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 17.5

Test on Beta Prototype of KDD and Simulation Platform

Due date of delivery: 29th February 2016

Actual submission date: 16th March 2016

Start of the project: 1st March 2013
Ending Date: 28th February 2017

Partner responsible for this deliverable: ATHENA

Version: 1.0

Dissemination Level: Public

Document Classification

Title	Test on Beta Prototype of KDD & Simulation Platform
Deliverable	17.5
Reporting Period	
Authors	ATHENA
Work Package	WP17
Security	Public
Nature	Report
Keyword(s)	Testing KDD & Simulation Platform Beta Prototype

Document History

Name	Remark	Version	Date
MDP_D17.5_v0.1	Outline/templating	0.1	29/01/2016
MDP_D17.5_v0.2	First draft	0.2	03/03/2016
MDP_D17.5_v0.3	Second draft	0.3	10/03/2016
MDP_D17.5_v0.4	Reviewed version	0.4	14/03/2016
MDP_D17.5_v1	Final version	1.0	16/03/2016

List of Contributors

Name	Affiliation
Orfeas Aidonopoulos	ATHENA
Harry Dimitropoulos	ATHENA
Anna Gogolou	ATHENA
Omiros Metaxas	ATHENA
Mona Alimohammadi	UCL
Vanessa Diaz	UCL

List of reviewers

Name	Affiliation
Henning Müller	HES-SO
Bruno Dallapiccola	OPBG

Abbreviations

BMI	Body Mass Index
CP	Cerebral Palsy
eCRF	(Electronic) Case Report Form
CFD	Conditional Functional Dependency (rules)
CR	Constraint Rules
CSV	Comma-Separated Values (file format)
CV	Cross-Validation
DCV	Data Curation and Validation (tool)
FD	Functional Dependency (rules)
GUI	Graphical User Interface
KDD	Knowledge Discovery and Data Mining / Knowledge Discovery in Databases
ML	Machine Learning
NaN	Not a Number
NB	Naive Bayes (algorithm)
PCDR	Paediatric Cardiology Digital Repository
RF	Random Forest
UDFs	User Defined Functions

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 Beta Prototype of the KDD & Simulation Platform developed in WP16, and described in deliverable D16.2 “Beta Prototype of KDD & Simulation Platform”.

In particular, in this report we present the testing and validation of (1) the beta version update of the Data Curation and Validation (DCV) tool, and of (2) the further developments delivered on the new web-based KDD platform (AITION KDD).

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 beta update of the Data Curation and Validation (DCV) tool

In this section, we present the testing and validation of the beta update of the Data Curation and Validation (DCV) tool, which is a component of the AITION KDD platform. The Data Curation and Validation (DCV) tool is an exploration tool that offers an advanced (semi)-automatic data cleaning. DCV facilitates the detection of numeric outliers, missing values as well as alphanumeric typographical and logic errors. The computation of new derived columns is also possible. A history record is kept for each of the above actions. The user can undo/redo history and run his data cleaning script on other/additional data (for full description of all functionalities, please see deliverable D15.2 for the alpha release and D16.2 for the beta release).

As described in D16.2 "Beta Prototype of KDD & Simulation Platform", during the past year – i.e. since February 2015 and submission of deliverable D15.2 "DCV Curation Tools and Services to Automatically and Manually Acquire High Quality Curated Data" – important work has been carried out in terms of visual data exploration, automatic error detection and collection of data cleaning rules, as well as data handling, that accompany a data cleaning process.

These developments 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 "Test Report on MD-Paedigree Alpha Prototype"). 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 "Test Report on MD-Paedigree Beta Prototype", the methodology used was in line with the main agile concepts, but with less formalism. Particularly for DCV, 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.

To facilitate this process, we decided to focus on a specific use case that was also later used as the basis for the script of the training session in Rome, in mid February 2016.

2.1. Training script/use case for the DCV tool

The use case described in this section was based on the eCRF questionnaire data collected for *Cardiovascular Disease Risk in Obese Children and Adolescents*. Please note that DCV can be used with *any* kind of numeric and alphanumeric data. In other words, a similar "script" can be applied on data from other disease areas, or even for datasets that are not related to diseases or medical problems.

Name. Detecting and correcting errors in the data.
Brief description. The user, driven by the tool, can create his own data cleaning script and find errors in the data.
Actors. A physician / data curator / data miner.
Preconditions. The physician is allowed access to PCDR/MD-Paedigree Research Portal and data files.
Post Conditions. The physician can continue with the knowledge discovery process of his cleaned/curated dataset.

Basic Flow.

1. The physician uploads the training dataset;
2. The system loads the data on a data table. Above each column the distribution of its data is displayed together with a stacked bar chart divided into three subparts whose lengths are in proportion to the number of errors, missing and valid values of the column;
3. The physician selects a subpart of the stacked bar chart of a column;
4. The system filters the data of the column at hand based on the selected area of the bar chart and displays the results on the data table;
5. The physician selects from the submenu of a column the option “Detect misspellings”;
6. The system returns in groups similar misspelling errors of the column values;
7. The physician selects the groups he wants to merge and the corresponding merged values (that is the highest frequency value in the group);
8. The system updates the column values with their corresponding merged values and displays the results on the data table;
9. The physician creates a data-cleaning rule defined by relational and logical operators among the columns. Then, he runs the rule;
10. The system detects and displays the errors that violate the rule;
11. The physician corrects the errors based on the visualization the system provides;
12. The physician writes a formula for the computation of a medical score, e.g. the formula for the Body Mass Index (BMI) calculation;
13. The system computes the score and displays the result on a new derived column;
14. The physician can undo/redo history;
15. The physician extracts the above workflow and runs it in another project with other/additional data.

Requirements addressed:

- Ability to find outliers
- Ability to find missing values
- Ability to find alphanumeric typographical errors
- Ability to find logic errors
- Ability to derive new columns
- Ability to visualise the data
- Ability to undo/redo history
- Ability to run workflow

Users targeted:

- Clinicians, Data Curators, Data Miners

Data:

- Any dataset with numeric and alphanumeric data

Limitations:

- The tool currently supports the uploading of data only in CSV format.

Suggested questions for training:

- Is the tool useful and easy-to-use for the detection of erroneous data?
- Is the tool helpful and does it contribute to your decision for the right correction of your data?
- Do you have any problems with your data that the tool cannot presently address and would you like any such functionality to be added to the tool?
- How do you judge the overall effectiveness and efficiency of the tool?

2.2. Evaluation cycles

Based on the above training script/use case, a number of *development & testing* iterations/sprint cycles took place. At the end of each cycle, a version of DCV was released, which was first tested internally by our development/research team. Any bugs/issues were logged and a new development and evaluation cycle begins.

2.2.1. 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 5-10 researchers/students that volunteered for testing as "users". These users were all from the Informatics Department of the University of Athens (with which ATHENA collaborates on other projects) and would use DCV 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 its use of the tool 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 getting ideas that we had not thought off as developers.

2.2.1. By clinicians

Once the above evaluation with a group of University students was 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 DCV within MD-Paedigree.

These evaluations take place either in person (e.g. during the third biannual meeting held in Chania, Crete, in October 2015) or via *Skype/TeamViewer* videoconferencing. These sessions help us assess DCV 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 came from the end-users of the tool within the project. For the majority of these evaluations, the clinicians involved were Dr. Alex Jones and Jakob Hauser from UCL, since the use case was based on their data: the eCRF questionnaire data collected for *Cardiovascular Disease Risk in Obese Children and Adolescents*.

However, the tool was also presented to all attending physicians during the Crete meeting and the Annual Internal Meeting in Rome, in February 2016, which included a more formal training and evaluation session.

Repeating these development and evaluation cycles a number of times, led to the release of the beta version of DCV, as presented in Rome. A brief report of the feedback received during the Rome training session follows. Since the Rome training, the tool has been available online for clinicians to use, and it is currently being evaluated by them on their own. We expect that this will bring more feedback and recommendations, based on which the next stable version of the tool will be released.

2.3. Feedback from the Rome training session

The beta version of DCV was evaluated during the third annual internal review meeting held in OPBG, Rome (February 2016). A demonstration of the tool was conducted and the physicians of MD-Paedigree provided their comments and recommendations via questionnaires that were handed out at the end of the session (see the Appendix). In addition, many voiced their comments during the actual training session (the training was conducted by Mona Alimohammadi, UCL; Anna Gogolou, Orfeas Aidonopoulos and Harry Dimitropoulos from ATHENA were present for assisting clinicians during the session and noting any feedback or questions that arose).

Before the training sessions started, printed colour booklets of the training script were handed out. These provided step-by-step instructions with screenshots, to help clinicians follow along during the training, but also take with them as visual manuals and a reminder of how to use the tools presented. The training sessions were successful, the attendance was very good (room almost full) and the clinicians were engaged throughout the one-and-a-half-hour process. Next follows a summary of the feedback received via the completed questionnaires (8 physicians responded).

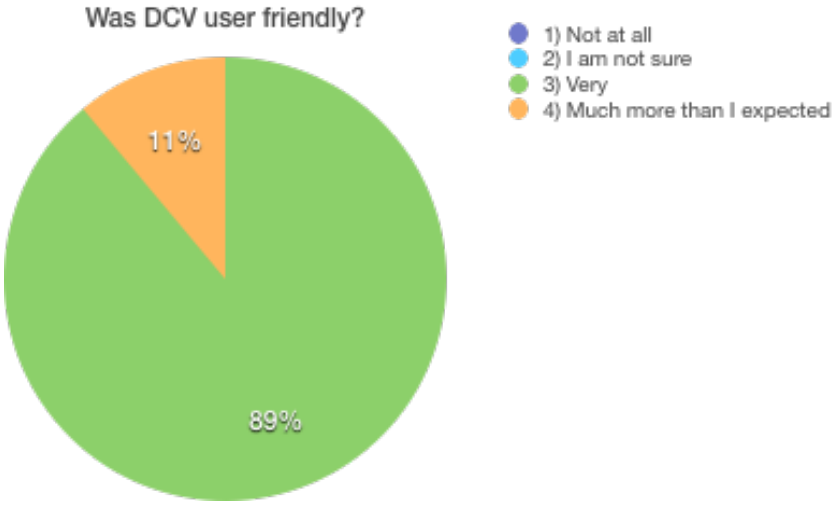

Figure 1: DCV user-friendliness: the majority of clinicians responded the tool was "very" user-friendly; none reported "not at all" or "I am not sure".

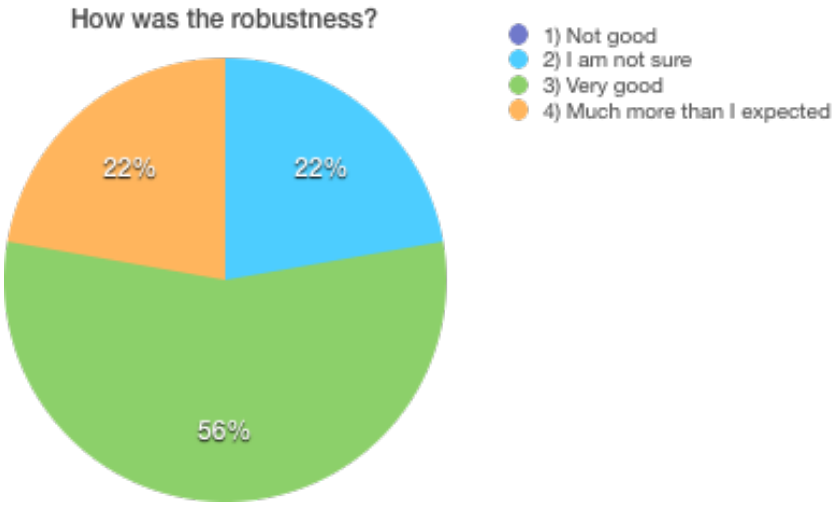

Figure 2: DCV robustness: over half of clinicians responded "very good"; the remaining ones were split between "not sure" and "much more than expected".

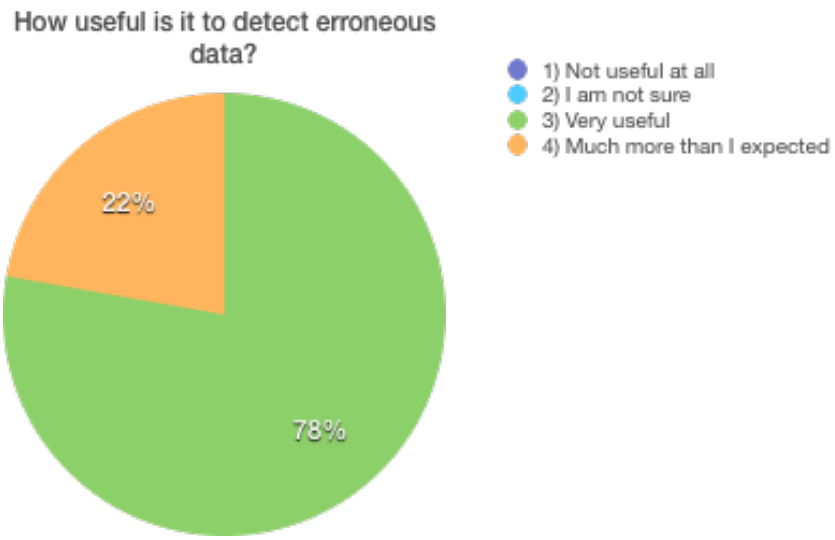

Figure 3: DCV error detection: the majority of clinicians responded the tool was "very useful"; none reported "not useful at all" or "I am not sure".

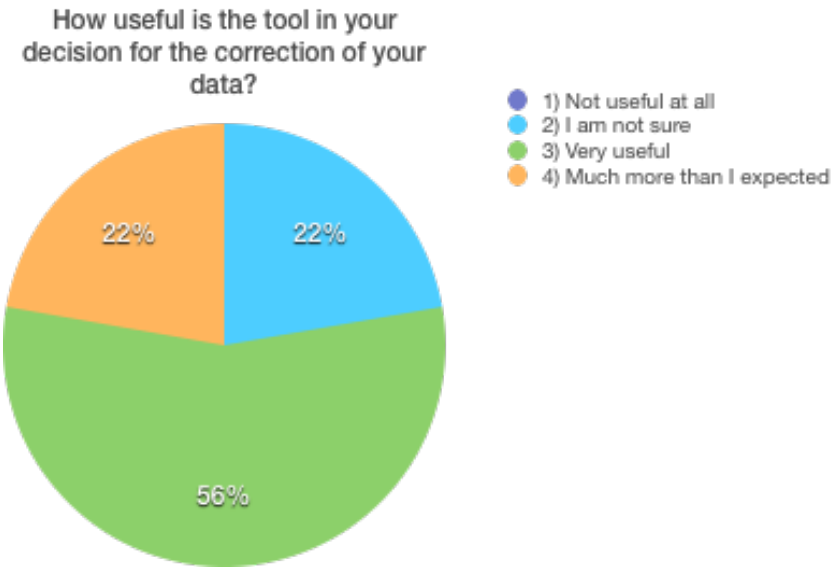

Figure 4: DCV usefulness: over half of clinicians responded "very good"; the remaining ones were split between "not sure" and "much more than expected".

The above pie charts (Figure 1 to Figure 4) show the results of the first four multiple-choice questions of the questionnaire. Overall, the great majority of the clinicians judged the tool as very friendly, robust, very useful in detecting erroneous data, and very useful in helping them decide how to correct their data.

The second set of four questions on the questionnaire allowed the clinicians to respond by writing their comments (free text). A brief summary of the answers follows.

Q5) Do you have any problems with your data that the tool cannot presently address and you would like for that functionality to be added?

Answers ranged from "No" (1 case), to "N/A" or no response (2 cases), but the majority responded with something along the lines of "Not sure/Probably/Maybe, in the future after using the tool more" (4 cases), which is a very reasonable answer. There was also one clinician that made a very specific request about handling repeated measurements: in their datasets they have patients repeated in multiple columns (e.g. follow-up visits) and they would like to be able to create rules that check if values on one row correspond

to/are greater than/etc. values in another row of the same patient. In other words, they would like to use DCV's *Conditional Functional Dependency* (CFD) *rules* functionality not only between two columns and their data, but also between specific row values. This is something that we will consider implementing for the final release of the tool.

Q6) How do you judge the overall effectiveness and efficiency of the tool?
Answers ranged from "I think it will be very useful/seems very effective/pretty effective but need to use the tool more" (3 cases), to more certain answers such as "very good/very efficient" (4 cases), all the way up to "excellent" (1 case).

Q7) Which information are you missing?
The majority answered "None", "N/A", "Don't know", or left a blank answer or a dashed line, indicating nothing missing. One user responded that it was not clear if anything was missing, as they would need to first use the tool for a concrete goal. Another user would like to be able to use a function for calculating statistical analysis p-value. Finally, one user was not sure if one could write several functions on their own for their data, which is already implemented in DCV.

Q8) How do you think we could improve this tool?
The majority of the suggested improvements and added functionalities were requested during the training session (discussed further below), so a number of clinicians responded with "as discussed" or "suggestions already noted during the tutorial"/"suggestions made during training". A couple of responders left the question unanswered or used a dashed line indicating no suggestions.

One user found that if s/he wrote a formula for a derived column that had a mistake, DCV would keep display "Loading…" and get stuck; s/he had to out to the home page and do it all over again. This is a know bug and will be taken care of in the next release. The same user also encountered a problem with scrolling after entering a couple of rules. Again, this appears to be an issue when the tool is used with Mac computers. Scrolling is actually possible, but it is less obvious that with other operating systems.

Another user suggested that a different colour palette should be used in the graphs, something easy to fix but we will need to experiment with in order to find something that satisfies most users.

Finally, one user also stated "very promising tool! Well done". In general, DCV received very good reviews/testimonials and a couple of attendees expressed the hope for further development/proper exploitation of the tool, after the project ends.

Of the verbal feedback/comments received during the training session – which were all logged and will be prioritised and evaluated for implementation during the next development iteration – a few are noted below as examples:

1. The users requested to be able to use the tools on their iOS devices (iPad, iPhone, etc.). DCV can already be used on a Mac, apart from the scrolling difficulty mentioned earlier. Since this is a web-application, it can also be accessed via an iPad, but it is not optimized for such use, so this is something to consider.
2. The colours of the pie chart were an issue; a different pallet of colours would be preferred. This was also addressed in writing, as mentioned above.
3. There were a few questions regarding some of the functionalities of DCV. These were not bugs or errors, or even suggestions, but were actually requests for a more detailed explanation (due to the

time constraints of the training sessions, we could not go into depth for some of these questions), e.g. “Please explain more on when can the pie chart be used? What will I use it for?” One possible answer is that Pie charts are used in DCV to visualize data and they are also very useful for interactively filtering the data by clicking on a specific value/slice of the pie.

4. Similarly to #3 above, another question for a more detailed explanation was “What is the one-to-one relationship for?” This is an option that can be used with Functional Dependency (FD) rules. The text within DCV’s “Rules Info” pane, which appears when an FD rule is selected, probably needs to be updated to provide a clearer explanation (maybe the word “bidirectional” should be substituted).
5. One user would like to add specific error margins for some rules, i.e. so that DCV would not consider that an entry violates a rule when the values are within a specific error margin. This was reported in the context of FD rules, where it cannot really apply. Functional Dependency (FD) and Conditional Functional Dependency (CFD) rules are logical data consistency checks and so cannot allow error margins. However, maybe an error margin can be introduced as a function (e.g. *epsilon*) to be used for some types of Constraint Rules (CR) over a relation (e.g. equality +/- some acceptable error margin).
6. “Data cleaning doesn’t have directions.” This was a comment relating to the 4 circular green buttons that appear on the top part of DCV (can be seen in Figure 8). The way these buttons become active from left to right, suggest a kind of sequence of steps/a specific workflow: first the user loads the data or selects an already loaded dataset (“Projects” button), then applies data cleaning rules (“Data Cleaning” button) rule, then proceeds to derive new columns (“Compute Scores” button), and finally runs some data mining algorithms (“Knowledge Discovery” button). This user suggested that we consider redesigning this part of the interface, since a user can skip a step of this process, or can work in an iterative way (e.g. use knowledge discovery to help him find additional rules to implement on a later stage).
7. Some clinicians were confused by the way the tool reports numbers of errors. This can be easily fixed by changing the wording so that there is no confusion between numbers of cells that had errors and numbers of rules that returned violations.
8. “Can the rule be applied prospectively?” This can be done via DCV’s “workflows”. Unfortunately, we had no time to demonstrate this during the training.
9. Once DCV loads a dataset it checks for some predefined errors (e.g. it colours red the text of cells with improper/erroneous dates, or cells with values that appear to be outliers, etc.) Some users would like to have control of these predefined error checks.
10. Following problem #9, when some columns have multiple errors (e.g., during the training we presented one column with numerous dates that were incorrect (years appearing as ../../0015, instead of ../../2015), the user needs to go to each cell and correct/update the value of each cell one-by-one. A request was made for a way of applying this value update to all similar errors.
11. “Could mapping fields/semantic integration be added as functionality?” If mapping fields refers to schema matching, then yes, this is functionality already planned for a future version (preliminary work has already begun). Semantic integration is a bit more complicated to implement at the moment, but is added for future work.
12. “Has DCV been tested with big data?” It has been tested with very large synthetic data sets, but we are planning to test it with millions of rows from a true dataset (in collaboration with an external partner on an actual use-case that they proposed).

The above are only a few of the very valuable comments made during the tutorial. After the training session in Rome, we also gave the tool for testing to two Professors of Bioinformatics form the University of Athens, and we received additional valuable suggestions (e.g. to be able to save part of a dataset, when part of the data has already been processed, or to be able to save the data that was filtered via the interface/graphs, and so on.) All suggestions will be taken into consideration and will be prioritized for implementation in upcoming DCV releases.

Before closing this section, we should point out that more qualitative evaluation is required. Although, we have been testing each one of DCV's implemented functions with a number of unit tests and small benchmarks, what has not yet been done but is planned, is to use as benchmarks datasets that clinicians have already processed using other tools (Excel, scripts, SPSS, etc.) and then repeat the process with DCV, to compare the results. This is of course not meaningful for all of DCV's functionalities, but is a nice test for things such as computing functions/medical scores (as derived columns). For this, we are already planning to use as benchmarks a larger set of the eCRF obesity questionnaire dataset, a NND dataset, and a JIA dataset.

3. Testing KDD extensions to DCV and the AITION KDD platform

As described in D16.1 and D16.2, ATHENA is developing an end-to-end data cleaning, analysis and KDD platform. The proposed platform builds upon, combines, and extends the existing tools developed by ATHENA for data curation & cleaning (DCV - Data Curation and Validation tool) and medical knowledge discovery (AITION KDD Platform), on top of the EXAREME (ex ADP/madIS) distributed data management and processing platform. The platform consists of three well-defined "application" modules, each one responsible for a specific data analysis task, that are seamlessly integrated as depicted in Figure 5.

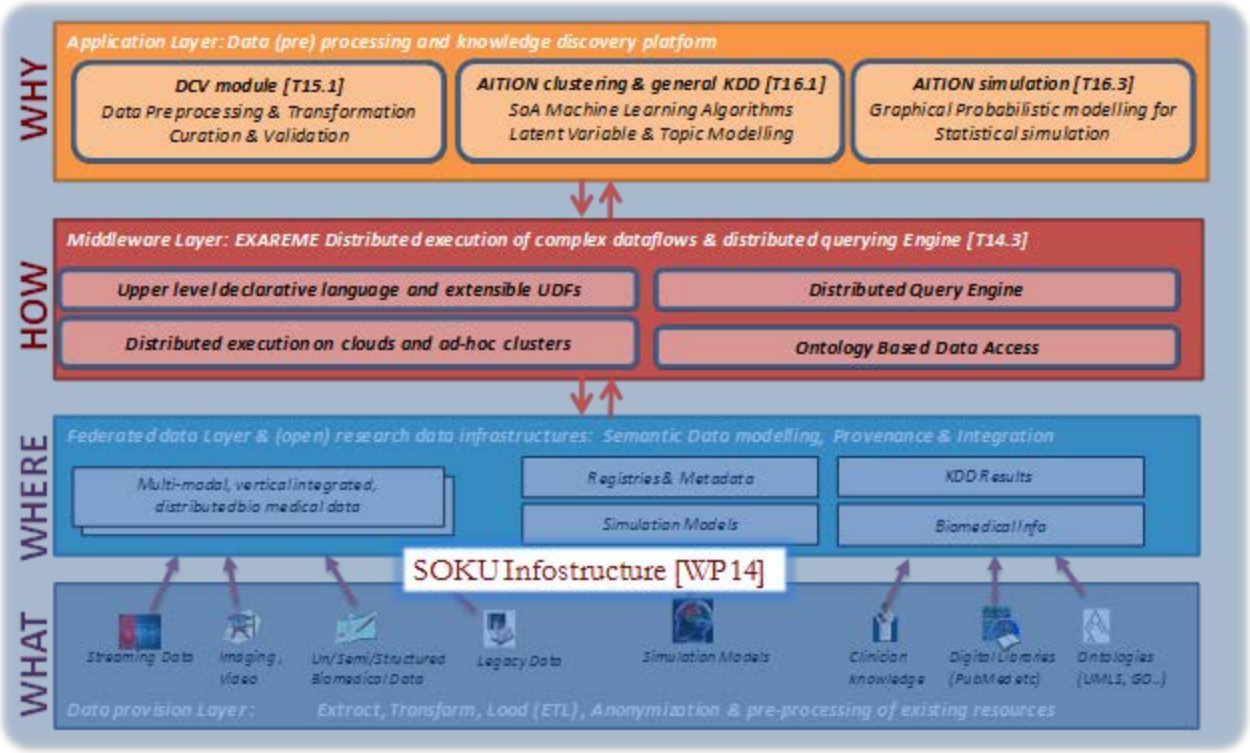

Figure 5: ATHENA data analysis platform

For task T16.1, some well-established Machine Learning (ML) techniques and algorithms were implemented on top of ATHENA's EXAREME (ex ADP/madIS) data flow processing system following the same architecture used by the DCV tool. This way a user is able to work on an integrated pipeline from data curation and pre-processing to the training of predictive models, using a common web-based data analysis platform.

3.1. EXAREME: madIS KDD operators

Aiming to expand our knowledge discovery repository, we exploited the *madIS* component of EXAREME to develop new functionalities for clustering data, reducing high-dimensionality and training predictive models for classification and regression. *madIS* is an extensible relational database system developed by ATHENA and built on top of the SQLite database with extensions implemented in Python (via APSW SQLite wrapper) [1]. Queries and data flows can be expressed via a declarative language, which is based on SQL with user-defined functions (UDFs) extended with parallelism primitives, iterations and an inverted syntax to easily express data pipelines. UDFs with arbitrary user code are natively supported within the data management engine of the system.

As descried in D16.2, four new KDD user-defined functions were developed:

a. one for creating *clustering* (unsupervised) models;

b. one for *training* the classification/regression (supervised) models, and one for *predicting* new values for unlabelled data samples based on the already trained models; and

c. one for *dimensionality reduction*

We tested all operators on synthetic and toy datasets (such as the Iris data set [2]) and also designed use-cases for real data in cooperation with clinicians (e.g. see NND use case, described in the next section). Separate scripts were constructed to test the functionality of KDD UDFs and all operators implement the *scikit*-algorithms presented in D16.1 successfully.

We also tested the information-loss when we reduce the dimensions of a dataset and then cluster the samples on the new space. A k-means clustering was held directly on the initial dataset (Figure 6 - model 'kmraw') and then we applied the model on the data after principal component analysis (PCA) had been applied (Figure 6 - model 'kmpca'). Both techniques grouped the data successfully in the same clusters.

```
In [69]: kmraw=km.fit(x)

In [70]: kmpca=km.fit(x_r)

In [71]: kmraw.labels_
Out[71]:
array([1, 1, 0, 0, 0, 1, 0, 2, 0, 1, 0, 0, 2, 2, 2, 0, 0, 0, 0, 1, 2, 0, 1,
       1, 1, 1, 2, 0, 2, 1, 0, 0, 1, 2, 2, 2, 0, 1, 2, 2, 0, 0, 0, 1, 2, 2,
       0, 2, 2, 0, 0, 0, 1, 2, 0, 0, 1, 1, 2, 0, 1, 2, 1, 0, 0, 0, 2, 0, 2,
       2, 1, 0, 0, 2, 1, 0, 0, 1, 1, 0, 2, 1, 1, 1, 1, 0, 2, 2, 0, 2, 0, 1,
       0, 0, 1, 2, 0, 1, 0, 1], dtype=int32)

In [72]: kmpca.labels_
Out[72]:
array([1, 1, 0, 0, 0, 1, 0, 2, 0, 1, 0, 0, 2, 2, 2, 0, 0, 0, 0, 1, 2, 0, 1,
       1, 1, 1, 2, 0, 2, 1, 0, 0, 1, 2, 2, 2, 0, 1, 2, 2, 0, 0, 0, 1, 2, 2,
       0, 2, 2, 0, 0, 0, 1, 2, 0, 0, 1, 1, 2, 0, 1, 2, 1, 0, 0, 0, 2, 0, 2,
       2, 1, 0, 0, 2, 1, 0, 0, 1, 1, 0, 2, 1, 1, 1, 1, 0, 2, 2, 0, 2, 0, 1,
       0, 0, 1, 2, 0, 1, 0, 1], dtype=int32)

In [73]: kmraw==kmpca
Out[73]: True
```

Figure 6: Clustering after dimensionality reduction: Identical results between applying k-means on both raw and dimensionally reduced data (Iris data).

```
988|0
989|0
990|0
991|0
992|0
993|0
994|0
995|3
996|3
997|3
998|3
999|0
1000|0
--- [0|Column names ---
[1|sampleID [2|predicted_label
Query executed and displayed 1000 rows in 0 min. 0 sec 17 msec.
```

Figure 7: Results returned by the 'skpredict' operator. Columns: ID from sample to be classified, the predicted class. query: create table as skpredict filename:DTmodel select * from t_preds; (where t_preds is the table with the new unlabelled data we want to classify.)

3.2. DCV: Knowledge Discovery Extension

In D16.2 ("Beta Prototype of KDD & Simulation platform"), we presented the beta version of DCV. We also described the machine learning extension that was added on the basic flow (Data cleaning-Compute Score-*Knowledge Discovery*) by adding an extra 'Knowledge Discovery' tab (see interface illustrated in Figure 8). Thus, the end-user besides being able to pre-process data (e.g. detecting errors or outliers), also has the opportunity to identify groups and similar cases or create models that predict the value of one or more target variables using the same platform.

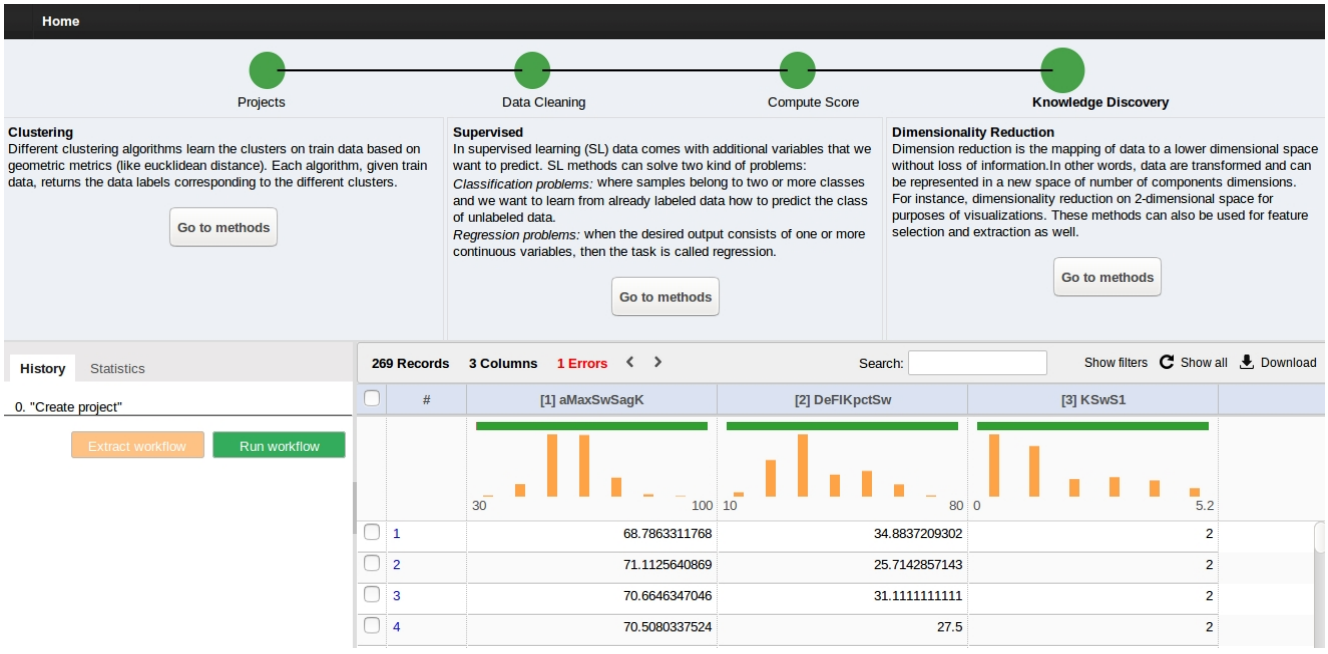

Figure 8: KDD user interface within the DCV tool, provided under the new "Knowledge Discovery" tab (i.e. the rightmost round green button).

3.2.1. NND Use-Case

Having a classified dataset of 270 children trials with Cerebral Palsy (CP) available, we tested some of our methods on these data. Our primary aim was to search for similar samples, which have been classified by clinicians in several groups of clinically accepted distinct gait movement patterns.

Specifically, there is a need for a classification that characterizes each CP gait by different degrees of membership for several gait patterns, which are considered by clinical experts to be highly relevant [3]. Machine learning techniques fit very well for such kind of problems. Predictive models can be created in order to automate the CP gait classification. Thus, our aim was to train classification models, based on rules developed by clinicians, and identify their prediction accuracy. Models with high accuracy (>80%) can be considered reliable in classifying new patients and can also predict possible future behaviour in their movement. Furthermore, search for patient similarities is facilitated as we use also methods (Decision Trees, Random Forests) that construct decision rules. Thus, in contrast to black-box methods, the clinicians can explore and see based on which rule/path the model classified each sample, and hence compare with their own rules.

Our analysis was based on a three-step workflow:

1. **Pre-processing:** NaNs, discretization
    a. Datasets including samples with no measured values (NaNs) were deleted from the training set as "NaN-samples" cannot represent real samples and decrease the model's accuracy.
    b. Discretization of continuous values is requested, as there are variables with only discrete measurements (integers 1, 2, 3, etc.)
2. **Training of models:** Two supervised learning methods were used
    a. (Naive) Bayesian approach: A probabilistic model represents a set of variables and their conditional dependences (the 'naive' assumption of independency among variables is used).
    b. Random Forests: Creates an ensemble (*forest*) of decision trees, each of which has been trained on a random subset of the data.
3. **Predictions** (Results): We calculated the mean prediction accuracy to evaluate the constructed models, using *k*-fold cross-validation (D16.2). Classification performance per class and sample contribution (confusion matrices) in classification accuracy are also measured. Finally, we extracted all the posterior-probabilities for each sample in the training dataset.

The available data consist of eleven joints in three planes. Each joint contains different movement patterns in which patients are classified. Hip and Foot below are excluded from the analysis, as they contained only one parameter-feature (Table 1).

Table 1: NND use case: joints and movement patterns in three planes. Hip and Foot below are excluded from the analysis, as they contained only one parameter-feature (highlighted in red).

Sagittal plane	Coronal plane	Transverse plane
Pelvis *(6 patterns)*	Pelvis *(4 patterns)*	Pelvis *(4 patterns)*
Hip *(3 patterns)*	Hip *(4 patterns)*	Hip *(3 patterns)*

Knee in stance & swing *(6 patterns each)*		Foot *(3 patterns)*
Ankle in stance & swing *(5 and 4 respectively)*		

For instance, knee joint in sagittal plane during stance consists of three predictor variables and one response variable PS1 (separated into 6 patterns):

Predictor variables:

a. Increased knee flexion at initial contact (*aIcSagK*) – continuous,

b. Earlier knee extension movement (*pctaMaxMStSagK*) – continuous, and

c. Knee extension in stance (*aMinStSagK*) – discretized.

Response variable (Classes to classify/predict):

- *KStS0:* Normal pelvic posture/motion
- *KStS1:* Increased pelvic range of motion
- *KStS2:* Increased pelvic anterior tilt on average
- *KStS3:* Increased pelvic anterior tilt and increased range of motion
- *KStS4:* Decreased pelvic anterior tilt on average
- *PS5:* Decreased pelvic anterior tilt and increased range of motion

Table 2: NND dataset example for knee joint in sagittal plane during stance

	aIcSagK	pctaMaxMStSagK	aMinStSagK	*KStS1*
2	26.7439651489	0	2	***2***
3	11.3626565933	12.6213592233	2	***0***
4	22.2974491119	7.1428571429	2	***2***
5	13.4327917099	9.9099099099	2	***0***
6	37.5202064514	0	3	***5***
7	38.2755737305	0	3	***5***
8	37.1203918457	0	3	***5***
9	40.7301979065	0	3	***5***
10	22.3935604095	9.6774193548	3	***5***
11	21.725107193	11.6279069767	2	***1***
12	23.8037586212	11.2359550562	3	***5***
13	19.2760543823	11.9565217391	3	***1***
14	21.5129776001	12.9032258065	3	***5***
15	20.0130290985	12.0879120879	2	***1***

3.2.2. Workflow

Pre-processing step – Discretization

In the NND use case, some variables such as *aRomSagP* consist of continuous values, while others such as *PelvisAntTilt* include only integers. Models cannot handle such kind of problems well, as the distributions among variables are different. Thus, continuous variables are discretized based on rules that are provided by the NND team of clinicians based on their experience. A special *madIS* UDF for such discretization operations was thus developed for general purpose and for the integration with the DCV platform. See Figure 9 for an illustration of a discretization example on an NND data sample.

	aIcSagK	pctaMaxMStSagK	aMinStSagK	*KStS1*
2	26.7439651489	0	2	***2***
3	11.3626565933	12.6213592233	2	***0***
4	22.2974491119	7.1428571429	2	***2***
5	13.4327917099	9.9099099099	2	***0***
6	37.5202064514	0	3	***5***
7	38.2755737305	0	3	***5***
8	37.1203918457	0	3	***5***
9	40.7301979065	0	3	***5***
10	22.3935604095	9.6774193548	3	***5***
11	21.725107193	11.6279069767	2	***1***
12	23.8037586212	11.2359550562	3	***5***
13	19.2760543823	11.9565217391	3	***1***
14	21.5129776001	12.9032258065	3	***5***
15	20.0130290985	12.0879120879	2	***1***

	aIcSagK	pctaMaxMStSagK	aMinStSagK	*KStS1*
2	2	1	2	***2***
3	1	2	2	***0***
4	2	1	2	***2***
5	1	1	2	***0***
6	2	1	3	***5***
7	2	1	3	***5***
8	2	1	3	***5***
9	2	1	3	***5***
10	2	1	3	***5***
11	2	2	2	***1***
12	2	1	3	***5***
13	2	2	3	***1***
14	2	2	3	***5***
15	2	2	2	***1***

Figure 9: Discretization step: aIcSagK example: if *1.5 <* aIcSagK *< 14: normal (value=1) else if aIcSagK > 14: increased (value=2)*

Training step

After discussion with the NND clinicians, we decided to use the Naive Bayes (NB) and Random Forests (RF) algorithms for this specific use-case. For this purpose, we focused on the integration of these two supervised learning methods with DCV. Thus, the user can define the corresponding parameters to initialize a predictive model (refer to D16.2 for parameters used), and then the model fits the data during the training step. Finally, the outcome predictions and results are presented.

Both methods are applied on the nine discretized and two continuous (knee in swing, ankle in stance) datasets.

1. **Naive Bayes approach:** We used the *MultinomialNB() scikit* function, which is suitable for classification with discrete features. For continuous ones, a Gaussian classifier (*GaussianNB()*) is selected, where the distributions of features are assumed to be Gaussian. This method relies on the "naive" assumption that all variables are independent among each other and no feature correlations are taken into consideration.
2. **Random Forests:** Building an ensemble of 100 *decision trees* the 'forest' 'votes' for the best class and classifies the sample. We define three parameters for the forest:
    a. Number of estimators: 100
    b. Maximum tree depth: If None (not recommended due to over fitting), nodes are expanded until all leaves contain less than *min_samples_split samples* (see 'c' below). We defined *max_depth* = 4

c. Minimum number of samples to split an internal node in each tree. A very small number will usually mean the tree will overfit, whereas a large number will prevent the tree from learning the data. *(*selected: *min_samples_split = 40)*

Cross-validation

In order to evaluate the model's general prediction accuracy, it is necessary to have some unlabelled samples to test our classifier predictions, and so a test set is held out for final evaluation. In the basic *k*-fold CV approach, the training set is split into *k* smaller sets and the following procedure is followed for each of the *k* "folds":

- A model is trained using the folds as training data;
- The resulting model is validated on the remaining part of the data and a performance measure, such as *accuracy*, can be calculated.

Herein, we used *10-fold* cross-validation for estimating the predictor's precision scores. However, the classification performance per class (movement pattern) of each joint were measured based on all samples, for more stable results.

Predictions and Results

At the end of the pipeline referred above, we produced specific results according to clinicians' feedback and the clinical objectives we would like to achieve. Particularly, *for each joint*:

1. We trained a model and assessed its accuracy based on 10-fold CV.
2. We calculated the classification performance for each class (range: 0 to 1) and the number of samples that classified correctly (into the class that they really belong to – i.e. *true positives*).
3. Confusion Matrices were constructed to compare the expert versus the predicted patterns (number of samples classified into each pattern).
4. We produced CSV files including all the posterior probabilities for each patient.

The first thing we observed is that Naive Bayes (NB) has a poor estimation performance on discretized datasets, while Random Forests (RF) seem to have a high overall prediction accuracy. For instance, for *pelvis joint in sagittal plane* there is a large "distance" between the two methods. NB classified all samples into the 3rd class (PS3), something totally meaningless. Yet, this seems legitimate since the overall accuracy is too low (36.68%). On the other hand, RF works well achieving a score of 77.87%.

The following two tables (Table 3 & Table 4) summarise the results of this comparison between the NB & RF methods for the *pelvis joint in sagittal plane* case.

Table 3: Totally uncorrelated results between two methods for pelvis joint in sagittal plane. In the Naïve Bayes (NB) case, all samples are classified on PS3 class, while the distribution given by Random Forest (RF) is much more informative. Classification score: number showing the classification performance as a value between 0 (no sample classified correctly) and 1 (all samples classified correctly).

Pelvis - sagittal plane (Classification performance)

NB	Classification Score	Samples
PS0	0	0 of 57
PS1	0	0 of 54
PS2	0	0 of 56
PS3	1	98 of 98
PS4	0	0 of 1
PS5	0	0 of 2
		Total samples: 268

Prediction Accuracy (cv: 10-fold): 36.68%

RF	Classification Score	Samples
PS0	0.7	40 of 57
PS1	0.83	45 of 54
PS2	0.66	37 of 56
PS3	0.91	89 of 98
PS4	0	0 of 1
PS5	0	0 of 2
		Total samples: 268

Prediction Accuracy (cv: 10-fold): 77.87%

Table 4: Confusion matrices for pelvis joint in sagittal plane case, comparing the Naïve Bayes with the Random Forest method.

Pelvis - sagittal plane (Confusion matrix)

Expert's knowledge	Predicted (number of samples)					
	PS0	*PS1*	*PS2*	*PS3*	*PS4*	*PS5*
PS0	**0**	0	0	57	0	0
PS1	0	**0**	0	54	0	0
PS2	0	0	**0**	56	0	0
PS3	0	0	0	**98**	0	0
PS4	0	0	0	1	**0**	0
PS5	0	0	0	2	0	**0**
NAIVE BAYES						

Expert's knowledge	Predicted (number of samples)					
	PS0	*PS1*	*PS2*	*PS3*	*PS4*	*PS5*
PS0	**40**	8	8	1	0	0
PS1	3	**45**	3	3	0	0
PS2	2	0	**37**	17	0	0
PS3	0	8	1	**89**	0	0
PS4	0	1	0	0	**0**	0
PS5	2	0	0	0	0	**0**
RANDOM FOREST						

The above case is the most representative of all joint movement patterns except for the *pelvis in coronal and transverse plane* case (see Table 5 & Table 6 for results).

However, for the joints that we have available, continuous dataset predictors behave with a very good performance. These two datasets refer to *knee in sagittal plane during stance (KSwS)* and *ankle in sagittal plane during stance (AStS).* Results can be seen in Table 7 & Table 8.

Table 5: Classification performance results with Naive Bayes (NB) and Random Forest (RF) methods for pelvis coronal plane case.

Pelvis - coronal plane (Classification Performance)

NB	Classification Score	Samples
PC0	0.9	95 of 105
PC1	0.9	105 of 117
PC2	0	0 of 21
PC3	0	0 of 25
		Total samples: 268

Prediction Accuracy (cv: 10-fold): 74.34%

RF	Classification Score	Samples
PC0	0.89	93 of 105
PC1	0.88	103 of 117
PC2	0.71	15 of 21
PC3	0.72	18 of 25
		Total samples: 268

Prediction Accuracy (cv: 10-fold): 84.76%

Table 6: Classification performance results with Naive Bayes (NB) and Random Forest (RF) methods for pelvis transverse plane.

Pelvis - transverse plane (Classification Performance)

NB	Classification Score	Samples
PT0	0.92	98 of 107
PT1	0.98	98 of 100
PT2	0	0 of 32
PT3	0	0 of 30
		Total samples: 269

Prediction Accuracy (cv: 10-fold): 72.90%

RF	Classification Score	Samples
PT0	0.92	98 of 107
PT1	0.97	97 of 100
PT2	0.97	31 of 32
PT3	1	30 of 30
		Total samples: 269

Prediction Accuracy (cv: 10-fold): 95.17%

Table 7: Classification performance results with NB and RF methods for KSwS case.

Knee in swing - sagittal plane - continuous (Classification Performance)

NB	Classification Score	Samples
KSwS0	1	96 of 96
KSwS1	0.88	69 of 78
KSwS2	0.89	24 of 27
KSwS3	0.87	26 of 30
KSwS4	0.92	23 of 25
KSwS5	0.85	11 of 13
		Total samples: 269

Prediction Accuracy (cv: 10-fold): 90.74%

RF	Classification Score	Samples
KSwS0	0.99	95 of 96
KSwS1	0.91	71 of 78
KSwS2	1	27 of 27
KSwS3	0.97	29 of 30
KSwS4	1	25 of 25
KSwS5	0.92	12 of 13
		Total samples: 269

Prediction Accuracy (cv: 10-fold): 93.14%

Table 8: Classification performance results with NB and RF methods for AStS case.

Ankle in stance - sagittal plane - continuous (Classification performance)

NB	Classification Score	Samples
AStS0	0.83	74 of 89
AStS1	0.87	71 of 82
AStS2	0.85	29 of 34
AStS3	0.95	19 of 20
AStS4	0.91	40 of 44
		Total samples: 269

Prediction Accuracy (cv: 10-fold): 84.36%

RF	Classification Score	Samples
AStS0	0.94	84 of 89
AStS1	0.83	68 of 82
AStS2	0.79	27 of 34
AStS3	1	20 of 20
AStS4	0.93	41 of 44
		Total samples: 269

Prediction Accuracy (cv: 10-fold): 84.72%

In conclusion, we observe that in both, the pelvis in coronal and transverse plane, the predictors we use achieve good performance for classes 1 and 2. For the remaining cases, only RFs seem to behave appropriately and in line with the rules and decisions of experienced clinicians. However, both methods perform very well on continuous datasets. Although in gait analysis we mainly work with categorical variables (e.g. in case we want to examine also the conditional dependences among the variables and classes), the results for continuous joint datasets make sense and it might be useful to work on more patterns with continuous variables.

3.2.3. Clustering and Dimensionality reduction

A cluster analysis was also held for *knee in sagittal plane during swing*. The results depicted below () show that we cannot find the correct number of clusters (except for k-means of course, where we predefine the number of clusters to match the number of classes). This is not surprising for clustering methods applied on a dataset with only two variables. Hence, there is a need to enrich the datasets with more features.

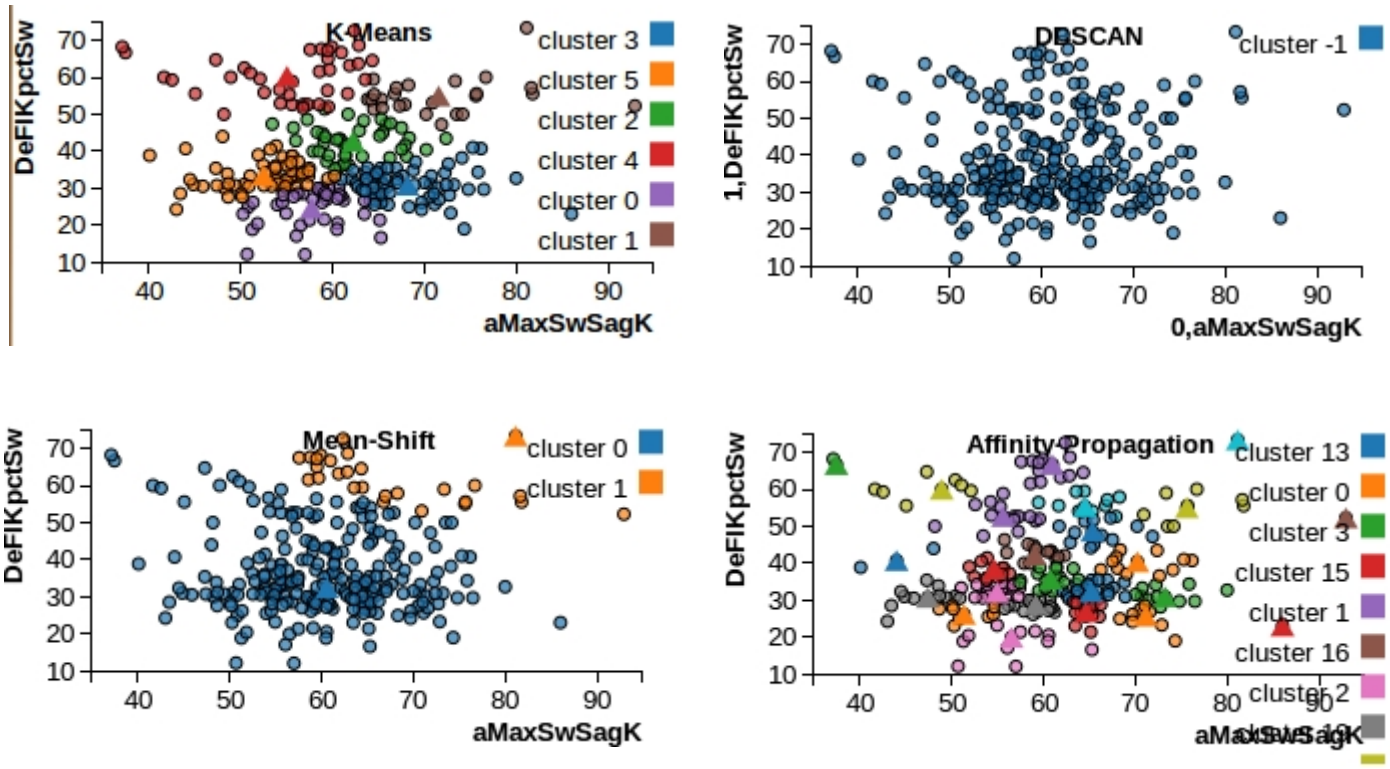

Figure 10: Cluster analysis for knee in sagittal plane during swing, using four different algorithms: K-Means, DBSCAN, Mean-Shift, and Affinity Propagation.

4. Conclusion

In this report we present the WP17 activities regarding testing and validation of the Beta Prototype of the KDD & Simulation Platform developed under WP16, focusing on testing of (1) the beta version update of the Data Curation and Validation (DCV) tool, and of (2) the further developments delivered on the new web-based KDD platform (AITION KDD).

Particularly for DCV, 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 DCV tool.

The beta prototype was presented and used by clinicians during the training session in Rome in mid February 2016. The great majority of clinicians judged the tool as very friendly, robust, very useful in detecting erroneous data and in helping them decide how to correct their data. In addition, they provided us with extremely useful feedback and comments, as well as pointing out a couple of minor bugs. All suggestions will be taken into consideration and are being prioritized for implementation in upcoming DCV version releases.

5. References

[1] madIS https://code.google.com/p/madis/ : “Complex data analysis/processing made easy”

[2] Iris data set: https://archive.ics.uci.edu/ml/datasets/Iris

[3] L. Van Gestel *et al.*, Probabilistic gait classification in children with cerebral palsy: A Bayesian approach, 2011

6. Appendix: DCV Questionnaire used during the Rome training session

MD-PAEDIGREE

DCV

Feedback questionnaire

Was it user friendly?

1) Not at all 2) I am not sure 3) Very 4) Much more than I expected

How was the robustness?

1) Not good 2) I am not sure 3) very good 4) Much more than I expected

How useful is it to detect erroneous data?

1) Not useful at all 2) I am not sure 3) Very useful 4) Much more than I expected

How useful is the tool in your decision for the correction of your data?

1) Not useful at all 2) I am not sure 3) Very useful 4) Much more than I expected

Do you have any problems with your data that the tool cannot presently address and you would like for that functionality to be added?

How do you judge the overall effectiveness and efficiency of the tool?

Which information are you missing?

How do you think we could improve this tool?