Model Driven Paediatric European Digital Repository

Call identifier: FP7-ICT-2011-9 - **Grant agreement no**: 600932

Thematic Priority: ICT - ICT-2011.5.2: Virtual Physiological Human

Deliverable 17.6

Test on Final Release Prototype of KDD and Simulation Platform

Due date of delivery: 31st May 2017

Actual submission date: 22/06/2017

Start of the project: 1st March 2013
Ending Date: 31st May 2017

Partner responsible for this deliverable: ATHENA

Version: 2.0

Dissemination Level: Public

Document Classification

Title	Test on the Final Release Prototype of KDD & Simulation Platform
Deliverable	17.6
Reporting Period	
Authors	ATHENA
Work Package	WP17
Security	Public
Nature	Report
Keyword(s)	

Document History

Name	Remark	Version	Date
MDP_D17.6_v0	Outline/templating	0.0	26/05/2017
MDP_D17.6_v1	First draft	1.0	01/06/2017
MDP_D17.6_v1.1	Draft	1.1	04/06/2017
MDP_D17.6_v1.2	Draft for internal review	1.2	08/06/2017
MDP_D17.6_v1.3	Draft after internal review	1.3	20/06/2017
MDP_D17.6_v2	Submitted version	2.0	22/06/2017

List of Contributors

Name	Affiliation
Orfeas Aidonopoulos	ATHENA
Harry Dimitropoulos	ATHENA
Omiros Metaxas	ATHENA

List of reviewers

Name	Affiliation
Henning Müller	HES-SO

Abbreviations

DCV	Data Curation and Validation (tool)
KDD	Knowledge Discovery and Data Mining / Knowledge Discovery in Databases
NND	Neurological and Neuromuscular Diseases

Table of Contents

1. Introduction

The purpose of this deliverable is to report on the activities of task T17.3 "Beta Prototype of KDD & Simulation Platform testing and validation", which is focused on ensuring the timely and efficient completion of the activities necessary to test and validate the Prototype of the KDD & Simulation Platform developed in WP16, and described in deliverable D16.3 [1]. It follows as an update of D17.5 [2] delivered last year.

The testing and validation of a service is of major importance when designing a medical informatics application to ensure its use in real conditions. An assessment should be performed along three dimensions: the usefulness of the system (i.e. to ensure it fits with the requirements of the end-users), its robustness (i.e. to ensure it will not entail negative consequences), and its facility of use (i.e. to ensure its acceptance and use). It is important to ensure that we carry an objective evaluation of the quality of the delivered applications by addressing the questions: (i) have we built the software application correctly? (verification) and (ii) have we built the right Infostructure? (validation), i.e. do the requirements satisfy the end users, and will they use the tools?

2. Testing the final prototype of the KDD platform

The developments of the final KDD platform (described in D16.3 [1]) – like with previous releases of the platform – were driven by trying to follow the proposed *Scrum agile process*, with a view on user validation and interfaces quality improvement (described in D17.1 [3]). Within this process, user representatives are consulted on the products features, improvements and bug fixes to be dealt with. Testing and coding are done incrementally and iteratively, building up each feature until it provides enough value to release to production.

Although the standard Scrum methodology (in the strict sense) failed to apply, for the reasons explained in D17.2 [4], the methodology used was in line with the main agile concepts, but with less formalism. Particularly for the DCV tool and the KDD platform, a number of *development & testing* iterations took place (similar to agile *sprints*) in close communication with a number of end users (clinicians, data analysts, researchers, etc.). A quick production process was adopted to make the functionalities usable (and so testable) as soon as possible.

During the last 15 months, we have mainly tested the new web-based KDD platform which was fully integrated with the Data Curation and Validation (DCV) tool. Due to the redesign of the whole DCV interface in order to support the KDD platform, all integrated algorithms were tested again on synthetic and real data. More specifically, changes were applied on the user-interface for facilitating the training of the KDD models. Rows can now be deleted and no re-initialization of the uploaded CSV is needed any longer. Missing values can also be imputed automatically (by the median to avoid outlier effects). Similarity search was also enriched with new parameter selection options, and posterior probability tables were added after a classification analysis. The cross-validation technique was also optimized to separate training and test sets based on the patient cohorts. After the platform was tested internally by the infrastructure team, we proceeded with validation steps in cooperation with clinicians. They started to use the platform during the 4th bi-annual MD-Paedigree meeting (Leuven, mid-September 2016) applying real data from clinical cases. The clinicians managed to load, curate and process their data successfully on the platform.

Prediction models were also developed by the NND group in order to compare with their own manual classification system, as reported in detail in D16.3 [1].

As we have already described in previous deliverables, the platform uses extended SQLite queries with operators written in python-2.7 ("madIS" worker) to handle the connection between the data stored in the user's database and the interface. For instance, when a model is trained, the operator ***sktrain*** is included on the query that selects the desired data which are to be fitted into this model. Particularly, the madIS operator *sktrain* is used to train a model. The following query actually implements the supervised algorithm initialized by *initstr* and fits the data provided by table *t*. The response variable is declared as a parameter (*classname*) inside the query. When a user enters the cross-validation parameter, it is automatically added as a parameter of the operator; otherwise a default 5-fold cross-validation is performed:

sktrain filename:MyModelFile initstr: DecisionTree() cv:10 classname:2 select "0","1","2","3",…"N" from table t;

Where "*0*" to "*N*" are the column IDs of table *t*. All the columns in basic query (*select COL_IDs from table t*) are used as predictor variables except for the *id* which is declared in *classname* attribute (in this example response variable is column 2).

The main method for validating the prediction models is the well-known cross-validation. Since now, validation has been held on simple k-folds where the user selects an integer for k and the operator and random splits into k-folds are performed. In the latter months, the operator for training predictive models was optimized in order to test and validate the models more efficiently and in many cases the model's accuracy increased.

An additional *grouping* parameter can now be used in the operator, ensuring that each patient cohort is never separated into different sets. For instance, in Figure 1, this NND sample dataset consists of 84 samples corresponding to 35 patients. This means that there are medical data collected from multiple patients, with multiple samples taken from each patient. For instance, each patient may have more than one trial or visit in the hospital and so his/her measurements amongst the visits may be highly correlated. Applying simple k-fold cross-validation, records referring to the same patient should be in the same set (train or test) due to the possible high correlations. Otherwise, we risk using a patient's visit for training and another visit of the same patient for validation. Essentially, we would like to know if a model trained on a particular set of groups generalizes well to the unseen groups. With this approach, a patient cohort is never separated into different sets.

Therefore, the query with the addition of the *grouping* parameter is reformed like this:

sktrain filename:MyModelFile initstr: DecisionTree() cv:10 classname:2 grouping:0 select "0","1","2","3",…"N" from table t;

Now the column with ID 0 is not included either in the set of predictors nor for the training of the models. It is just used as a grouping variable to ensure that all the samples in the validation fold come from groups that are not represented at all in the training fold.

The default cross-validation technique was also optimized. In case a user does not select a grouping variable, a **Stratified** k-fold cross validation is applied. Stratification of the folds efficiently tackles the

problem when a large imbalance in the distribution of the target classes exists. For instance, there could be several times more negative samples than positive samples. In such cases, stratified sampling is used to ensure that relative class frequencies are approximately preserved in each train and validation fold.

+

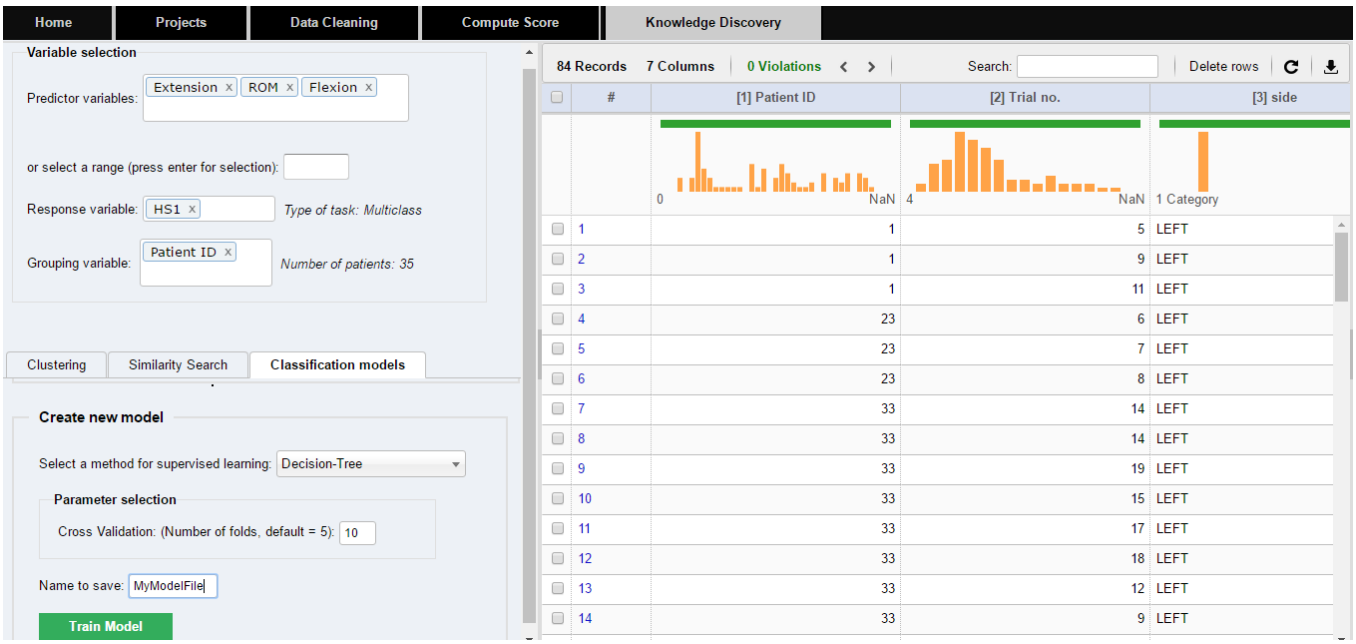

Figure 1: A user is now able to select a grouping variable to ensure that all the samples in the validation fold come from groups that are not represented at all in the training fold.

2.1. Evaluation cycles

Similar to the process described in D17.5 (for the beta version of the KDD platform) [2], for the final version of the platform, a number of *development & testing* iterations/sprint cycles took place. At the end of each cycle, a version of the platform was released, which was first tested internally by our development/research team. Any bugs/issues are logged and a new development and evaluation cycle begins.

2.1.1. By a group of University students and researchers

Once a version was stable enough, we extended the evaluation phase of the cycle by giving the tool and the script to researchers/students that volunteered for testing as "users" (a minimum of 3 users for simple changes; a maximum of 8 users for substantial updates). These users were all from the Informatics Department of the University of Athens (with which ATHENA collaborates on other projects) and used it during the same session. The reason for this was that we wanted to also test how the tool (being a web-based tool) responded to multiple requests/actions at the same time, as well as, how it handled multiple users.

This group of users/testers was monitored during the use of the platform and any comment, complaint, suggestion, bug, etc. was noted during the evaluation session. These sessions proved to be very useful for finding problems that we had not encountered, as well as for getting ideas that we had not thought off as developers, either relating to new functionalities or user-interface improvements, such as the following (already implemented):

- adding a search functionality, so a user can find a specific column quickly
- suggestions for making much simpler and easier the way which workflows are managed
- suggestions for improved usage of the history (undo/redo) functionality
- ideas for the redesign of the KDD interface (look and feel)
- ideas for improving the model training reports (graphs, visualisations, etc.)

Some ideas given by testers/users were not implemented, as we needed to prioritise. For example, the ability to run the tool on a mobile device (can be done, e.g., on an iPad, but not optimised to do so). Other suggestions have been seriously considered but require substantial research and effort. However, we are planning to continue the development of the platform after the end of the project, and some of the directions we are moving towards are described in one of our recent papers [5].

2.1.1. By clinicians

Once the above-mentioned evaluation with a group of University students is completed, a further development cycle corrects the most serious problems/bugs and also implements the best suggestions. We then expand the testing/evaluation phase even further by having the tool used by clinicians, i.e. the end-users of the platform within MD-Paedigree.

These evaluations took place either in person (e.g. during the clinicians' hands-on training at the 4th biannual MD-Paedigree meeting in Leuven, Belgium) or via *Skype/TeamViewer* videoconferencing. These sessions helped us assess the tool both qualitatively (i.e. for ergonomics, comprehensiveness of information, etc.) and quantitatively (i.e. for effectiveness, precision, etc.). These sessions provide the most valuable feedback, as it comes from the end-users of the tool within the project.

Repeating these development and evaluation cycles a number of times, led to the release of the final version of the platform.

2.2. Clinicians' hands-on training

At the 4th biannual MD-Paedigree meeting in Leuven, Belgium (12-13 September 2016), training of clinicians on the DCV module of the /KDD platform was completed very successfully. Clinicians managed to handle and run curation analysis in less than twenty minutes, while in the previous hands-on training meeting (in Rome, in February 2016), the training session took at least one hour for DCV [2]. The clinicians once more confirmed the tool's ease of use and its user-friendly interface. They were also very pleased that is was integrated with the knowledge discovery platform and that would now be able to design an end-to-end pipeline: from curation and pre-processing, to development of predictive models.

Especially for the knowledge discovery extension, a number of teleconferences with clinicians via Skype followed for more detailed training. Every time we updated a version of the platform, the training was repeated introducing the new features to the clinicians. During the final months of the projects, this cycle was repeated even more frequently, sometimes once every week or two, as we were optimising the platform's features based on the clinical needs and the specific use-cases (described in detail in D16.3 [1]). In particular, clinicians were very specific in requesting what should be included in the comprehensive report that is produced after each model's training (see section 2.1.3 of D16.3 [1]).

3. Conclusion

In this report, we present the WP17 activities regarding testing and validation of the Final Prototype of the KDD & Simulation Platform developed under WP16. A number of *development & testing* iterations took place (similar to agile *sprints*) in close communication with a number of end users (clinicians, data analysts, researchers, etc.). A quick production process was adopted to make the functionalities usable (and so testable) as soon as possible. This led to major improvements and the release of the beta prototype of the KDD platform. The near-final prototype was presented and used by clinicians during the training session in Leuven in September 2016 and thereafter via Skype video-conferencing until the release of the final prototype. The great majority of clinicians judged the platform as very friendly, robust, very useful in detecting erroneous data and in helping them decide how to correct their data. Regarding KDD usage, prediction models were also developed by the NND group in order to successfully compare with their own manual classification system, validating the use of the KDD tool for this use-case, as reported in detail in D16.3 [1].

4. References

[1] MD-Paedigree deliverable D16.3, “Final Prototype of KDD & Simulation Platform”, June 2017.
[2] MD-Paedigree deliverable D17.5 “Test on Beta Prototype of KDD and Simulation Platform”, March 2016.
[3] MD-Paedigree deliverable D17.1 “Test Report on MD-Paedigree Alpha Prototype”, March 2015.
[4] MD-Paedigree deliverable D17.2 “Test Report on MD-Paedigree Beta Prototype”, March 2016.
[5] A.Gogolou, M. Kyriakidi, and Y. Ioannidis, “Data Exploration: A Roll Call of All User-data Interaction Functionality”, Proc. of the Third Int'l Workshop on Exploratory Search in Databases and the Web, ExploreDB '16, San Francisco, California, US, 2016, pp.31-33.