CONNECARE

WP3 – SMART ADAPTIVE CASE MANAGEMENT SYSTEM

D3.5: SELF–ADAPTIVE CLINICAL PATHWAYS CDSS

H2020-EU.3.1: Personalised Connected Care for Complex Chronic Patients

Project No. 689802

Start date of project: 01-04-2016

Duration: 45 months

Project funded by the European Commission, call H2020 – PHC - 2015	
☑ **PU**	Public
PP	Restricted to other programme participants (including the Commission Services)
RE	Restricted to a group specified by the consortium (including the Commission Services)
CO	Confidential, only for members of the consortium (including the Commission Services)

Revision: 02

Date: 30-11-2019

Document Information

Project Number	689802	**Acronym**	CONNECARE
Full title	Personalised Connected Care for Complex Chronic Patients		
Project URL	**http://www.CONNECARE.eu**		
Project officer	Birgit Morlion		

Deliverable	**Number**	3.5	**Title**	Self-Adaptive Clinical Pathways CDSS
Work Package	**Number**	3	**Title**	Smart Adaptive Case Management System

Date of delivery	**Contractual**	Month 39 (June 2019)	**Actual**	Month 44 (Nov. 2019)
Nature	Prototype ☑ Report ☐ Dissemination ☐ Other ☐			
Dissemination Level	Public ☑ Consortium ☐			

Responsible Author	Stefano Mariani	**Email**	stefano.mariani@unimore.it
Partner	UNIMORE	**Phone**	–

Abstract	This deliverable has the goal of reporting on the activities carried out to develop a prototype of the CDSS for clinical pathways. Accordingly, Section 1 introduces the document by motivating the need for the Pathways CDSS and its goals, Section 3 reports on the requirements collection stage informing the Pathways CDSS design, Section 2 briefly summarises the state of art of research works in clinical pathways related ICT tools, Section 3 presents the requirements, Section 4 describes the design of the Pathways CDSS, Section 5 describes the implemented prototype, Section 6 discusses next steps, and Section 7 concludes the document.

Table of contents

EXECUTIVE SUMMARY **4**

1. INTRODUCTION **6**

2. STATE OF ART **8**

2.1 RESEARCH ON CLINICAL PATHWAYS 8

2.2 AI METHODS FOR CDSSS 8

3. REQUIREMENTS **10**

3.1 NON-FUNCTIONAL REQUIREMENTS 10

3.2 SITE-SPECIFIC REQUIREMENTS 11

4. SOLUTION DESIGN **13**

4.1 DRUG-DRUG INTERACTION 13

4.2 SIMPLE TASKS SUGGESTION 13

4.3 SUGGESTION OF INTERVENTIONS 15

4.4 THE PROPOSED ARCHITECTURE 16

5. THE CDSS PROTOTYPE **19**

5.1 THE WEB SERVICE FOR DRUG-DRUG INTERACTION 19

5.2 THE RULE ENGINE FOR SIMPLE TASKS SUGGESTION 21

5.3 THE ML PIPELINE FOR DELIVERING SUGGESTION OF INTERVENTIONS 22

5.3.1 Advise to use LAMA *24*

5.3.2 Advise to use both ICS and LABA *26*

5.3.3 Advise to use β_2 bronchodilation *28*

5.3.4 Definition of diagnosis *30*

5.3.5 Summary of models evaluation *32*

5.4 ON INTEGRATION 32

6. OUTLOOK OF FUTURE RESEARCH AND IMPLEMENTATION **36**

7. CONCLUSION **37**

8. BIBLIOGRAPHY **38**

9. LIST OF FIGURES **40**

Executive Summary

This document describes the activities performed towards design of the Clinical Decision Support System (CDSS) meant to accomplish **Task 3.6 "Self-Adaptive Clinical Pathways CDSS"**, as well as the co-design process adopted in order to maximize likelihood of adoption in clinical sites. This document is thus the result of collaborations with the clinical partners and data science teams of the clinical sites, in particular, ASSUTA and UMCG.

This deliverable directly contributes to the specific objectives of WP3 regarding the provisioning of "*ICT tools for the adaptive case management of personalised clinical pathways (OBJ2), which takes into account the patient's medical history*" (objective OBJ2 in DoA). Also, it contributes to make "*Healthcare professionals* [to be] *continuously and proactively supported in their decisions through Decision Support Systems (DSSs) for: screening and risk stratification; mapping; and intervention and surveillance (suggesting personalised clinical pathways)*" (OUT3-4 in DOA).

Complementary to the design of the software tool described in this deliverable is that of the DSS for Risk as described in D3.2 *"First Screening and Risk Stratification DSS"* and D3.4 *"Stratification and Mapping DSS"* (the latter released simultaneously to the current one). It is also worth mentioning that reading D3.6 *"Final Smart Adaptive Case Management System"* is also recommended, as it describes the SACM system as a whole, with which all the three DSSs are meant to integrate or interact.

The table below summarizes the suggested readings and their role with respect to the present document.

Table 1 List of related deliverables, either preparatory or complementary.

Number	Title	Description
D2.3	Patient-based Health Risk Assessment and Stratification	This CONNECARE document (D2.3) has a threefold aim. Firstly, to describe the consensus achieved by the clinical partners of the consortium regarding conceptual and pragmatic aspects of health risk assessment. The document proposes an operational formulation of enhanced clinical risk predictive modelling to be adopted in CONNECARE (Task 2.3). It has been worked out iteratively with Task 3.4 for elaboration of clinical decision support systems (CDSS) supporting dynamic risk assessment using multilevel data sources.
D3.2	First Screening and Risk Stratification DSS	This deliverable has the twofold goal of *(i)* reporting on the development activities carried out to deliver the 1st prototype of the risk assessment DSS for screening and risk stratification, as well as of *(ii)* describing the resulting software artefact. Accordingly, section 1 motivates and gives context to the work done, section 3 summarises the requirement collection phase, then we describe the

		DSS architecture, section 5 provides technical details on the implementation, section 6 looks forward to future iterative improvement steps, and section 7 concludes the document.
D3.4	Stratification and Mapping DSS	This deliverable has the threefold goal of *(i)* reporting on the activities carried out to improve the 1st prototype of the risk assessment DSS presented in D3.2 "First Screening and Risk Stratification DSS", *(ii)* reporting the same for the mapping DSS currently released in CONNECARE production environment, and *(iii)* describing the resulting software artefacts. Accordingly, section 1 motivates and gives context to the work done, section 3 recaps the main characteristics of the DSS for risk assessment and summarises the improvements done.
D3.6	Final Smart Adaptive Case Management System	This deliverable goes with the final release of the Smart Adaptive Case Management system (SACM) by TUM and ADI, integrated to the SMS by EURECAT and the contribution of UNIMORE for the clinical decision support systems.

This document is an update of the preliminary version submitted by the end of June, reporting on a first analysis of available solutions and design of the DSS for Clinical Pathways. As better commented in Section 1, due to the evolutionary approach of the project and the requirements for the implementation studies, it has not been possible to have the CDSS available for the implementation studies in the 4 sites of CONNECARE. Nevertheless, a stand-alone software is ready and has been evaluated in a lab setting, and future integration with systems like SACM has been already designed from a technical perspective, as outlined in Section 5.4.

CONNECARE project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement Nº 689802 (Art. 29.4)

1. Introduction

The proposal of ICT tools for the analysis, synthesis, and execution of clinical pathways is motivated by the fact that, especially for complex chronic patients, treatment plans can quickly become complex, in turn, hence software support to help clinicians in systematically analysing all the available patient information to devise out the best treatment plan tailored to the patient needs would be highly beneficial. Within CONNECARE, clinical partners expressed both the desire to have ICT tools specifically dedicated to suggest interventions based on the clinical data of the patient, but also concerns about the extent to which they should rely on such tools in everyday clinical practice, from an ethical and liability perspective.

Accordingly, the goal of the activities reported in this document is to develop a CDSS for the automatic suggestion of specific interventions along a treatment plan ("Pathways CDSS" henceforth). The technical goals of such a tool would be to propose to the clinicians interventions in the form of, for instance, simple tasks for pre/re-habilitation in surgery, or usage of medications for asthma / COPD syndromes, whereas the research goal is to check feasibility of such automated approach even for small datasets and commodity hardware and software, conversely to what happens with big companies' products such as IBM Watson, which require loads of carefully crafted data and raw computational power to process it – and still is far away to provide "automated diagnosis" and recently re-purposed its goals to more realistic decision support in suggesting interventions [1].

Implementation studies have finished in M42 (whereas the recruitment at M40) in each clinical site covered by CONNECARE clinical partners. Since the studies started, data generated by patients through the adoption of the SMS during their participation in the CONNECARE system, by filling questionnaires and feeding automated monitoring tools, as well as data from the SACM consisting on questionnaires and forms filled by clinicians during the overall work plan, have been collected. Such data are crucial for the design of advanced ICT tools to give support to the professionals such as CDSS. In fact, the definition and implementation of the CDSS needs and leverages data analysis and machine learning techniques to provide insights and suggestions about personalised clinical pathways. Such techniques require large amounts of data, e.g. describing the conditions of a large number of patients (tenths of thousands) sampled for prolonged time windows (tenths of months to few years, depending on the attribute of interest), to generate scientifically sound and relevant clinical knowledge. Also, data exploratory analysis, pre-processing, and models training are time-consuming tasks in which a lot of trial-and-error is involved (training a single model can require hours, and fine-tuning the parameters of a model requires multiple training stages).

In the case of CONNECARE, gathered data cannot cope with these needs, as patients enrolled in the implementation studies are about 220 and data spans different time frames that depend on CS1 and CS2 definition in each site. Moreover, clinical pathways vary from site to site and, thus, a different model in each site should be defined, reducing further the number of patients to build the models. In other words,

application of Artificial Intelligence (AI) techniques such as Machine Learning (ML) for training an automated DSS based on CONNECARE data would be unfeasible and meaningless, as it would lead to results not scientifically relevant because stemming from a limited representation of the population.

The lack of data has thus prompted CONNECARE consortium to find solutions to advance towards definition and design of the Pathways CDSS in a different way. Section 4 of the present document describe the decisions made to overcome the aforementioned issues. In particular, as better clarified in the aforementioned sections, the Pathways CDSS focusses on interventions suggestion based on risk prediction and available clinical data contextual to each individual patient, as generated by a past project of UMCG partner involving COPD and Asthma patients, shared with UNIMORE under a non-disclosure agreement. Given a pool of patients data, the Pathways CDSS is trained to find hidden patterns and correlations between clinical conditions, measurements, lab tests, etc. and outcomes such as diagnosis, medications, advices, interventions in general, to learn to suggest such interventions on its own for each individual patient introduced in the system, once appropriate data is available –e.g., in the case of CONNECARE SACM after the Case Evaluation stage, to suggest elements of the workplan to be defined by the clinician.

The Pathways CDSS has been recognised by clinical partners of the CONNECARE consortium, on behalf of the clinical staff at their sites, as a valuable experimental feature, hence they accept it as an interesting, valuable, and potentially impactful research effort / product, and agree on testing it in a lab setting in the spirit of the research that a Research and Innovation Action such as CONNECARE should foster and leverage. Even if the system has not been considered mature enough to be available during the implementation studies, its integration has been already planned and could be applied with minor modifications and ready for further investigation or adoption, as discussed in Section 5.4.

2. State of art

2.1 Research on clinical pathways

Research work in electronic (or, computerized) clinical pathways can be roughly divided in three macro-areas:

- **Clinical pathways analysis**. In this category, ICT tools are created either *(i)* to analyse real-world clinical pathways so as to build their digital representation, or *(ii)* to allow clinicians to define computer-interpretable specifications of pathways to be then enforced in clinical practice, usually with the goal of checking compliance or spotting bottlenecks and devise out reasons for under-performance of care plans. In [2] for instance, *process mining* is used to mine Electronic Health Records (EHR) and build the resulting clinical pathways a posteriori. In [3] instead, an ontology-based approach is proposed for the definition of pathways in a workflow style.
- **Clinical pathways synthesis**. In this category, the proposed ICT tools focus on either *(i)* automated definition of clinical pathways, or *(ii)* enforcement and execution thereof, with the goal of practically assisting clinicians in the definition and enactment of care plans. Both [4] and [5] for instance, propose a semantic rule-engine configured with domain expert knowledge and a suitable rule-set so as to suggest adaptations to care plans depending on the specific patients conditions or unexpected events.
- **CDSS within clinical pathways**. In this category, the focus is not on the whole clinical pathway, but on the kind of CDSSs that can profitably support both the pathways definition and their enactment. The work in [6] for instance proposes a support system to suggest to clinicians the next steps to accomplish depending on the ever evolving patients conditions.

The work described in this deliverable is positioned in the latter area, as clinical partners in the CONNECARE consortium are mostly interested in receiving support at specific stages of already defined care plans.

2.2 AI methods for CDSSs

The research landscape proposing how to build the needed CDSSs may be in turn roughly divided to follow three well established research lines:

- **Case-based Reasoning**. Case-based Reasoning is a methodology proposing the core idea of looking at past cases to find solutions to new ones: a new clinical case is compared against a knowledge base of old ones, the most similar past case is then find and selected, the successful treatments for that case are taken as a blueprint for new one, and finally adaptations to those treatments are made to tailor the suggestion to the peculiarities of the new case. Case-based reasoning may remain a methodology, a conceptual framework to guide CDSS implementations,

or become the actual implementation backbone of a CDSS: in this latter case the typical implementation features a custom database optimised for similarity computation and case retrieval, and a rule engine to perform adaptations. Case-based Reasoning is a well-established methods for delivering intelligence in healthcare systems, and applications range from headache [7] and diabetes [8].

- **Rule / Workflow engines**. Likewise Case-based Reasoning, rule and workflow engines have both a long history of application in clinical practice, as they guarantee predictable outcomes due to the fact that, traditionally, they are based on expert clinical knowledge encoded in some computer-interpretable format. This line of research usually produces software tools able to monitor clinical events so as to trigger appropriate action when specific conditions hold, regardless whether they are implemented as full-fledged workflow engines [9] or rule engines [10].
- **Machine Learning**. This latter research line, in which we include the even more recent deep learning research field, is both the most recent and the most actively exploited nowadays. The promise of letting algorithms crunch thousands of clinical cases data and get back to the clinician with a treatment plan based on empirical evidence is too alluring not to be actively pursued [11] [12]. Accordingly, a plethora of innovative approaches to CDSS and even automated diagnosis emerged in the last decade, with applications ranging from analysis of physiological signals [13] to medical imaging [14].

All the methods to build CDSSs for clinical pathways described above can be applied within the CONNECARE project to deliver functionalities meeting the requirements raised by clinical partners of the consortium, as each approach is essentially independent of the application domain and the specific goal pursued –as long as it regards triggering actions based on monitored events, and specific conditions.

The choice UNIMORE made in designing and implementing the Pathways CDSS is to use both a rule-based approach and machine learning techniques, depending on the functionality:

- for triggering drug-drug interaction alerts, neither of the two is needed as 3rd party services can be exploited, hence a dedicated web service is implemented
- for delivering simple tasks suggestions a rule-based approach is preferred, as the simple tasks required by clinicians in Section 3.2 are well suited to be expressed accordingly
- for automated suggestion of interventions (diagnosis and delivering advices) in the context of asthma / COPD patients, a suitable machine learning pipeline is set up, as UMCG provided a suitable dataset to analyse and exploit (described in 5.3)

3. Requirements

Requirements collection has been conducted mostly in collaboration with clinical partners in the ASSUTA and UMCG sites. As already described in Section 1, the advanced and experimental nature of the Pathways CDSS led clinicians to be cautious about the expected functionalities of the software tool, which turned out to be more akin to desirable features than strict requirements on delivered functions. In this section, requirements have been conveniently divided into two categories: non-functional requirements about the structure and properties of the software tool, such as how it should operate and how users should interact with it, and site-specific requirements about the expected functionalities the tool should deliver as value added in the context of daily clinical practice, such as which suggestions it has to provide.

3.1 Non-functional requirements

Non-functional requirements expressed by clinical partners are similar to those expressed for the design of the DSS for risk screening and assessment as described in D3.2. In particular:

- **Plugin operation mode**. It is desirable that the Pathways CDSS is able not only to train models on its own, but also to work with already trained models, as the DSS for risk screening and assessment. The value added by this mode of operation has been appreciated both by the partners of the consortium and by the scientific community, and witnessed by acceptance of a manuscript on the topic at an international conference on computer-based medical systems [15]
- **Transferability**. Factors such as *(i) license* binding constraints, *(ii) proprietary* software; *(iii)* lack of availability for *inspection*, and/or *(iv) rigidity* of computational algorithms are limiting factors for transferability across regions and domains. The Pathways CDSS should thus be developed with an effort to *(i-ii)* rely on *open-source* software with adequate licensing, *(iii)* be open not only to source code inspection, but also to run-time inspection of *working parameters*, *(iv)* be flexible enough to allow for both design-time (*extensibility*) and run-time (*configurability*) change.
- **Privacy and security**. Issues related to privacy, security of data transfer, as well as risks associated with decisions are traditionally brought along by the very essence of the healthcare domain. It is thus of foremost importance that the DSS is developed with the necessary privacy-preserving and security-compliant *methodologies* and *technologies*.
- **Accessibility**. The Pathways CDSS functionalities should be accessible from clinicians' workstations, and, thus, integrate it with the SACM system, in the future. It is worth noting that the CDSS should be not perceived as a separate entity by the clinicians, but as a "behind the scenes" service provided by CONNECARE and accessible (visible) through the SACM User Interface.
- **Standards**. The Pathways CDSS services should rely on standards to be seamlessly available to any software client, regardless of its implementation language. For instance, XML / JSON data

formats and REST APIs guarantee seamless interoperability with the vast majority of programming languages.

3.2 Site-specific requirements

Specific requirements naturally differ depending both on the goals of clinical partners in each clinical site and also on the implementation studies in progress at the site. These kind of requirements are, as already said, more akin to desirable features that clinicians would like to play with in an experimental environment, rather than strict requirements on functionalities to see production-ready in the SACM.

In ASSUTA site, for instance, two functionalities have been deemed worth of attention, the former as it is becoming increasingly standard in modern medical systems (although not yet available in legacy systems pervading most clinical structures), the latter as a nice value added currently missing from ASSUTA software tools eco-system, including CONNECARE SACM:

A. **Drug-drug interaction**. A first level of decision support, recognised as being very basic but at the same time valuable for clinicians in daily activities, is the detection and triggering of alerts for drug-drug interaction during prescription of medications. In other words, a valuable functionality not requiring any sophisticated machine learning technique, hence any dataset to analyse, is the display of alerts whenever a clinician prescribes a drug which is known to have interactions with another drug. The kind of alert triggered may vary depending on the severity, likelihood, and size of population in which the interaction manifested. Existence and suitability of 3rd party databases or services for doing so are analysed in Section 4.1.
B. **Simple tasks suggestion**. A second level of decision support that would instead fill a gap in current software tools available in ASSUTA is the ability of the Pathways CDSS to suggest to the clinician which tasks related to surgery pre/re-habilitation would benefit the patient the most, if prescribed. Acceptance of the suggestion is anyways left to the expertise of the clinician, which may decline the suggestion. The kind of tasks which could be suggested mostly include simple physical exercises such as[1]:
    a. Straightening and bending the knee while lying on the back; strengthening the elbow, buttocks, and fingers; strengthening the thigh
    b. Keeping the leg away from the body while lying on your back
    c. Lifting the pelvis lying on the back with bent legs; lifting the leg upright while lying on a side

[1] These tasks are those that currently can be prescribed by clinicians through the SACM and that the patient has to report if s/he did or did not, through the SMS. This functionality is available, according to the CONNECARE requirements, only in ASSUTA. The full list of rehabilitation task is given in D4.3 “Advanced monitoring tools” and provided by the “Simple task” service.

d. Bending the knee and bringing it to the stomach while lying on the back; bending the knee in the sitting; bending the knees standing with the back to the wall
e. Straightening and bending the seated knee
f. Getting up from the chair and sitting back
g. Walking outside at moderate speed; up and down stairs
h. Training session for physiotherapy

How this kind of suggestions can be implemented in the CDSS, such as which kind of technical tools they need, is discussed in Section 5.2.

In UMCG site, instead, focus is on asthma / COPD patients, hence any kind of **suggestion of interventions** for those conditions are welcome. Here, while full discussion is again delayed to Section 5.3, we anticipate that UMCG would like to craft suggestions based on the dataset they already used in [1], which has been shared with UNIMORE with a bilateral non-disclosure agreement. Exploratory analysis of such dataset has been already described in Appendix 8.1 of the deliverable D3.4, hence is here omitted to avoid repetition. Nevertheless, whenever additional data analysis or pre-processing has been done, it is dutifully described in Section 5.3. UMCG identified the following potential targets for suggestion:

A. **Advices**. About advising patients to take a given treatment or action, for instance:
   a. Advise to use Long-Acting Muscarinic Antagonists (LAMA)
   b. Advise to use (low / high dose of) Inhaled Corticosteroids (ICS), or Long-Acting β-Agonists (LABA), or both
   c. Advise to use β_2 bronchodilation
B. **Diagnosis.** The most valuable suggestion that the Pathways CDSS could provide to clinicians is the diagnosis of the disease, for instance, amongst the following alternatives:
   a. Asthma/COPD overlap
   b. COPD
   c. Asthma
   d. Diagnosis unclear

   Nevertheless, this is also the most difficult suggestion to obtain as well as the most critical to deliver: the former because of the many factors that contribute to define a diagnosis, which is often times also enhanced by the expertise of the clinician (that machine learning cannot emulate, yet), the latter because of the same ethical issues concerning liability of accepting automated diagnosis in clinical practice mentioned in Section 1.

4. Solution design

4.1 Drug-drug interaction

A few ready (and free) to use databases and services upon which the drug-drug interactions functionality can be implemented have been found and analysed. For instance, a few drug-drug interaction databases are actively exploited by health organisation worldwide, both in Europe and the United States. They can be easily integrated with the list of medications already present in the SACM as defined by each clinical site: the online list supported by Agencia Española de Medicamentos y Productos Sanitarios (https://www.aemps.gob.es/home.htm) in Lleida site, an ad-hoc list of drugs selected from Farmacotherapeutisch Kompas (https://www.farmacotherapeutischkompas.nl/) in Groningen, and a custom list provided by clinical partners for ASSUTA (see deliverable D4.3 for more details).

A widely used database is, for instance, **DrugBank**[2], maintained, amongst other, by the **Canadian Institutes of Health Research**. DrugBank is *"a comprehensive, freely accessible, online database containing information on drugs and drug targets. As both a bioinformatics and a cheminformatics resource, DrugBank combines detailed drug (i.e. chemical, pharmacological and pharmaceutical) data with comprehensive drug target (i.e. sequence, structure, and pathway) information."* DrugBank is a freely available resource allowing usage and re-distribution of the data, in whole or in part, for non-commercial purposes. The **U.S. National Library of Medicine** (NLM), instead, maintains a set of free-to-use services[3] amongst which a **"Drug Interaction RESTful API"** is available[4]. Such API allows to retrieve drug-drug interaction information mined from multiple drug datasets, amongst which DrugBank itself. Although API rate limiting is put in place to avoid too many requests, a Docker version of the service is available from download, to be deployed on on-premise servers (hence avoiding API restrictions).

Section 5.1 describes how the Pathways CDSS exploits such service to deliver the intended alerts.

4.2 Simple tasks suggestion

Many rule engine frameworks exist for many programming languages, but most of them are geared towards complex business rules involving triggering of many complex computations upon monitoring of many heterogeneous events. Here instead, our aim is that of providing a simple, easy to use and easy to understand rules engine, possibly which can be configured even by non-programmers, such as directly by clinical staff.

[2] https://www.drugbank.ca/about

[3] https://rxnav.nlm.nih.gov/APIsOverview.html

[4] https://rxnav.nlm.nih.gov/InteractionAPIREST.html

Easy-rules[5] has exactly this goal, as it enables to specify rules in configuration files exploiting the MVEL[6] or SpEL[7] languages written in the YAML format[8], which somewhat resembles JSON and is specifically conceived for being easily understandable by non-programmers. MVEL and SpEL are two well-known expression languages which enable to express If-Then-Else rules such as

```
rule "apply 10% discount to all items over US$ 100,00 in an order"
when
    $order : Order()
    $item  : OrderItem( value > 100 ) from $order.items
then
    # apply discount to $item
end
```

where:

- Keyword `rule` provides for a description of the rule
- Keyword `when` specifies the conditions to apply the rule
- Keyword `then` specifies the action to perform if and when the condition is satisfied

In the framework of Easy-rules, the above MVEL specification would be translated in a YAML file as follows

```
name: "10_discount_over_100"
description: "apply 10% discount to all items over US$ 100,00 in an order"
priority: 2
condition: "$item.value() > 100"
actions:
  - "$item.apply($discount)"
```

where:

- `name` is a convenience name for the rule and description a natural language description of its purpose
- `priority` enables to control the order of triggering rules in case conditions are the same
- `condition` monitors when the rule should be applied or discarded

[5] https://github.com/j-easy/easy-rules

[6] https://en.wikibooks.org/wiki/Transwiki:MVEL_Language_Guide

[7] https://docs.spring.io/spring/docs/4.3.10.RELEASE/spring-framework-reference/html/expressions.html

[8] https://yaml.org/spec/1.2/spec.html

`actions` is the list of actions to execute, which usually refers to some existing Java code---for instance, in the example above `$item` and `$discount` refer Java objects and `apply()` is a Java method.

Section 5.2 describes how such a Java library has been used to deliver the simple tasks suggestions functionality, and how encoded rules actually look like.

4.3 Suggestion of interventions

Besides usage of the UMCG dataset as described in Section 5.3, exploitation of a recently proposed third-party service has been checked for suitability and seems to provide the required functionalities to link patient clinical conditions to diagnosis, then to medication prescriptions: the **Patient-Disease-Drug Graph**[9] (PDD Graph) [15].

PDD Graph is a large graph linking patients, diseases, and drugs in EMRs mined from the **MIMIC-III** data mart [16], by using existing biomedical knowledge graphs such as the **ICD-9 ontology**[10] and the already mentioned DrugBank. This way, the gap between clinical data and biomedical knowledge graphs can be filled: MIMIC-III contains multi-format electronic data including clinical outcomes, while biomedical knowledge graphs cover basic medical facts, but contain little information about clinical outcomes. In other words, as depicted in Figure 1, PDD Graph allows to associate a diagnosis to a set of clinical measures (thanks to MIMIC-III), and then to find suitable prescriptions (thanks to DrugBank), all based and guided by data patterns mined from MIMIC-III databases.

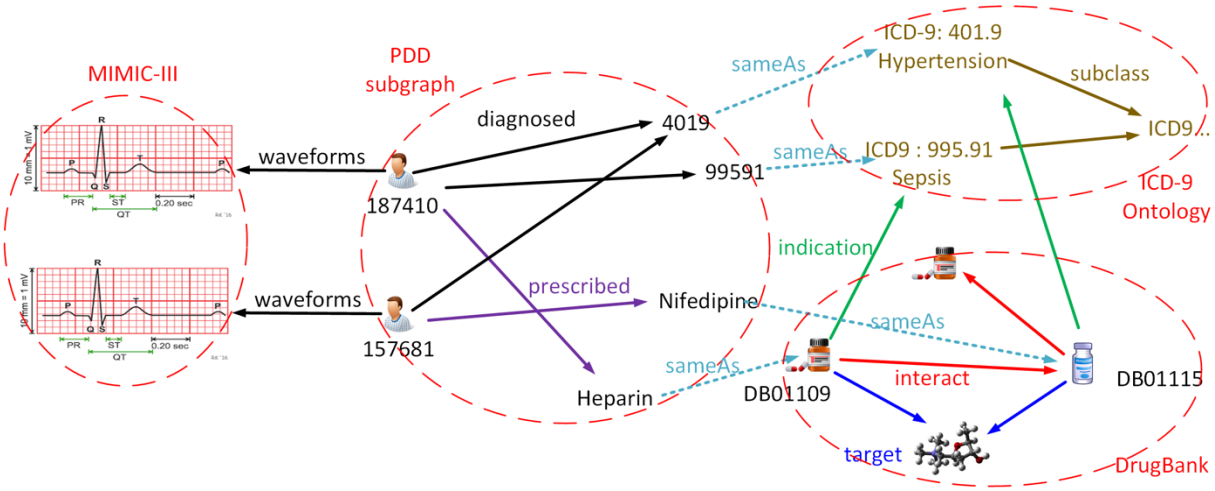

Figure 1 The PDD Graph, connecting data mined from MIMIC-III to ICD9 diagnoses, and DrugBank prescriptions (image taken from https://kmap.xjtudlc.com/pdd/index.html#Introduction).

[9] https://kmap.xjtudlc.com/pdd/index.html#Introduction

[10] http://bioportal.bioontology.org/ontologies/ICD9CM

However, after careful inspection of the dataset and experimentation of the PDD graph tool, it has been decided not to use it for the following reasons: *(i)* usage of semantic web technologies presents a steep learning curve for both technicians and potentially clinicians, compared to the simple rule engine or off-the-shelf machine learning algorithms, *(ii)* the UMCG dataset is more representative of the actual CONNECARE patients, and explicitly of interest for the partners. Nevertheless, the PDD graph should be kept in mind for future exploitation beyond CONNECARE lifespan and for additional implementation studies.

4.4 The Proposed architecture

The Pathways CDSS is designed to be a **service** in the CONNECARE ecosystem, likewise the DSS for risk assessment and screening. Although in this current version clinicians are not going to use it through the SACM, its design follows the service-oriented perspective for the sake of future integration and to easily made it available for further exploitations beyond CONNECARE project lifespan.

Figure 2 depicts the proposed architecture, emphasising interaction with *(i)* the aforementioned third-party services (on the right), and *(ii)* data made available by clinical sites to satisfy their requirements (on the left).

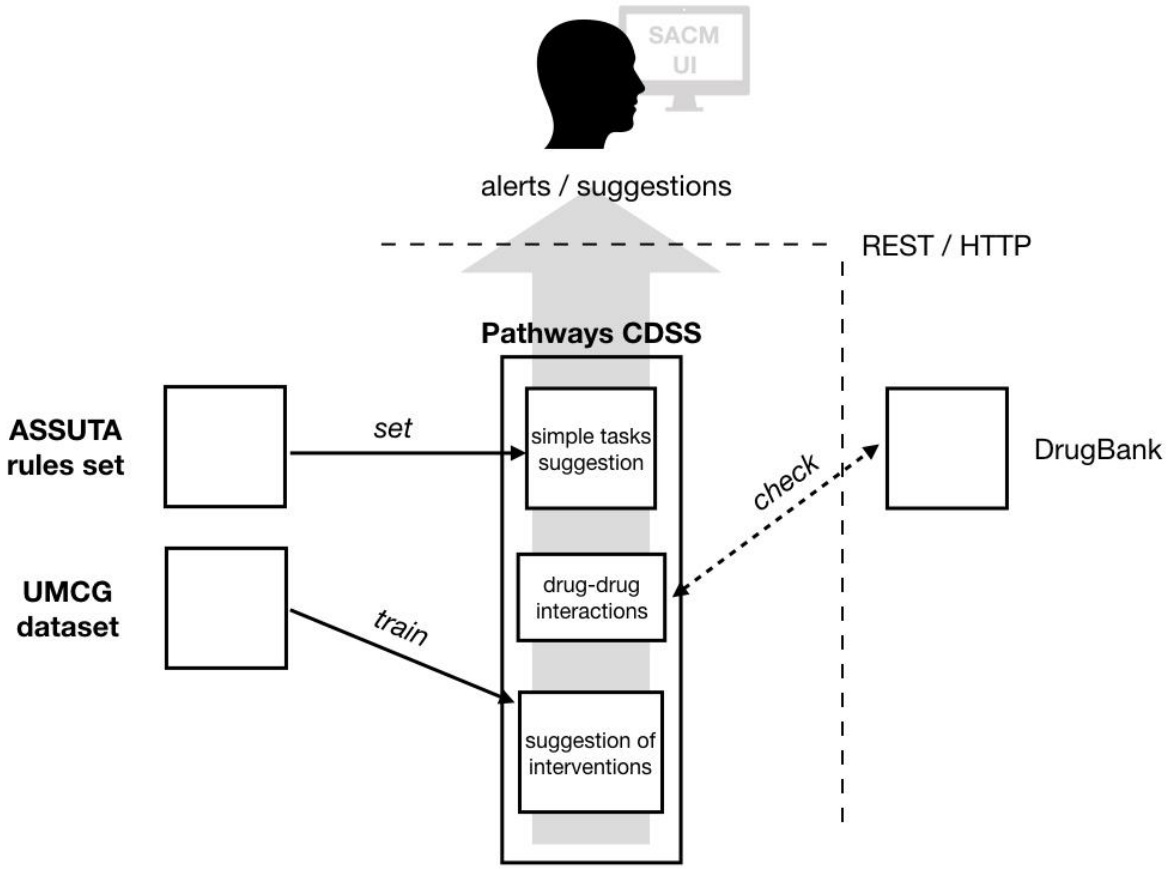

Figure 2 The proposed Pathways CDSS architecture.

The Pathways CDSS is composed by three main modules corresponding to the three aforementioned functionalities:

- A **web service** delivering drug-drug interactions alerts, by checking the medications prescribed by each of clinical sites in comparison with the DrugBank.
- A **rule engine** delivering simple tasks suggestions, encoding ASSUTA requirements for pre/re-habilitation exercises in the form of "If-Then-Else" rules, dictating which suggestion should be triggered, under which circumstances. As an example, such rules may establish to suggest the clinician to prescribe a given set of physical activities amongst the pool of those listed in Section 3.2 as soon as the patient is registered for surgery.
- A **ML pipeline** delivering suggestions of interventions, built by working with UMCG dataset, leveraging on *classification* techniques to predict advices and prescriptions given symptoms and other clinical data.

The architecture just described leverages modularity to enable independent development of the different components, and loose coupling with external services. It is actually implemented as a standalone web

service, as for the DSS for risk assessment and screening, so as to be easily integrated in the SACM, and, in further exploitations, in other systems.

5. The CDSS prototype

5.1 The web service for drug-drug interaction

The web service for drug-drug interaction (DDI henceforth) has been implemented as a RESTful web service currently connected to a fake 3rd party service (meant to represent the SACM) for testing in a lab setting. As already said, it relays on the DrugBank drug-drug interaction dataset exposed by the Drug Interaction RESTful API service[4], as well as the RxNorm RESTful web API[11]. In particular, the latter is required by the former, as the Drug Interaction RESTful API allows searching for drugs (hence their interactions) by using their RxCUI codes, which are unique identifiers for drugs defined in the RxNorm system[12], that is a free, publicly available resource that provides "normalized" names for brand-name and generic drugs and supplies a unique identifier for each entity –making it possible to unambiguously identify a given drug.

Due to the above requirements, the DDI works as follows: whenever prescription of a given drug is triggered by a 3rd party service (e.g., the SACM) through a dedicated RESTful endpoint exposed by the DDI, this contacts the RxNorm API[13] first to resolve the drug name unambiguously (this step is required as misspelling a drug name, using a synonym, or using abbreviations may be common in medical practice, but does not cope well with RxCUI codes lookup), then to lookup the RxCUI code of the drug with the resolved name[14]. Once the RxCUI code is obtained, the DDI contacts DrugBank to fetch the list of known interactions[15], and finally publishes the results to a configured 3rd party service (e.g. the SACM).

It is worth emphasizing here that, as for the rule engine module, the DDI is structured this way because it is meant to be integrated in an existing platform, such as the CONNECARE SACM: in that case, the SACM would trigger drug interactions lookup anytime needed by the workflow of the clinical professionals (e.g. after a prescription is done in the workplan), and the CDSS would publish to the SACM the list of known interactions fetched by DrugBank, do be displayed to the clinician.

As an example useful to understand the kind of information that the DDI could provide to 3rd party services, hence display to clinicians, the following is an excerpt of the information returned by DrugBank in JSON format:

[11] https://mor.nlm.nih.gov/download/rxnav/RxNormAPIREST.html

[12] https://rethinkingclinicaltrials.org/resources/using-the-rxnorm-system/

[13] https://mor.nlm.nih.gov/download/rxnav/RxNormAPIREST.html#uLink=RxNorm_REST_getApproximateMatch

[14] https://mor.nlm.nih.gov/download/rxnav/RxNormAPIREST.html#uLink=RxNorm_REST_findRxcuiByString

[15] https://rxnav.nlm.nih.gov/InteractionAPIREST.html#uLink=Interaction_REST_findDrugInteractions

```
...
"interactionPair": [
            {
              "interactionConcept": [
                {
                  "minConceptItem": {
                    "rxcui": "341248",
                    "name": "ezetimibe",
                    "tty": "IN"
                  },
                  "sourceConceptItem": {...}
                },
                {
                  "minConceptItem": {
                    "rxcui": "5011",
                    "name": "Gramicidin",
                    "tty": "IN"
                  },
                  "sourceConceptItem": {...}
                }
              ],
              "severity": "N/A",
              "description": "The serum concentration of Ezetimibe can be increased when it is combined with Gramicidin D."
            },
            {
              "interactionConcept": [
                {
                  "minConceptItem": {
                    "rxcui": "341248",
                    "name": "ezetimibe",
                    "tty": "IN"
                  },
                  "sourceConceptItem": {...}
                },
                {
                  "minConceptItem": {
                    "rxcui": "3008",
                    "name": "Cyclosporine",
                    "tty": "IN"
                  },
                  "sourceConceptItem": {...}
                }
```

```
            ],
            "severity": "N/A",
            "description": "The serum concentration of Ciclosporin can be increased when it is
combined with Ezetimibe."
          },
...
```

In the excerpt above:

- The “interactionPair” field is the list of known interactions (in the example it featured over 500 interactions)
- Each item in the list (enclosed in curly braces) has three fields
  - An “interactionConcept”
  - A “severity”, which indicates the known severity of the interaction
  - A “description”, which is a textual description of the interaction
- The “interactionConcept” always has two “minConceptItem” fields
  - The former is always the drug looked up for interactions
  - The latter is the drug with the described known interaction (in boldface font in the example)

Field “name” of this second “minConceptItem” together with “description” is the information that the DDI could store and dispatch to the configured 3rd party service, as envisioned in Section 5.4 for the SACM.

5.2 The rule engine for simple tasks suggestion

The rule engine has been implemented as a RESTful web service in line with the other modules of the CDSS for pathways. As the specific software framework used, the Java library Easy-rules[16] has been used.

In practice, the rule engine module of the CDSS for pathways works as follows: the user (e.g. the domain expert, such as a clinician) defines the rules in a YAML specification file similar to the one above and uploads them to the CDSS service (as a single file upload); then the CDSS offers RESTful endpoints for 3rd party applications to trigger rules execution, and waits for such triggering to happens; finally, when this happens, the CDSS executes the rule, whose outcome is the interaction with another 3rd party application (again, in the form of a custom RESTful endpoint to contact).

The reason why the rule engine module is structured this way is that it is meant to be integrated in an existing platform, such as the CONNECARE SACM: in that case, the SACM would trigger rule executions anytime needed by the workflow of the clinical professionals (e.g. after an evaluation stage when surgery pre-habilitation is deemed necessary), and the CDSS would publish to the SACM the suggestion of the

[16] https://github.com/j-easy/easy-rules

activities to prescribe, as a consequence of the `actions` part of the rule specification. A depiction of the process is provided in Section 5.4.

It is worth emphasising here that rules definition would anyway require some level of support from a technical expert, as even if the YAML and MVEL syntax are human-readable, definition of conditions and actions would necessary refer to some programmed Java object and methods that the clinical user has to know. However, we believe this is indeed acceptable as a compromise with respect to having all hard-coded and immutable rules that clinicians cannot inspect, change, add, nor remove.

To better illustrate rules definition, here follows the translation to YAML of rule (f) described in Section 3.2 ("Getting up from the chair and sitting back") –keep in mind that the whole ASSUTA ruleset is meant to assist clinicians in prescribing pre-/re-habilitation activities, hence they share the same condition:

```
name: "chair_exercise"
description: "Getting up from the chair and sitting back"
priority: 1
condition: "sacm.evaluation.done($patient)" AND "sacm.workplan.surgery($patient)"
actions:
  - "sacm.workplan.prescribe($patient, activity("chair", 3, 10, 7))"
```

The conditions actually ensure that the input 3rd party service (in this example the SACM) has informed the rule engine module that rule activation is necessary for a given patient. It is worth emphasising that as the CDSS for pathways has not been integrated with the SACM, yet, the 3rd party service actually hard-coded in the CDSS implementation (both for input, in the condition, and for output, in the actions) is simply a web service setup for testing purpose. Nevertheless, we use the "sacm" keyword to be ready for its actual integration.

The actions command the rule engine to activate the correct procedure regarding another 3rd party service (in this example the SACM, again). In particular, execution of the rule triggers a call to a RESTful endpoint hard-coded in the CDSS, with the given parameters. The syntax `activity("chair", 3, 10, 7))` is what actually encodes the activity to suggest to the clinician for prescription: the first argument is a keyword identifying the kind of activity, while the other arguments are the parameters of the prescription (in the above example, do the activity 3 times a day, 10 repetitions, for 7 consecutive days). Would we want to encode activity (e) described in Section 3.2 (e.g. "Straightening and bending the seated knee"), syntax would have been something like `activity("knee_seated", 3, 15, 5))`. In other words, the rule engine maps a set of "activity keywords" to a pre-defined text to be sent to the connected 3rd party service (e.g. to be displayed in the SACM).

5.3 The ML pipeline for delivering suggestion of interventions

UMCG provided UNIMORE with a dataset on asthma / COPD patients under a non-disclosure agreement as the dataset has not been generated within CONNECARE, but in another past project. The dataset

contains baseline assessment of 19077 patients' conditions concerning Asthma / COPD, and follow-ups information at different time points (mostly, 3 and 12 months, as per below analysis). Attributes describe basic data such as age, gender, BMI, and family history of the disease, habits associated to the syndrome such as smoking and inhalation technique, lab measurements such as Forced Expiratory Volume in 1 second (FEV), Forced Vital Capacity (FVC), and lung function assessed by pneumologist, symptoms such as coughing, short breath, and wheezing, questionnaires results such as ACQ and CCQ, and medications taken. All the measurements, symptoms, questionnaires, and in general any attribute whose value may change over time is repeated at each follow up.

Follow-ups are not homogeneous in the time interval between each other, hence data samples have been grouped according to the most common follow-up intervals, which are at 3 and 12 months. Then, the dataset has been restricted to samples (patients) having follow-ups at 3 and 12 months accordingly: 1299 patients for the former, 2360 for the latter. These restricted datasets are the ones representing the basis upon which different prediction models have been trained and compared, as described in the following paragraphs.

Before doing so, we performed several univariate, bi-variate, and multi-variate analysis on data attributes and their correlation, with the foremost goals of *(i)* devising out good candidate predictors amongst the several attributes at our disposal, and *(ii)* check data distribution for prediction targets, to decide, for instance, whether down-sampling would be necessary to better represent imbalanced distributions of class labels[17]. When deemed worth of attention for the problem at hand, additional data pre-processing performed is described in the following paragraphs, along with description of the model and its evaluation.

Pre-processing included imputation of missing values (median value for numbers, and a random category for categorical variables, drawn probabilistically according to value counts of categories –so as to preserve relative percentages), One Hot encoding for categorical variables, and scaling of numerical ones to achieve normal distribution with mean 1 and standard deviation 0 (required by most learning algorithms described below). For all learning algorithms described below, train/test splitting with a 0.33 (test) ratio has been done, k-Fold cross validation with k=10, and Grid Search for hyper-parameters tuning (e.g. C regularization factor for SVC models, n-neighbours for kNN, max depth of trees for random forest, etc.).

The Python models exploited are Linear SVC, Radial Basis Function SVC, k-Nearest Neighbours (kNN), Random Forest. For each model, prediction of the following variables has been investigated:

- 'ADV_LAMA': binary classification problem, about whether to advise the patient to use Long-acting muscarinic antagonists (LAMA)

[17] Since a very similar analysis has been conducted for developing the Risk DSS, hence is described in deliverable D3.4 appendix, it is here omitted to avoid repetition.

- ‘ADV_ICS.LABA’: binary classification problem, about whether to advise the patient to use both inhaled corticosteroid (ICS) and long-acting β_2-agonist (LABA)
- ‘ADV_B2in’: binary classification problem, about whether to advise the patient to use β_2 bronchodilation
- ‘Work.diagn’: multiclass classification problem, about which diagnosis to attribute to the patient

For all the models tested, the confusion matrix where predictions are compared against true labels is reported in the following. For the models featuring probability distributions amongst classes (e.g. kNN and Random Forest), also the precision-recall curves and the ROC curves (with AUC) of each prediction class is shown.

It is worth emphasizing that formulating the problem of devising out which suggestion for intervention to deliver to the clinical professional while examining a patient as a classification problem instead of, for instance, an association rules mining instance, enables to fully exploit the UMCG dataset for *supervised learning*: as the dataset also contains information about the medications and advices given, it is possible to treat the problem as predicting the medication or advise variable based on all the other clinical information available.

5.3.1 Advise to use LAMA

For delivering suggestions about whether or not to advise usage of LAMA, a Nearest Neighbour classifier has been the best performing one. Neighbours-based classification is a type of instance-based learning or non-generalizing learning: it does not attempt to construct a general internal model, but simply stores instances of the training data. Classification is computed from a simple majority vote of the nearest neighbours of each point: a query point is assigned the data class which has the most representatives within the nearest neighbours of the point. The specific nearest neighbours classifier adopted is kNeighborsClassifier [18], that implements learning based on the k nearest neighbours of each query point, where k is an integer value specified by the user (hence, which has to be tuned, e.g. though Grid Search automatic procedure).

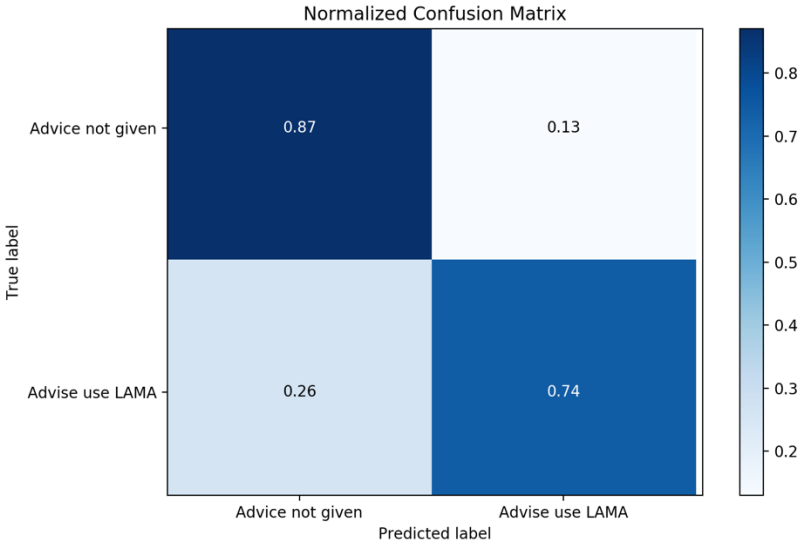

Figure 3 Confusion matrix for best model (kNN) delivering suggestion about LAMA usage.

Figure 3 shows the confusion matrix obtained by applying the kNN model on test data: performance is fairly good both for the positive and negative class, but a significant number of samples (patients) are incorrectly provided with suggestion to use LAMA when not necessary. The model uses weighted distance amongst samples to attribute classes, and Grid search auto-tuned "leaf_size" hyperparameter and number of neighbours to 10 and 4, respectively, while using weighted F1 score [19] as scoring metric. Performance is confirmed by ROC and precision-recall curves (in Figure 4 and Figure 5, respectively), although the latter also emphasises misclassification (advise usage when not due, green line).

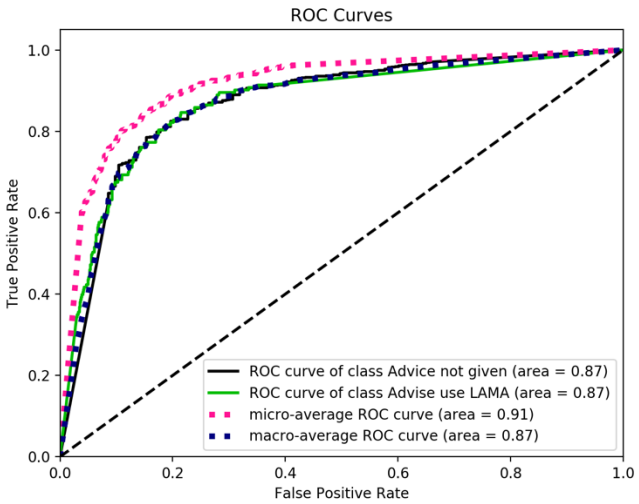

Figure 4 ROC curves for best model (kNN) delivering suggestion about LAMA usage.

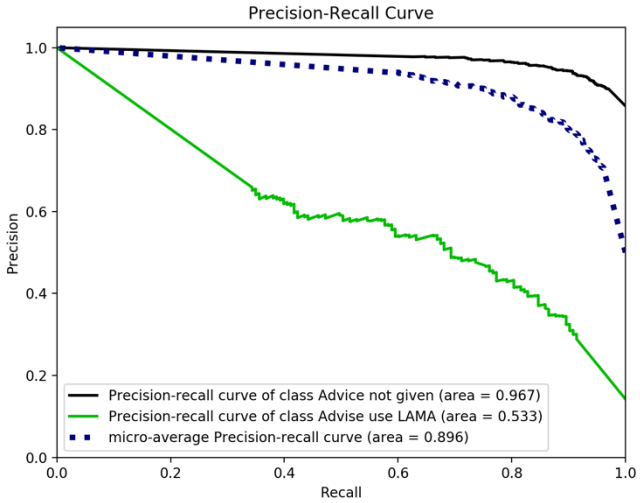

Figure 5 Precision-recall curves for best model (kNN) delivering suggestion about LAMA usage.

5.3.2 Advise to use both ICS and LABA

For delivering suggestions about whether or not to advise usage of both ICS and LABA, the best model has been a Random Forest [19] with balanced class weights, fully developed trees, and an auto-tuned (through Gird search) number of estimators (80) and minimum split samples (10% of population). The Random Forest method belongs to the family of "ensemble methods", whose main trait is to combine the predictions of several base estimators built with a given learning algorithm in order to improve generalizability / robustness over a single estimator. In random forests each tree in the ensemble is built from a sample drawn with replacement (i.e., a bootstrap sample) from the training set. Furthermore, when splitting each node during the construction of a tree, the best split is found either from all input features or a random subset of size max_features. The purpose of these two sources of randomness is to decrease the variance of the forest estimator. Indeed, individual decision trees typically exhibit high variance and tend to overfit.

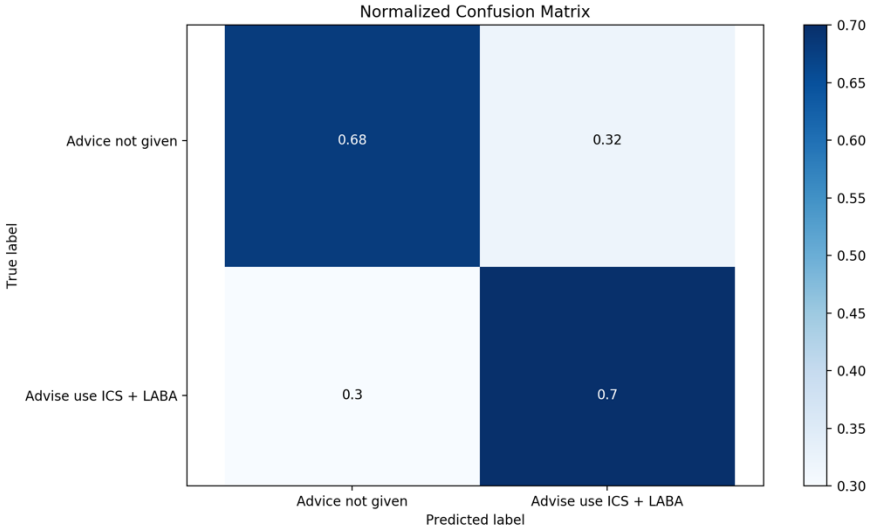

Figure 6 Confusion matrix for best model (Random Forest) delivering suggestion about ICS + LABA usage.

Figure 6 shows the confusion matrix obtained by applying the Random Forest model on test data, configured to produce fully developed trees, weight classes in a balanced way, and auto-tuned to optimal number of estimators and minimum samples to split nodes (90 and 2% of population, respectively). The model does a fair job in both delivering the advice when due and refraining to do so when unnecessary, as confirmed by the precision-recall curves reported in Figure 7 and the ROC curves in Figure 8 (both with corresponding AUC).

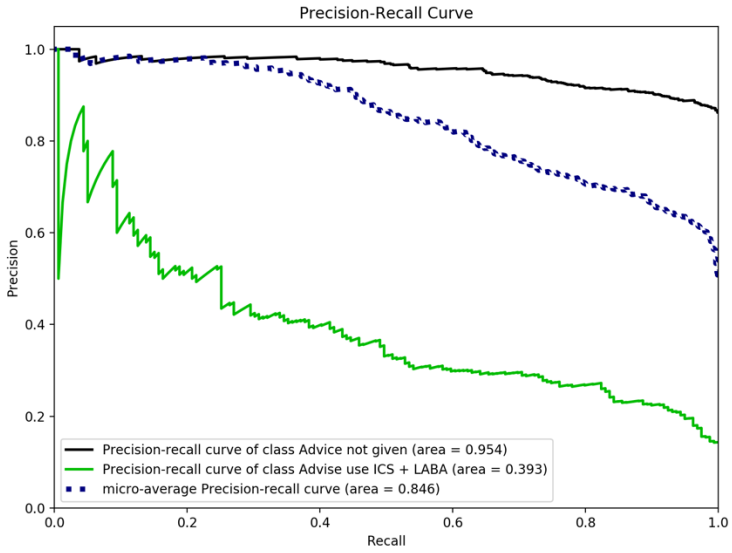

Figure 7 Precision-recall curves for best model (Random Forest) delivering suggestion about ICS + LABA usage.

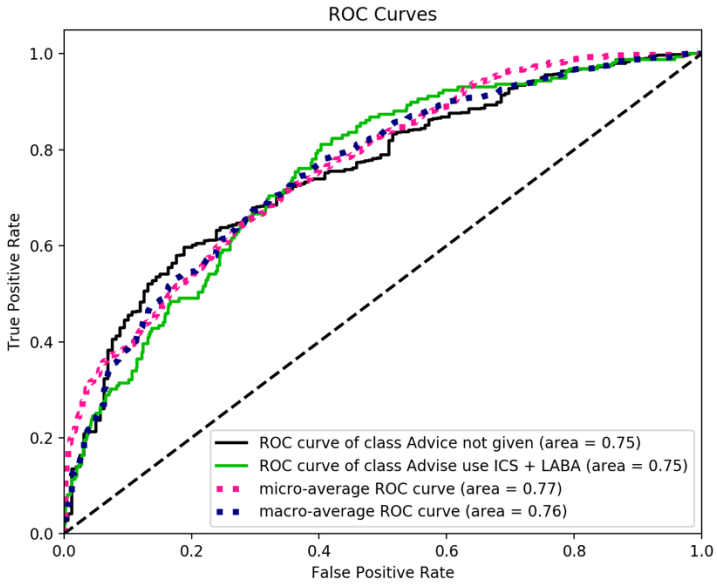

Figure 8 ROC curves for best model (Random Forest) delivering suggestion about ICS + LABA usage.

It is worth to remember that achieving good results across classes is complicated by the fact that class representation is usually imbalanced in the UMCG dataset (e.g. for ICS + LABA usage we have a ratio of 1:5) and that undersampling/oversampling are not deemed to be viable solutions to improve: the former would leave too few samples to train the model onto, whereas the latter would overfit the model to artificial data. We prefer to stick with fair results but actually reflecting what is possible to do with the actual UMCG dataset, with reflects the real-world situation of patients and hospitals (in reference to the amount and quality of data they have at their disposal).

5.3.3 Advise to use β_2 bronchodilation

For delivering suggestions about whether or not to use β2 bronchodilation no good model has been found. The two “best” performing ones are a Random Forest and a Linear SVC, whose confusion matrices are shown respectively in Figure 9 and Figure 11.

Support vector machines (SVMs) [20] are a set of supervised learning methods used for classification, regression and outliers detection. The advantages of SVMs are effectiveness in high dimensional spaces and versatility, as different Kernel functions can be specified for the decision function (e.g. Linear SVC is a SVM with a linear kernel). The main disadvantage is that SVMs do not directly provide probability estimates, but these are calculated using an expensive five-fold cross-validation (for Linear SVC they are not available at all).

The two mentioned models are complementary in what they are good at suggesting: the Random Forest is mostly right when deciding not to advise usage of β2 bronchodilation, whereas fails half the times when deciding to deliver the advice –as confirmed by the precision-recall curves shown in Figure 10.

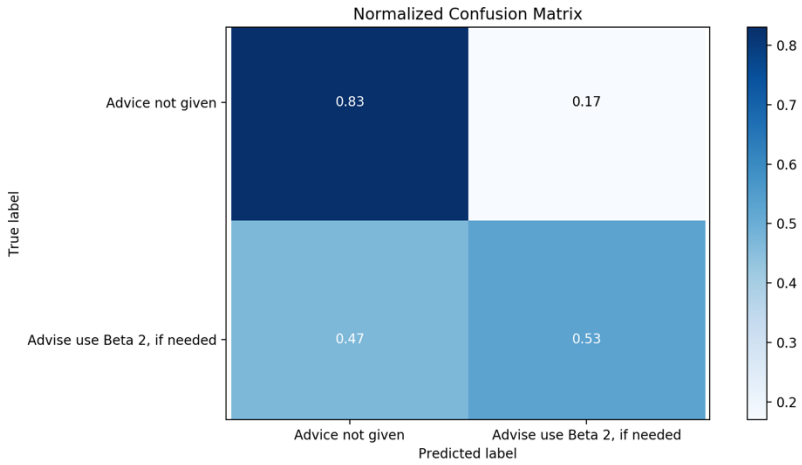

Figure 9 Confusion matrix for a Random Forest model delivering suggestions about β2 bronchodilation usage.

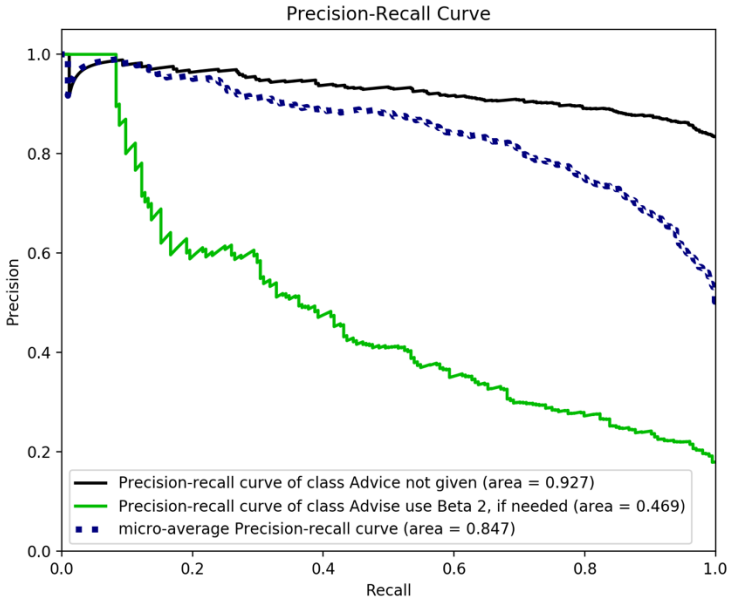

Figure 10 Precision-recall curves for a Random Forest model delivering suggestions about β2 bronchodilation usage.

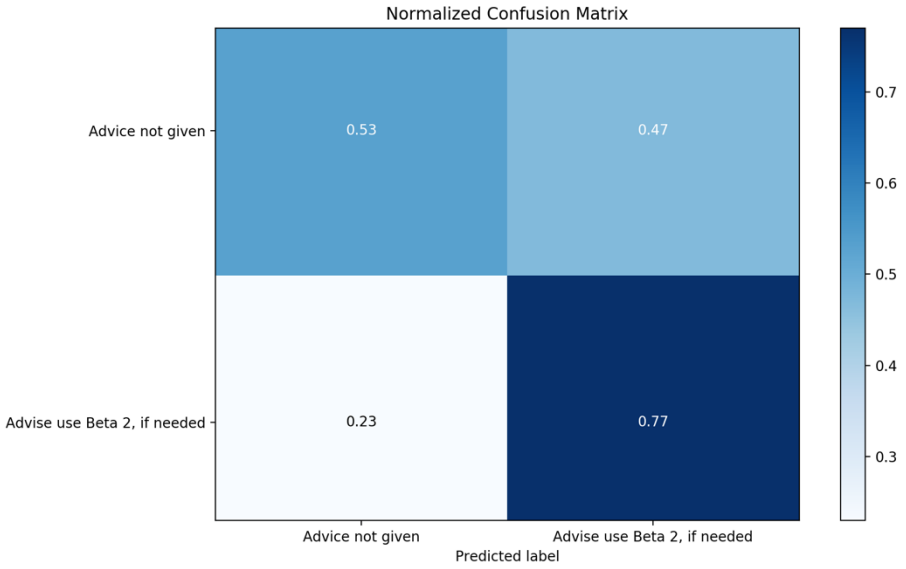

Figure 11 Confusion matrix for a Linear SVC model delivering suggestions about β2 bronchodilation usage.

The Linear SVC instead is mostly right when suggesting to advise usage, but fails half of the times when deciding not to do so (see Figure 11).

For these complementarity, a further attempt at improving classification results have been done with a Voting classifier, that is a kind of ensemble method (like the Random Forest) with a very simple idea at its core: combine different machine learning classifiers and use a majority vote (or the average predicted probabilities) to predict the class labels. Unfortunately, no improvement has been achieved.

5.3.4 Definition of diagnosis

Besides the standard pre-processing common to all pipelines here presented, as already described at the beginning of the section, additional filtering of features has been necessary. In particular, as we are interested in diagnosing COPD, asthma, asthma and COPD overlap, we had to remove from the training data all the features relative to such aspects, as they would erroneously inflate the results of the model---being essentially a “proxy” for the predicted variable. Accordingly, features 'Dich_asthma.1', 'Dich_COPD.1', and 'Dich_ACoverlap.1' have been removed, as they are direct indicators of whether the patient is suffering from COPD and/or asthma.

No model shown good results for this task. The two best performing ones are a Linear SVC model and a Random Forest, shown respectively in Figure 12 and Figure 13. The Linear SVC is only good in diagnosing COPD, whereas pretty bad in distinguishing COPD from asthma/COPD overlap. Extensive search through the hyper-parameters plane to find the best regularisation parameter (which helps especially in the case of unbalanced problems, as this one) did not help improve classification recall.

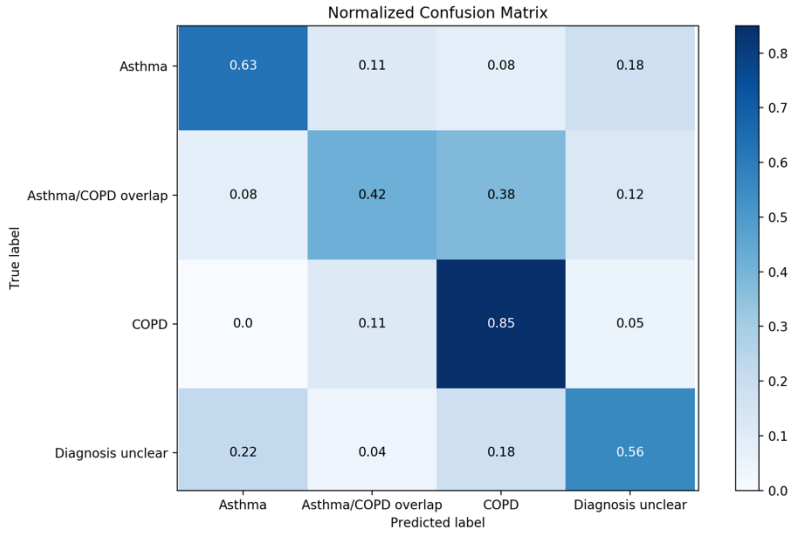

Figure 12 Confusion matrix for a Linear SVC model delivering suggestion about the diagnosis for the patient.

The Random Forest improves where the Linear SVC is already fairly good (asthma and COPD) but does not improve where the Linear SVC fails. This is particularly unlucky, because otherwise a Voting classifier may have helped, by combining the complementary classifications of different models.

We must emphasise that bad results are mostly due to the imbalance of the problem, where asthma/COPD overlap is less represented, likewise cases in which the diagnosis is unclear.

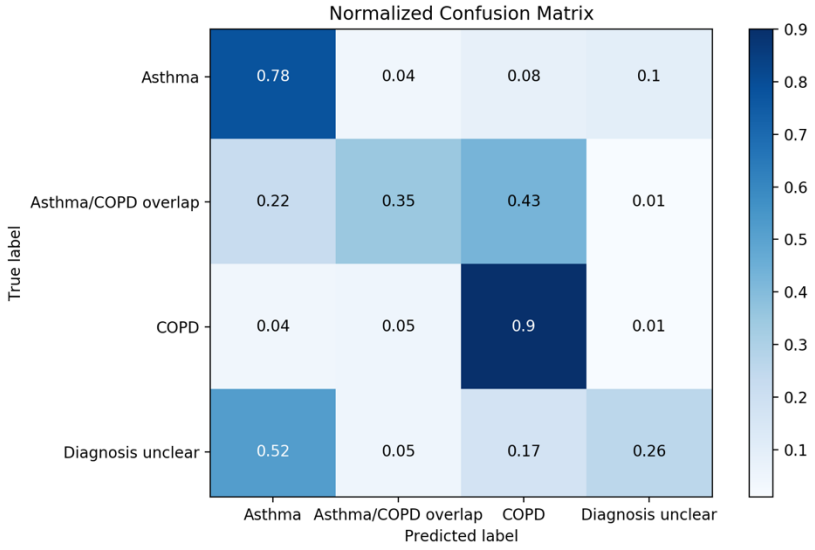

Figure 13 Confusion matrix for a Random Forest model delivering suggestion about the diagnosis for the patient.

5.3.5 Summary of models evaluation

Given the above analysis, we are satisfied with the CDSS performance as regards suggestion of interventions for asthma/COPD patients. Clinical partners highlighted how high accuracy of the models (>0.90 in confusion matrix main diagonal) is possibly extremely difficult to achieve as even for human professionals it is not so well-defined how to give the above advices and which diagnosis to make, hence the input dataset used for training has some “intrinsic error” that inevitably bias the ML models generated.

The results achieved are promising and encouraging to further improve models training, such as by attempting different approaches (e.g. neural networks and deep learning) or more pre-processing (e.g. clustering of similar patients before classification intra-cluster). We believe that our results confirm suitability of machine learning approaches even for application to dataset with modest size.

Clinical partners already expressed the desire to nurture collaboration beyond CONNECARE lifespan, possibly extending application of the approaches here describe to other clinical domains.

5.4 On integration

As already mentioned, integration of the Pathways CDSS with the SACM has not been done due to both the lack of data and the small overlap between UMCG dataset and data in the SACM.

Nevertheless, the whole Pathways CDSS has been designed by keeping in mind future integration, which may happen as depicted in Figure 14, where with respect to Figure 2 it is the SACM that feeds data for both training and application of models to the Pathways CDSS.

Also, we envisioned where the Pathways CDSS functionalities could be visible by clinicians in the SACM UI, hence designed mock-ups of what can be expected after integration.

In particular:

- Figure 15 showcases how drug-drug interaction could be reported by SACM in the workplan execution screens: the text to be displayed could be the one fetched by the DrugBank service (modulo suitable translation, if needed).
- Figure 16 shows how simple tasks suggestions could be reported in the same SACM pages devoted to the workplan. The text could be simply a fixed text corresponding to the prescription to deliver to the patient (suitably translated).
- Figure 17 shows how a suggestion for intervention (specifically, advise to use LAMA) could be delivered to the clinician. The text could be simply a fixed text corresponding to the variable predicted by the ML model currently in use.

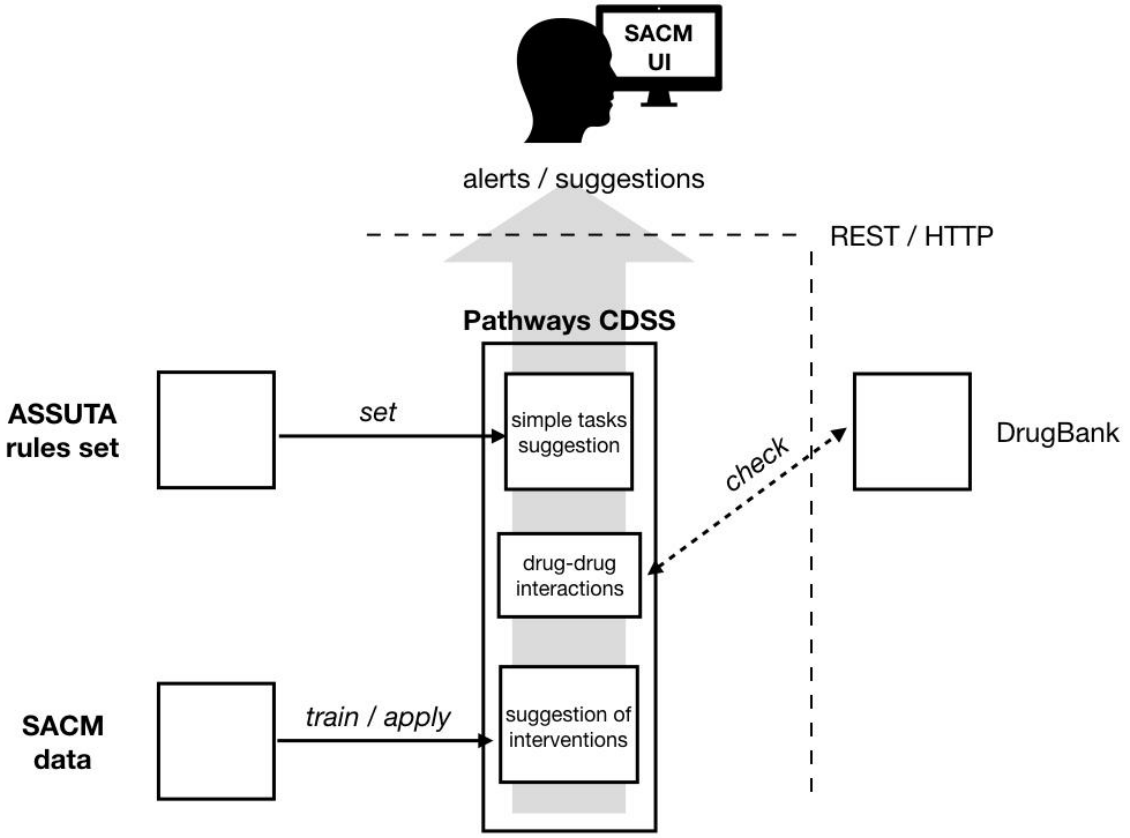

Figure 14 Pathways CDSS architecture for integration with SACM.

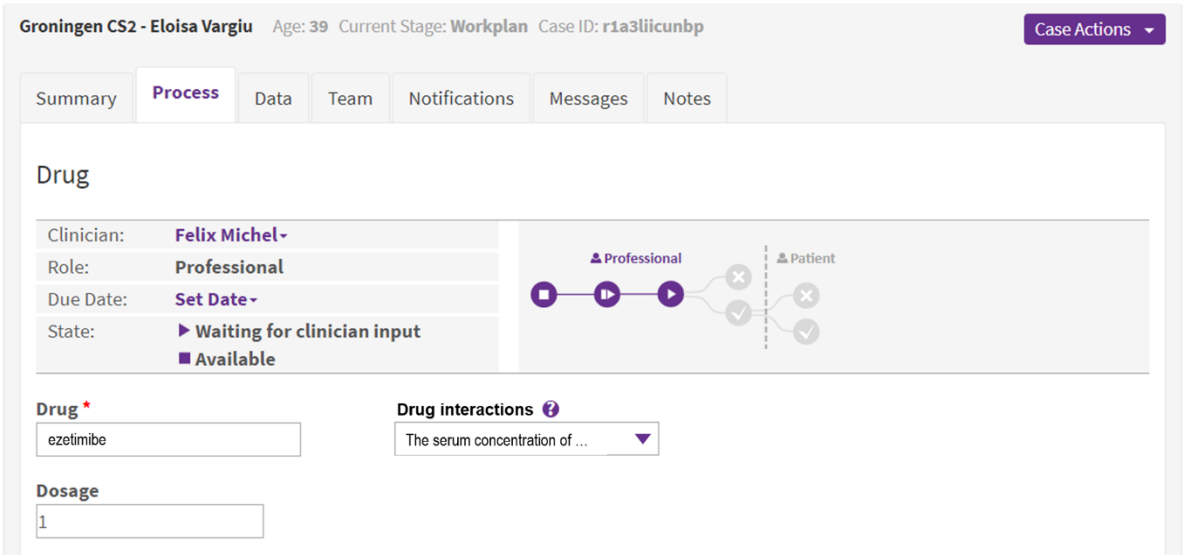

Figure 15 The drug-drug interaction alerts in the SACM.

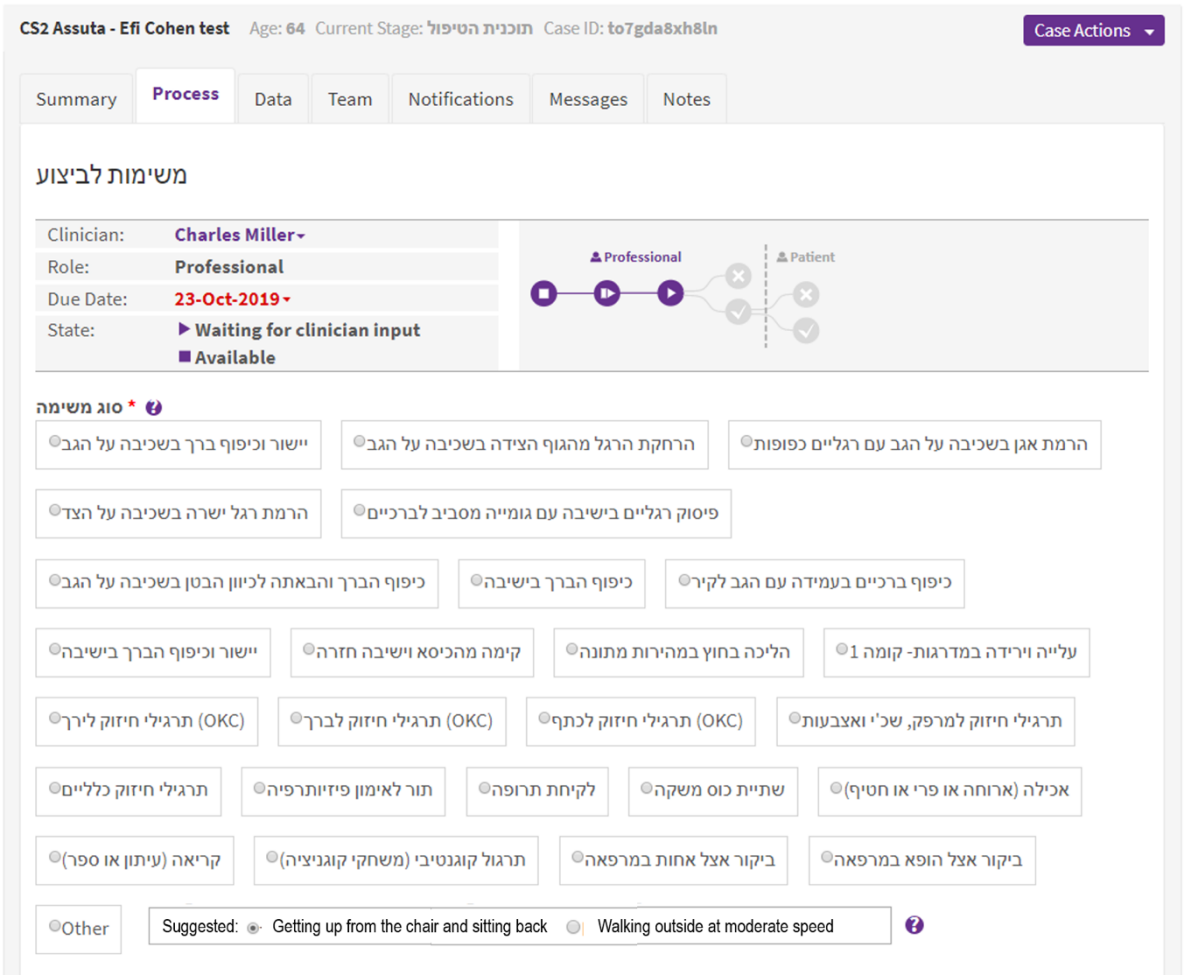

Figure 16 Simple tasks suggestion in the SACM (in English for the sake of clarity).

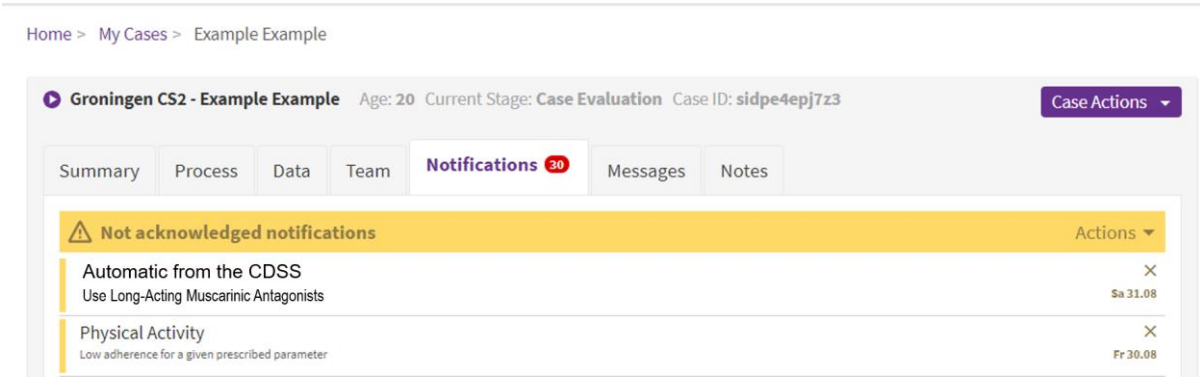

Figure 17 Suggestion for intervention in the SACM (in form of notification).

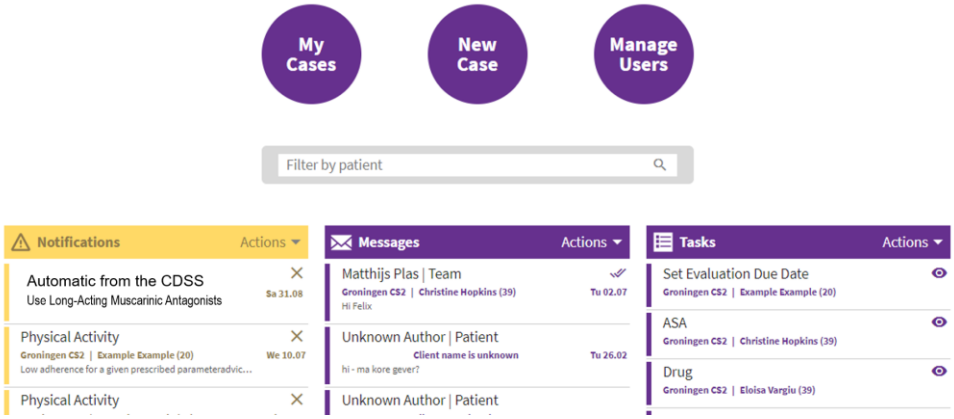

Figure 18 Suggestion for intervention in the SACM (in form of notification in the home page).

It is worth emphasising how the output of the Pathways CDSS are indeed suggestions, hence the clinician is always prompted for confirmation.

6. Outlook of future research and implementation

As the CONNECARE project is approaching the end of its lifetime, further development of the Pathways CDSS within the scope of the project is not feasible anymore. Nevertheless, all the clinical partners expressed a highly positive impression on the opportunities given by such a tool, hence would welcome further opportunities of collaboration with UNIMORE to put such DSS at work in a testing environment, for instance, selecting a pool of voluntary clinical professionals to which to display suggestions and providing feedback to the DSS itself by accepting them or not, explicitly (e.g. through a simple button). In particular, UNIMORE and UMCG already agreed to continue their collaboration beyond CONNECARE, to try to further improve the models generated, and IRBLL asked for opportunities to transfer the Pathways CDSS in their domain.

Besides trying to use other techniques, such as neural networks and deep learning, future efforts should be devoted to improve reproducibility and portability of the machine learning pipeline from data pre-processing to visualisation: as already said, the entire process is extremely time consuming, and without proper software engineering efforts has to be inevitably repeated when transferring the CDSS tool to another domain, even if slightly different (it sufficient to slightly change the input dataset or the variable to predict). Along this line, promising techniques are those already adopted, for instance, for the DSS for Risk assessment and stratification, in the “plugin mode” of operation (see D3.4) based on the PMML and PFA standards for models and machine learning pipelines representation.

7. Conclusion

In this document, the status of development of the Pathways CDSS has been described. Both a drug-drug interaction functionality, a rule engine for simple tasks suggestion, and set of ML models for suggestion of interventions have been implemented and validated in a lab setting, and are technically ready to be integrated into 3rd party services, such as the CONNECARE SACM.

Although the actual CONNECARE data generated by CONNECARE implementation studies will inevitably have differences, requiring further adaptations of the ML pipeline for data analysis and model training built for the UMCG dataset, the work carried out is valuable as it provided insights and experience about the issue of engineering a DSS for suggestion of personalized clinical workplans.

Results of the lab tests have been appreciated by the clinical partners, thus further exploitation plans have been outlined, as reported in D8.12 and D8.13.

8. Bibliography

[1] E. Strickland, "IBM Watson, heal thyself: How IBM overpromised and underdelivered on AI health care," *IEEE Spectrum,* vol. 56, no. 4, pp. 24-31, 2019.

[2] K. Bakera, E. Dunwoodie, R. G. Jones, A. Newshame, O. Johnson, C. P. Price, J. Wolstenholme, J. Leal, P. Mc Ginley, C. Twelves and G. Hall, "Process mining routinely collected electronic health records to define real-life clinical pathways during chemotherapy," *International Journal of Medical Informatics Volume,* vol. 103, pp. 32-41, 2017.

[3] Y. Ye, Z. Jiang, X. Diao, D. Yang and G. Du, "An ontology-based hierarchical semantic modeling approach to clinical pathway workflows," *Computers in Biology and Medicine,* vol. 39, no. 8, pp. 722-732, 2009.

[4] D. Alexandrou, F. Xenikoudakis and G. Mentzas, "SEMPATH: Semantic Adaptive and Personalized Clinical Pathways," in *International Conference on eHealth, Telemedicine, and Social Medicine (eTELEMED)*, Cancun, 2009.

[5] Y. K. A. Wen, "CONFlexFlow: Integrating Flexible clinical pathways into clinical decision support systems using context and rules," *Decision Support Systems,* vol. 5, no. 2, pp. 499-515 , 2013.

[6] H. Zhengxing, L. Xudong and D. Huilong, "Using Recommendation to Support Adaptive Clinical Pathways," *Journal of Medical Systems,* vol. 36, no. 3, p. 1849–1860, 2012.

[7] Y. Ziming, D. Zhao, L. Xudong, Y. Shengyuan, C. Xiaoyan and H. Duan, "A clinical decision support system for the diagnosis of probable migraine and probable tension-type headache based on case-based reasoning," *The Journal of Headache and Pain,* vol. 16, no. 29, 2015.

[8] S. El-Sappagh, M. Elmogy and A. Riad, "A fuzzy-ontology-oriented case-based reasoning framework for semantic diabetes diagnosis," *Artificial Intelligence in Medicine,* vol. 65, no. 3, pp. 179-208, 2015.

[9] V. Huser, L. V Rasmussen, R. Oberg and J. B Starren, "Implementation of workflow engine technology to deliver basic clinical decision support functionality," *BMC Medical Research Methodology,* vol. 11, no. 43, 2011.

[10] V. Kashyap, A. Morales and T. Hongsermeier, "On Implementing Clinical Decision Support: Achieving Scalability and Maintainability by Combining Business Rules and Ontologies," in *AMIA Annual Symposium Proceedings*, Washington, 2006.

[11] A. L. Beam and I. S. Kohane, "Big Data and Machine Learning in Health Care," *JAMA,* vol. 319, no. 13, pp. 1317-1318, 2018.

[12] B. Shickel, P. J. Tighe, A. Bihorac and P. Rashidi, "Deep EHR: A Survey of Recent Advances in Deep Learning Techniques for Electronic Health Record (EHR) Analysis," *IEEE Journal of Biomedical and Health Informatics,* vol. 22, no. 5, pp. 1589 - 1604, 2018.

[13] O. Faust, Y. Hagiwara, T. J. Hong, O. S. Lih and A. U Rajendra, "Deep learning for healthcare applications based on physiological signals: A review," *Computer Methods and Programs in Biomedicine,* vol. 161, pp. 1-13, 2018.

[14] F. Meherwar and P. Maruf, "Survey of Machine Learning Algorithms for Disease Diagnostic," *Journal of Intelligent Learning Systems and Applications,* vol. 9, pp. 1-16, 2017.

[15] S. Mariani, F. Zambonelli, A. Tenyi, I. Cano and J. Roca, "Risk Prediction as a Service: a DSS architecture promoting interoperability and collaboration," in *32th IEEE International Symposium on Computer-Based Medical Systems*, Cordoba, Spain, 2019.

[16] W. Meng, Z. Jiaheng, L. Jun, H. Wei, W. Sen, L. Xue and L. Wenqiang, "PDD Graph: Bridging Electronic Medical Records and Biomedical Knowledge Graphs via Entity Linking," Cornell University | arXiv.org, 2017.

[17] A. E.W. Johnson, T. J. Pollard, L. Shen, L.-w. H. Lehman, M. Feng, M. Ghassemi, B. Moody, P. Szolovits, L. A. Celi and R. G. Mark, "MIMIC-III, a freely accessible critical care database," *Scientific Data,* vol. 3, 2016.

[18] S. A. Dudani, "The Distance-Weighted k-Nearest-Neighbor Rule," *IEEE Transactions on Systems, Man, and Cybernetics,* Vols. SMC-6, no. 4, pp. 325-327, 1976.

[19] D. POWERS, "EVALUATION: FROM PRECISION, RECALL AND F-MEASURE TO ROC, INFORMEDNESS, MARKEDNESS & CORRELATION," *Journal of Machine Learning Technologies,* vol. 2, no. 1, pp. 37-63, 2011.

[20] L. Breiman, "Random Forests," *Machine Learning,* vol. 45, no. 1, pp. 5-32, 2001.

[21] C. Cortes and V. Vladimir, "Support-vector networks," *Machine Learning,* vol. 20, no. 3, pp. 273-297, 1995.

[22] M. U. Müller, "Playing Doctor with Watson: Medical Applications Expose Current Limits of AI," *DER SPIEGEL,* 3 August 2018.

9. List of Figures

Figure 1 The PDD Graph, connecting data mined from MIMIC-III to ICD9 diagnoses, and DrugBank prescriptions (image taken from https://kmap.xjtudlc.com/pdd/index.html#Introduction). 15

Figure 2 The proposed Pathways CDSS architecture. 17

Figure 3 Confusion matrix for best model (kNN) delivering suggestion about LAMA usage. 25

Figure 4 ROC curves for best model (kNN) delivering suggestion about LAMA usage. 25

Figure 5 Precision-recall curves for best model (kNN) delivering suggestion about LAMA usage. 26

Figure 6 Confusion matrix for best model (Random Forest) delivering suggestion about ICS + LABA usage. 27

Figure 7 Precision-recall curves for best model (Random Forest) delivering suggestion about ICS + LABA usage. 27

Figure 8 ROC curves for best model (Random Forest) delivering suggestion about ICS + LABA usage. 28

Figure 9 Confusion matrix for a Random Forest model delivering suggestions about b2 bronchodilation usage. 29

Figure 10 Precision-recall curves for a Random Forest model delivering suggestions about b2 bronchodilation usage. 29

Figure 11 Confusion matrix for a Linear SVC model delivering suggestions about b2 bronchodilation usage. 30

Figure 12 Confusion matrix for a Linear SVC model delivering suggestion about the diagnosis for the patient. 31

Figure 13 Confusion matrix for a Random Forest model delivering suggestion about the diagnosis for the patient. 31

Figure 14 Pathways CDSS architecture for integration with SACM. 33

Figure 15 The drug-drug interaction alerts in the SACM. 33

Figure 16 Simple tasks suggestion in the SACM (in English for the sake of clarity). 34

Figure 17 Suggestion for intervention in the SACM (in form of notification). 34

Figure 18 Suggestion for intervention in the SACM (in form of notification in the home page). 35