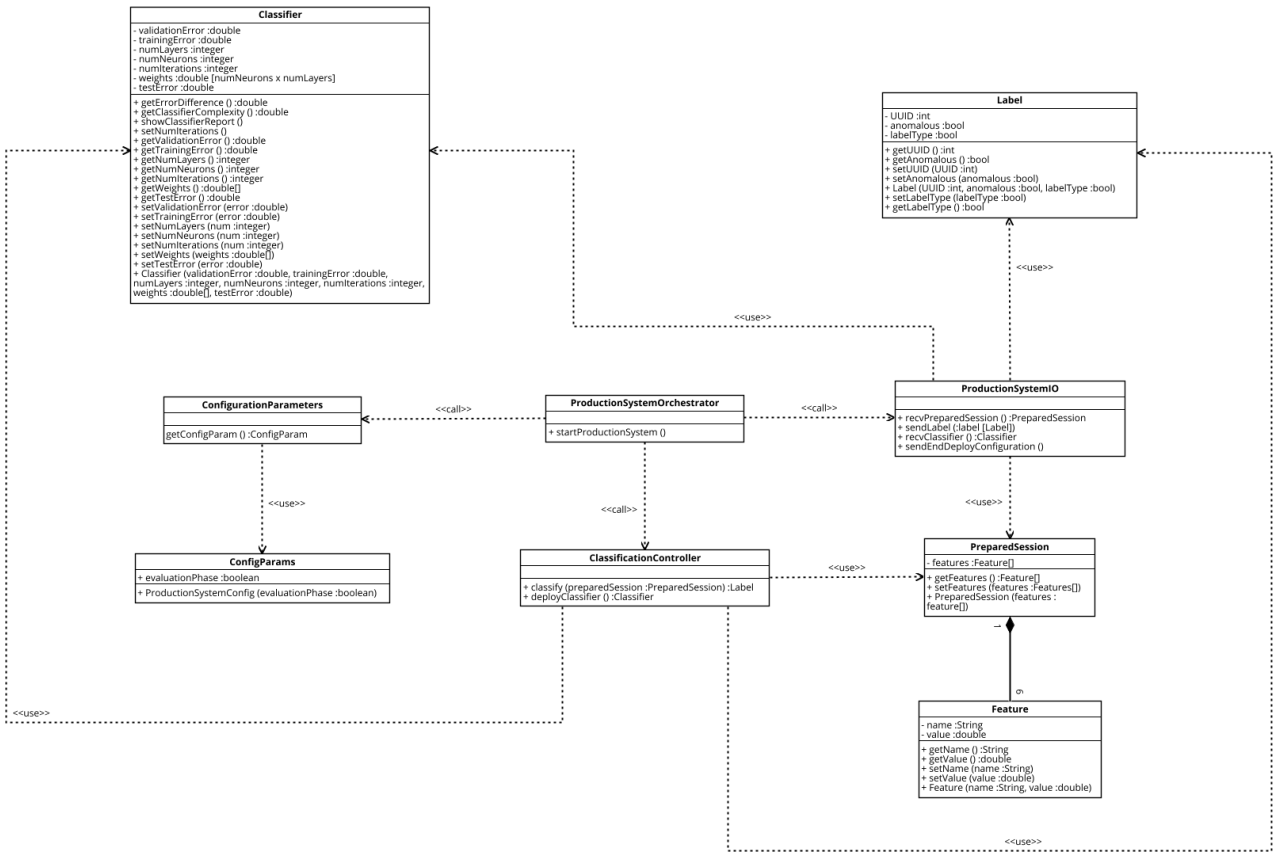


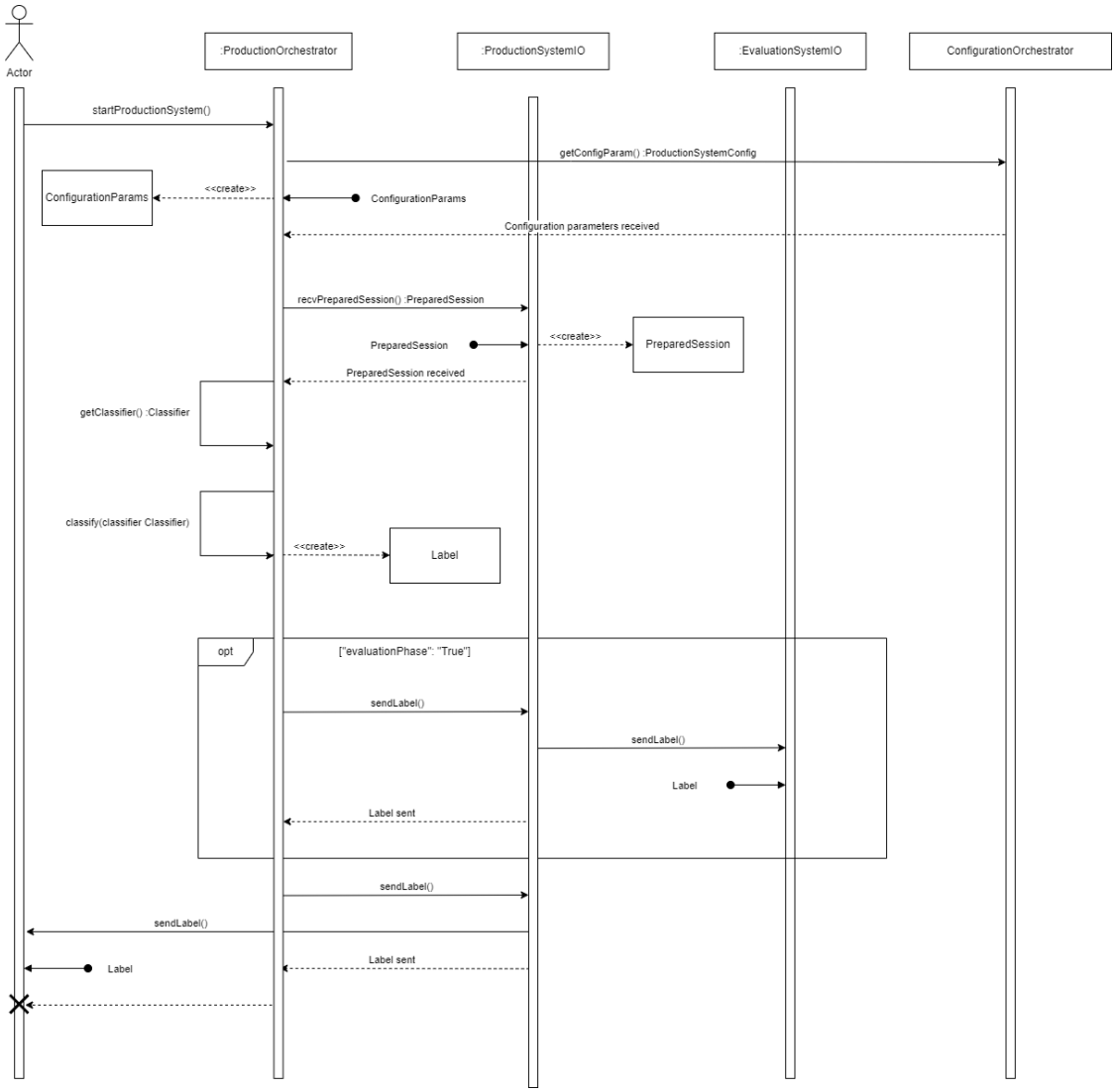
Process DEVELOP CLASSIFIER (Trasacco)



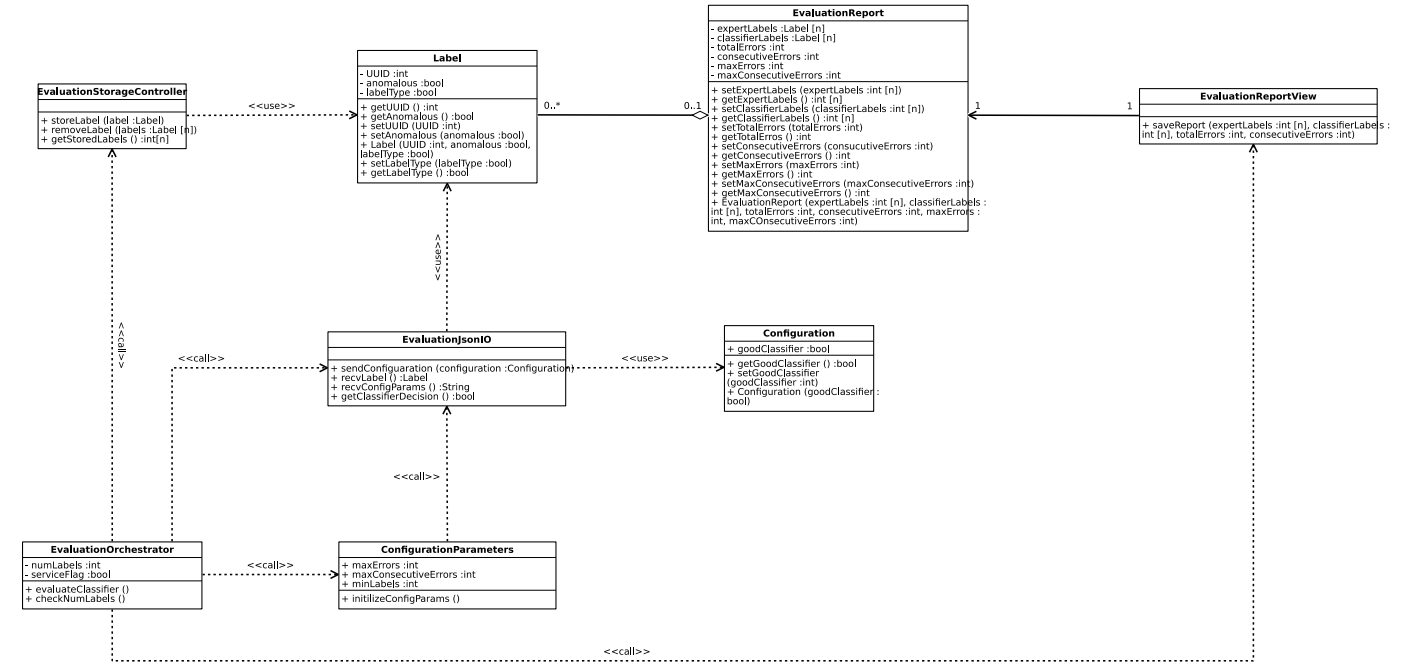


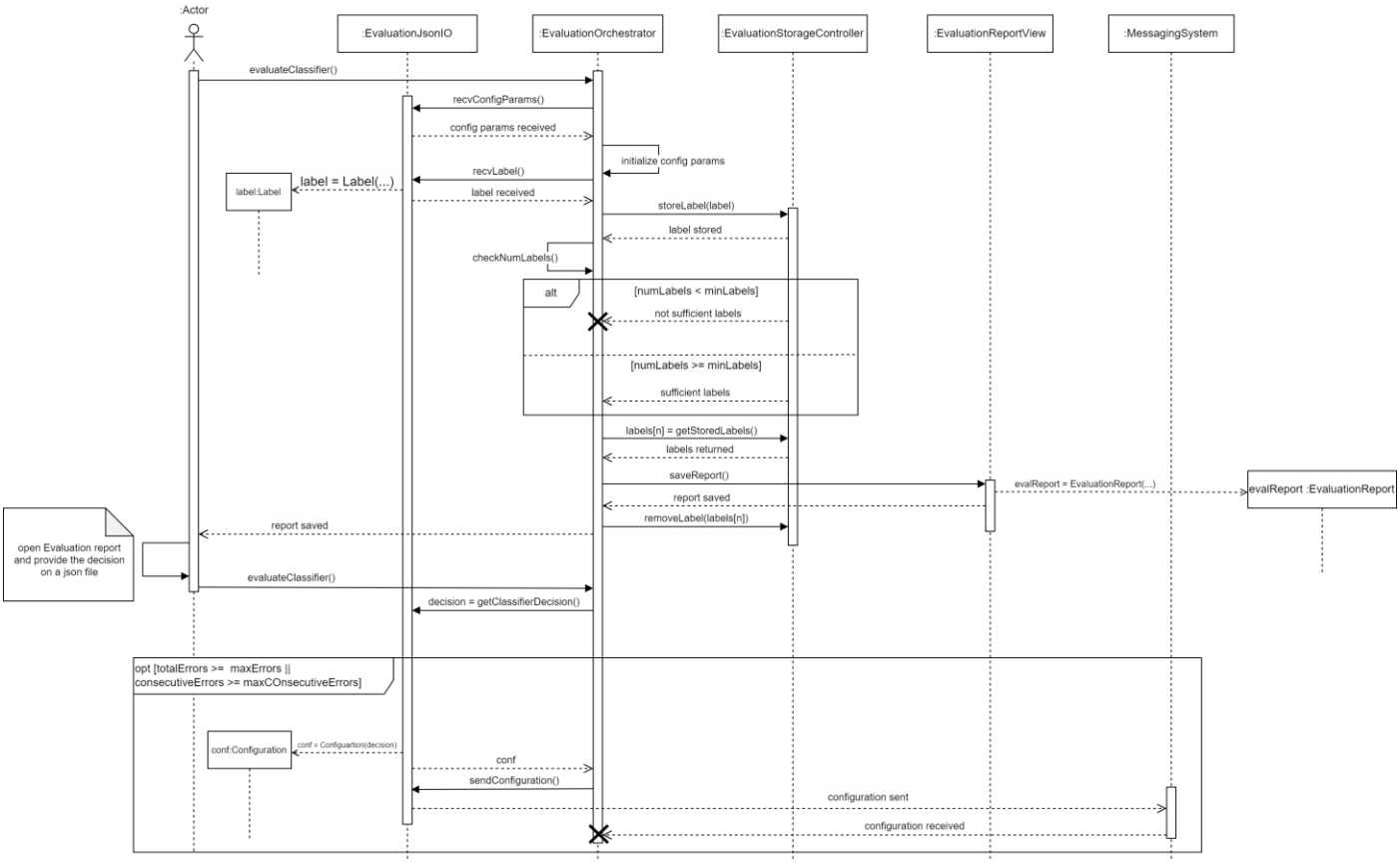
Process CLASSIFY SESSION (Cavedoni)





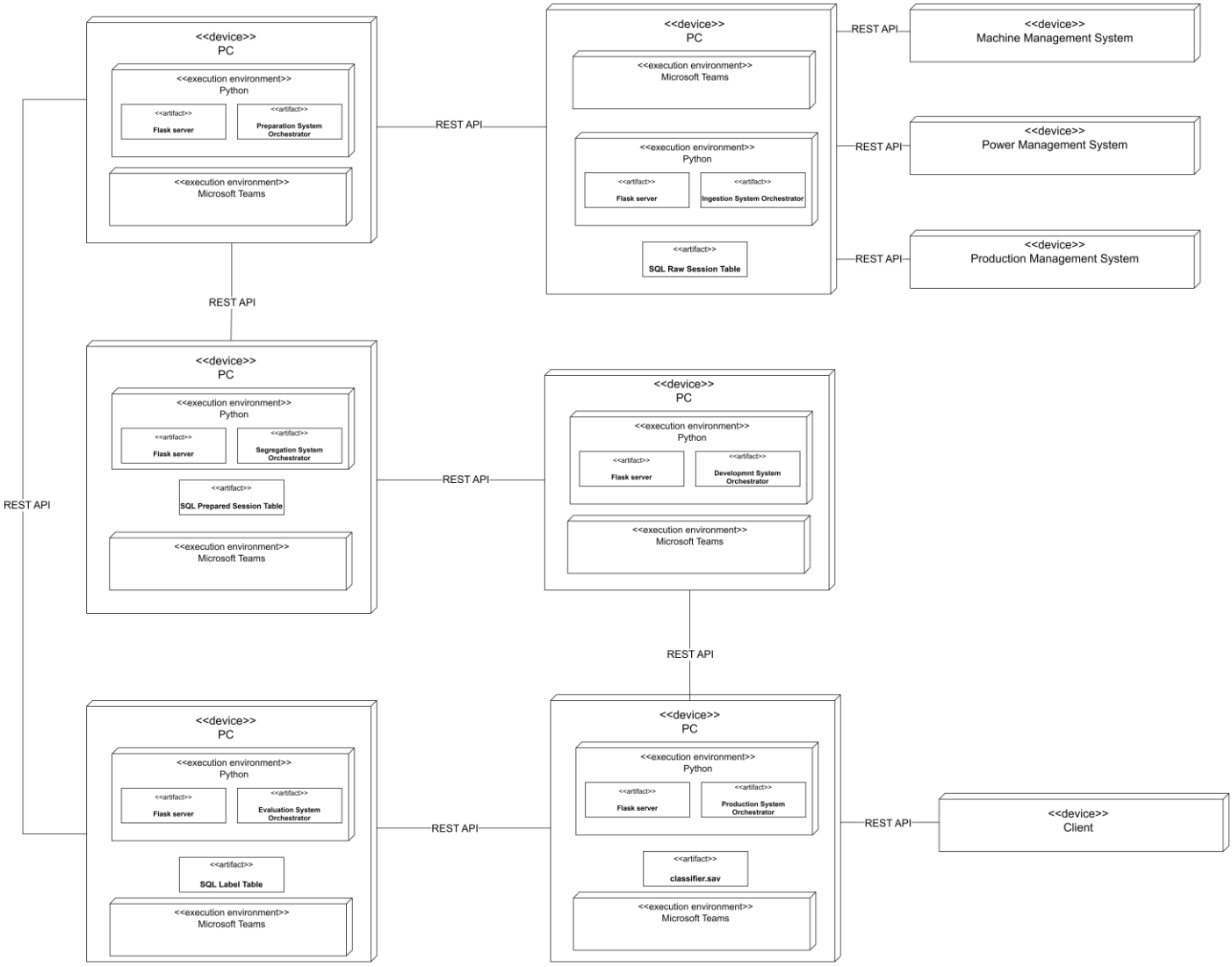
Process EVALUATION CLASSIFIER PERFORMANCE (Salti)





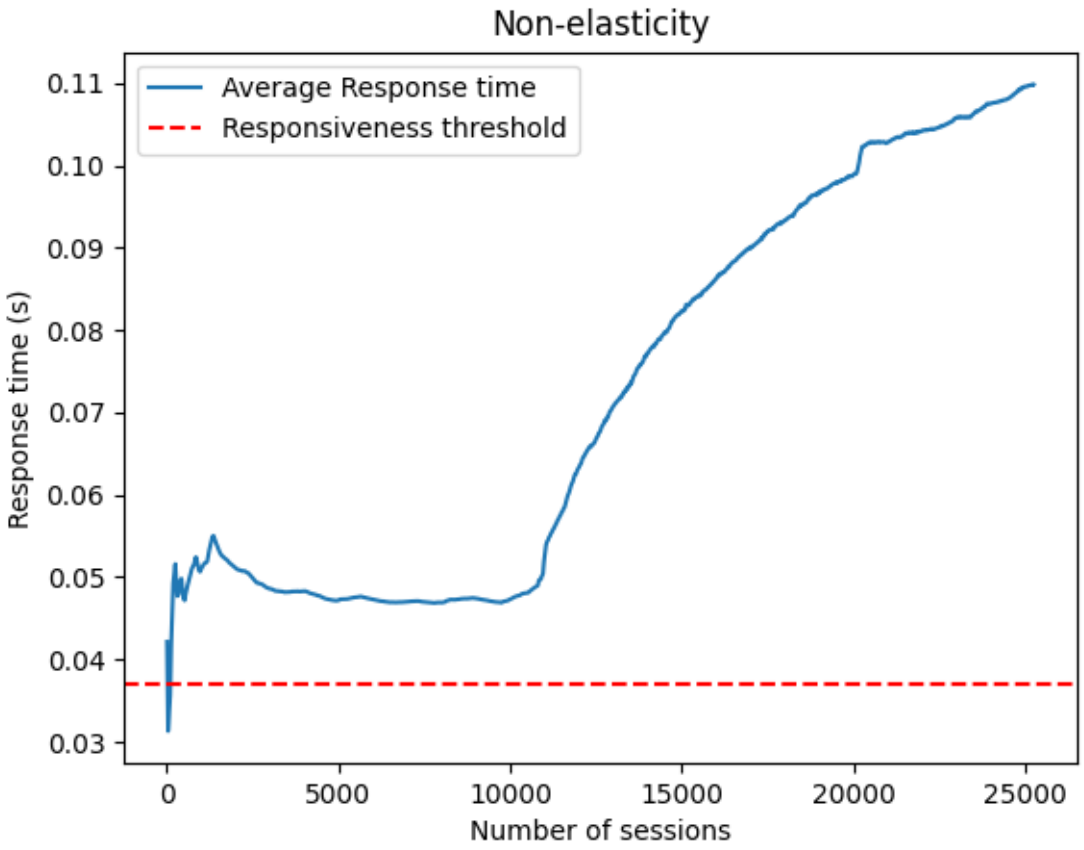
IMPLEMENTATION

UML Deployment Diagram



TESTING

RESPONSIVENESS-ELASTICITY of the production phase (Barbieri)



All'avvio del sistema, si osserva un periodo di tempi di risposta elevati. Tale comportamento è riconducibile a processi di inizializzazione, cache miss, caricamento di risorse o avvio di thread e processi.

Dopo l'assestamento iniziale, i tempi di risposta si abbassano drasticamente per poi ricominciare a crescere poco dopo, ho verificato che la coda dell'ingestion system (che è quella con più carico di tutte) si riempie molto velocemente perciò l'aumento dei tempi di risposta è senz'altro dovuto all'aumento dei tempi di accodamento.

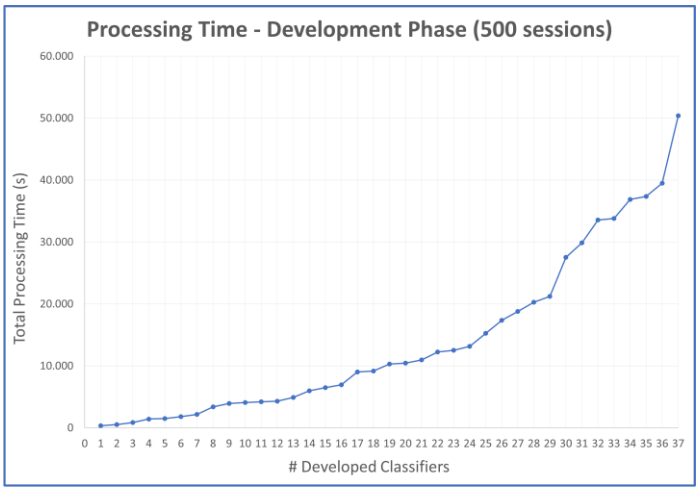
Ad un certo punto però i tempi di risposta hanno un tempo significativamente più alto, la motivazione che è che la coda raggiunge una dimensione tale da non poter stare più nella RAM passando perciò nella swap area, questo fa perciò alzare di molto i tempi di risposta.

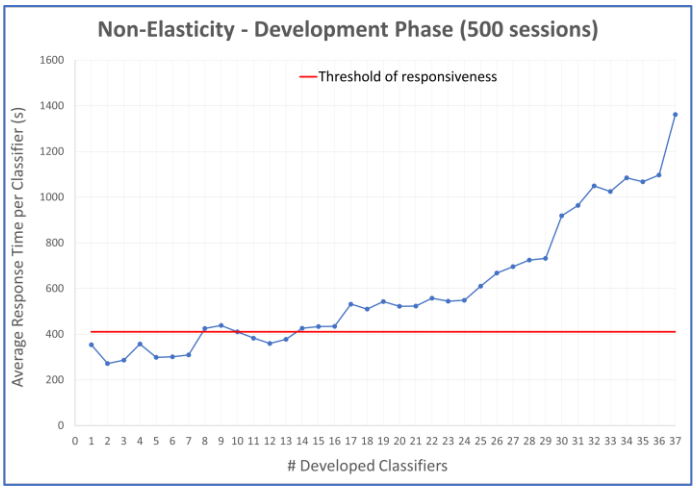
La responsiveness threshold è stata calcolata quando il sistema era basso carico, possiamo vedere che è molto più bassa rispetto al tempo di risposta del sistema per la maggior parte del tempo.

Posso perciò concludere che il sistema non è responsive quando abbiamo alti carichi, alcune possibili modifiche per poter migliorare questo aspetto possono essere l'utilizzo di più thread e ottimizzazione delle operazioni di I/O.

RESPONSIVENESS-ELASTICITY of the development phase (Trasacco)

Each classifier has been developed using **500 sessions**. The value of **non-responsiveness** has been fixed to **410 seconds** (almost 7 minutes) computed as the average response time with 10 classifiers. The **factory has been put under stress** giving as input a high number of records at a high rate. The results are shown by the following graphs:





Increasing the number of developed classifiers over 10, the average response time increases **exponentially**. The response time per classifier is above the **threshold of responsiveness** since the 14th classifier: this means that, even considering the exponential trend, the level of **non-elasticity is quite low** (2, considering a scale from 1 to 5) if we can assume the probability of developing "together" more than 10 classifiers very low. In the case in which the assumption doesn't hold, the level of non-elasticity is **high** (4), due to the exponential behavior.

What happens after the first 10/15 classifiers? **The queues of the various systems increase** since the beginning of the test and, after 15 classifiers, the **memory reaches its limit** (swap area starts to be used).

In particular, **the Ingestion, Preparation and Segregation systems** are the ones with **the highest load** since the factory stops most of the times at the Segregation System (due to unbalanced classes or bad input coverage) leaving free the Development and Production systems. The reason for the last statement is due to the high percentage of occurrence of the paths, in the Segregation System, which lead to restart the entire development.

To increase the elasticity of the factory for the development phase, we must **improve the computational capabilities of the nodes on which the systems run** and **improve the application logic going in parallel** at least for the first three systems of the pipeline.

AUTOMATION (Salti)

Configure Ingestion System	**COST CALCULATION (occ x cognitive x salary)**	**SC**
1. The use case starts when System administrator opens the interface to configure the ingestion system	1 x 1 x 1.6	1.6
2. for each parameter the System administrator digits the value in the specific input field	3 x 2 x 1.6	9.6
3. The System administrator press the OK button	1 x 1 x 1.6	1.6
	NON-AUTOMATION COST	12.8

Configure Preparation System	**COST CALCULATION (occ x cognitive x salary)**	**SC**
1. The use case starts when the System administrator opens the interface to configure Preparation System	1 x 1 x 1.6	1.6
2. for each parameter the System administrator digits the value in the specific input field	1 x 2 x 1.6	3.2
3. The System administrator press the OK button	1 x 1 x 1.6	1.6
	NON-AUTOMATION COST	6.4

Configure Segregation System	**COST CALCULATION (occ x cognitive x salary)**	**SC**
1.The use case starts when the system opens the configuration panel		
2. The System administrator sets the tolerance interval	1 x 2 x 1.6	3.2
4. The System administrator sets the validation set size	1 x 2 x 1.6	3.2
5. The System administrator sets the testing set size	1 x 2 x 1.6	3.2
6. The System administrator press the button “Set parameters”	1 x 1 x 1.6	1.6
	NON-AUTOMATION COST	11.2

Check data balancing	**COST CALCULATION (occ x cognitive x salary)**	**SC**
1. The Data analyst launches the img viewer	1 x 1 x 1	1
2. The Data analyst opens the plot that display the data balancing	1 x 1 x 1	1
3. The Data analyst evaluate if the data are well balanced between the classes	1 x 2 x 1	2
4. The Data analyst close the plot	1 x 1 x 1	1
5. if the data are balanced	0.2 x 2 x 1	0.4
5.1 The Data analyst enter “balanced” in the file “balancingEvaluation.json”	0.2 x 1 x 1	0.2
6. else	0.8 x 2 x 1	1.6
6.1 The Data analyst enter “not balanced” in the file “balancingEvaluation.json”	0.8 x 1 x 1	0.8
6.2 A new configuration is requested	0.8 x 1 x 1	0.8
7.end if	NON-AUTOMATION COST	8.8

Check input coverage	**COST CALCULATION (occ x cognitive x salary)**	**SC**
1.The Data analyst launch the application with the balancing report	1 x 1 x 1	1
2.The Data analyst analyze the report and check whether the input space is uniformly covered	1 x 4 x 1	4
3.FOR EACH features		
3.1.IF the distribution is not uniform	(6 x 2/3) x 4 x 1	16
3.1.1.The Data analyst click NO button	(6 x 2/3) x 1 x 1	4
3.1.2.The Data analyst request a new Data Configuration	(6 x 2/3) x 1 x 1	4
3.1.3.The Data analyst close the balancing report	(6 x 2/3) x 1 x 1	4
4. end for each		
5.The Data analyst click OK button	1 x 1 x 1	1
	NON-AUTOMATION COST	34

Configure Development System	**COST CALCULATION (occ x cognitive x salary)**	**SC**
1.The use case starts when the System shows the panel for the configuration.		
2. for each parameter to be configured		
2.1.The System shows the name of the parameter and an input field for its value		
end for each		
3. for each parameter to be configured		
3.1. The ML engineer inserts the value of the parameter in the correspondent input field.	8 x 2 x 1.4	22.4
end for each		
4.The ML engineer clicks on the Apply button.	1 x 1 x 1.4	1.4
5.The System reads the values of the parameters and closes the window.	NON-AUTOMATION COST	23.8

Set # of iterations	**COST CALCULATION (occ x cognitive x salary)**	**SC**
1.The use case starts when the ML engineer opens the interface for setting the number of iterations	1 x 1 x 1.4	1.4
2.The ML engineer digits the number of iterations in the "Number of iterations" field	1 x 2 x 1.4	2.8
3.The ML engineer presses the "OK" button	1 x 1 x 1.4	1.4
	NON-AUTOMATION COST	5.6

Check learning plot	**COST CALCULATION** **(occ x cognitive x salary)**	**SC**
1.The use case starts when the ML engineer clicks on "Check the learning plot".	1 x 1 x 1.4	1.4
2.The System shows a window with the learning error plot (MSE) against the number of iterations.		
3. if the ML engineer sees that the loss is flat for at least the half of the iterations then	0.4 x 3 x 1.4	1.68
3.1.The ML engineer reduces by one third the number of iterations via the proper button.	0.4 x 1 x 1.4	0.56
3.2.The System updates the new value for the number of iterations.		
3.3.The ML engineer closes the window using the "X" button.	0.4 x 1 x 1.4	0.56
4. else if the ML engineer sees that the loss is not flat at the end of the iterations then	0.4 x 3 x 1.4	1.68
4.1.The ML engineer enlarges by one third the number of iterations via the proper button.	0.4 x 1 x 1.4	0.56
4.2.The System updates the new value for the number of iterations.		
4.3.The ML engineer closes the window using the "X" button.	0.4 x 1 x 1.4	0.56
5. else	0.2 x 3 x 1.4	0.84
5.1.The ML engineer closes the window using the "X" button.	0.2 x 1 x 1.4	0.28
end if	NON-AUTOMATION COST	8.12

Check validation result	**COST CALCULATION** **(occ x cognitive x salary)**	**SC**
1.The use case starts when the System shows the Validation Report with the top 5 classifiers.		
2. for each classifier in the report		
2.1. The System displays validation error, training error, no. of layers, no. of neurons, network complexity.		
2.2. The cliSystem computes the difference between validation and training error.		
end for each		
3.The System displays the overfitting tolerance		
4. for each classifier in the report		
4.1.The  ML engineer checks if the difference is under the overfitting tolerance	5 x 2 x 1.4	14
end for each		
5. if The  ML engineer notices that there are no classifiers which satisfy the tolerance	0.05 x 2 x 1.4	0.14
5.1.The  ML engineer clicks on "No valid classifier"	0.05 x 1 x 1.4	0.07
6. else	0.95 x 2 x 1.4	2.66
6.1.The  ML engineer chooses the best network among the selected ones based on the value of the difference, the no. of layers, the no. of neurons.	0.95 x 3 x 1.4	3.99
6.2.The  ML engineer clicks on the button correspondent to the index of the chosen classifier.	0.95 x 1 x 1.4	1.33
end if	NON-AUTOMATION COST	22.19

Check test results	**COST CALCULATION** **(occ x cognitive x salary)**	**SC**
1.The ML engineer launches the Image viewer	1 x 1 x 1.4	1.4
2.The ML engineer opens the file “bestClassifierReport.png”	1 x 1 x 1.4	1.4
3. The ML engineer evaluates the classifier using the data from the report file	1 x 2 x 1.4	2.8
4. The ML engineer close the Image viewer	1 x 1 x 1.4	1.4
5. if The difference between validation and testing error is less than the tolerance interval	0.99 x 2 x 1.4	2.772
5.1 The ML engineer updates the json file “classifierEvaluation.json” with the positive response	0.99 x 1 x 1.4	1.386
5.2.The System send the classifier to the production system		
6. else	0.01 x 2 x 1.4	0.028
6.1 The ML engineer updates the json file “classifierEvaluation.json” with the negative response	0.01 x 1 x 1.4	0.014
6.2.The System request a new configuration		
	NON-AUTOMATION COST	11.2

Configure production system	**COST CALCULATION** **(occ x cognitive x salary)**	**SC**
1.The use case starts when the System administrator must configure the production system		
2. The System administrator launches the configuration panel	1 x 1 x 1.6	1.6
3. The System administrator enter the configuration parameter	1 x 2 x 1.6	3.2
5. The System administrator press the button “set parameter”	1 x 1 x 1.6	1.6
6. The System administrator closes the configuration panel	1 x 1 x 1.6	1.6
	NON-AUTOMATION COST	8

Configure evaluation system	**COST CALCULATION (occ x cognitive x salary)**	**SC**
1.The use case starts when the System administrator opens the configuration panel	1 x 1 x 1.6	1.6
2. The System administrator sets the maximum number of errors	1 x 2 x 1.6	3.2
3. The System administrator sets the maximum number of consecutive errors	1 x 2 x 1.6	3.2
4. The System administrator sets the minimum number of labels to start the evaluation phase	1 x 2 x 1.6	3.2
5. The System administrator press the button “Set parameters”	1 x 1 x 1.6	1.6
	NON-AUTOMATION COST	12.8

Evaluate classifier	**COST CALCULATION (occ x cognitive x salary)**	**SC**
1.The use case starts when the ML engineer clicks the button "show results"	1 x 1 x 1.4	1.4
2.The system shows a table having, for each session, a pair of labels (expert and classifier), the **max no. of consecutive errors,** the **total no. of errors**, and the thresholds related to the two quantities		
3. if the ML engineer finds that both thresholds are satisfied	0.86 x 2 x 1.4	2.408
3.1.The ML engineer clicks the button "Good classifier"	0.86 x 1 x 1.4	1.204
4. else	0.14 x 2 x 1.4	0.392
4.1.The ML engineer clicks the button "Bad classifier"	0.14 x 1 x 1.4	1.204
4.2.The system sends the configuration to the Messagging system.		
end if	NON-AUTOMATION COST	6.608

Configure client-side system	**COST CALCULATION** **(occ x cognitive x salary)**	**SC**
1.The use case starts when the System shows the panel for the configuration.		
2. for each parameter to be configured		
2.1.The System shows the name of the parameter and an input field for its value		
end for each		
3. for each parameter to be configured		
3.1.The  System administrator inserts the value of the parameter in the correspondent input field.	4 x 2 x 1.6	12.8
end for each		
4.The  System administrator clicks on the Apply button.	1 x 1 x 1.6	1.6
5.The System reads the values of the parameters and closes the window.		
	NON-AUTOMATION COST	14.4

System Administrator salary is £40,000

Data Analyst salary is £25,000

Machine Learning Engineer salary is £35,000

RESILIENCY (Di Ricco)

Ingestion System

ID	Input	Consequence	Score
I1	Missing records	Detected and managed by preparation system	1
I2	Missing labels	Session discarded	4
I3	Duplicated records sequentially	Unique constraint error	4
I4	Missing uuid	Record saved with missing uuid	5
I5	Missing time series samples	Detected and send to the preparation system	1
I6	Receive records from different session	Session discarded if label is missing	1
I7	Absolute outliers	Not detected and send to the preparation system	5
I8	Unbalanced classes	Not detected and send to the preparation system	5
TOTAL			26

Preparation System

ID	Input	Consequence	Score
P1	Wrong raw session structure	Validation error	4
P2	Raw session with wrong values	Not detected: prepared session will be sent with incorrect values	5
I1.2	Missing static records	Records get previous value	1
I4.2	Raw session with empty uuid	Prepared session sent to the segregation with empty uuid	5
I5.2	Missing time series values	Data interpolation	1
I6.2	Absolute outliers	Assign min/max value to data	1
I7.2	Unbalanced classes	Not detected and send to the segregation system	5
P3	Late raw sessions	System will wait until all the raw session have been delivered	5
TOTAL			27

Segregation System

ID	Input	Consequence	Score
S1	Wrong prepared session structure	Validation error	4
S2	Wrong prepared session values	Not detected: learning sets will be generated with wrong values	5
S3	Late prepared sessions	System will wait until all the prepared session have been delivered	1
I4.3	Prepared session with empty uuid	Learning sets generated with a session with empty uuid	5
I7.3	Unbalanced classes	Restart configuration msg send	3
S4	Data distribution not uniform	Restart configuration msg send	3
S5	Wrong user response after evaluating the graph	Error shown in console, the system will stop waiting for the correct response	3
TOTAL			**24**

Development System

ID	Input	Consequence	Score
D1	Wrong learning set structure	Validation error	4
D2	Wrong learning set values	Not detected: classifier trained with wrong values	5
D3	Late learning sets	System will wait until all the learning sets have been delivered	1
D4	Wrong user response after evaluating the graph (numeric values)	Not handle, Iteration made: 0	5
D5	Wrong user response after evaluating the graph (string values)	The system will stop until user sets ok command	4
TOTAL			**19**

Production System

ID	Input	Consequence	Score
PS1	Wrong prepared session structure	Validation error	4
PS2	Wrong classifier structure	Not handle, error	5
PS3	Classifier trained with wrong features values	Not detected: classifier executed	5
PS4	Wrong prepared session values	Not detected: Prediction made on wrong data	5
PS5	Late prepared session and classifier	System will wait until all the data have been delivered	1
TOTAL			**20**

Evaluation System

ID	Input	Consequence	Score
E1	Wrong label structure	Validation error	4
E2	Wrong label values (bad classifier or wrong features)	Not detected: bad accuracy report generated	5
E3	Missing uuid (just one label from one of the systems arrived)	Not detected: bad accuracy report generated	5
E4	Late label	System will wait until all the data have been delivered	1
TOTAL			**15**

INTEROPERABILIY (Cavedoni)

Feature	Ingestion System	Preparation System	Segregation System	Development System	Production System	Evaluation System	Score
A signal that allows the segregation system to request further prepared sessions if not enough have arrived	Yes	Yes	No	Yes	Yes	Yes	1
A signal that allows the evaluation system to request further labels if not enough have been received	Yes	Yes	Yes	Yes	Yes	No	1
The restart signal does not terminate the systems	Yes	Yes	No	No	Yes	No	3
Multiple clients send different sessions to the system at the same time	No	Yes	No	Yes	No	No	4
Score							9