Guided Data Cleaning through a Web Interface

Data Cleaning with Guided Analytics

Rosaria Silipo
Rosaria.Silipo@knime.com

Phil Winters
Phil.Winters@knime.com

Greg Landrum
Greg.Landrum@knime.com

Christian Albrecht
Christian.Albrecht@knime.com

Summary

Do you remember the KNIME whitepaper "Seven Techniques for Data Dimensionality Reduction"? In that whitepaper, we described and implemented seven commonly used techniques to remove uninformative data columns from a data set.

This whitepaper takes this further. It implements some of the described techniques and discusses additional web-based interactive guided implementation, which also allows domain expert users to be involved in the process.

The goal of this whitepaper is to implement some of the most common dimensionality reduction techniques through an interactive step-wise web interface.

Percentage of missing values, standard deviation, skewness, and correlation are calculated for each column of the data set. A summary web page then displays the least performing columns according to those measures. Here columns can be manually selected and, if selected, will be removed from the original data set.

The web page is generated by a wrapped node in the KNIME workflow containing Quickform and Javascript based nodes. Indeed, the views of such nodes are displayed on a web page, when the workflow is running on KNIME WebPortal.

Now, while the data cleaning process is generally regarded as a tedious and repetitive process with little space for ingenuity, this step-wise implementation, which runs in a web browser, allows easy and creative experimentation to reduce the dimensionality of the original data set.

We hope you will enjoy it as much as we did!

The workflow is available for download from the EXAMPLES Server under **50_Applications/ 25_DataCleaning_WebPortal**

Table of Contents

Summary 1

The 80% Problem 2

The Data Cleaning Process 3

- Techniques for Dimensionality Reduction 4
- Outlier Removal 5
- Data Quality Measure 5

Data Cleaning through KNIME WebPortal 6

- Step 1. Select and Read the Data File 7
- Step 2. Select the Column to be the Prediction Target 8
- Step 3. Quality of Original Data Set and Interactive Dimensionality Reduction 9
- Step 4. Quality Measure of New Data Set 13

Data Cleaning through KNIME Workflow 15

Conclusions 19

The 80% Problem

If you have implemented any data analytics application, even a very simple one, you will know that data cleaning is a huge key to success. Present poor quality, dirty data to the model and the model will produce poor quality, dirty classification/prediction results. There is no way around it: data cleaning has to be done and must be performed sufficiently well before proceeding with other more challenging steps in the analytics.

This is the so-called 80% problem. Around 80% of a data analytics project involves data cleaning.

For KNIME Analytics Platform this translates into mostly yellow workflows with sparse blobs of other colors here and there. Yellow is in fact the color of Data Manipulation nodes. Beware of workflows with little yellow!

Could everyone who likes data cleaning raise their hands? I cannot see you, but I am sure only a few hands have been raised. Here is the problem: data cleaning is seen as unappealing work and therefore delegated as much as possible to less expert colleagues. As an example of this attitude, I would like to draw attention to the New York Times' headline shown in figure 1.


HOME SEARCH

The New York Times

TECHNOLOGY

For Big-Data Scientists, 'Janitor Work' Is Key Hurdle to Insights

By STEVE LOHR AUG. 17, 2014

Data scientists, according to interviews and expert estimates, spend from 50 percent to 80 percent of their time mired in this more mundane labor of collecting and preparing unruly digital data, before it can be explored for useful nuggets.

http://www.nytimes.com/2014/08/18/technology/for-big-data-scientists-hurdle-to-insights-is-janitor-work.html

Figure 1.

The New York Times on August 18th 2014, describing the 80% problem of data cleaning in any data analytics project.

The New York Times, indeed, describes data cleaning as 'Janitor Work', giving it a negative and boring connotation. Not everybody agrees with that.

Almost one year later, Monica Rogati, at the time VP of data at Jawbone, posted the tweet in figure 2. She describes data preparation as detective work, where intuition, experience, ingenuity, and pragmatism play an underestimated role.

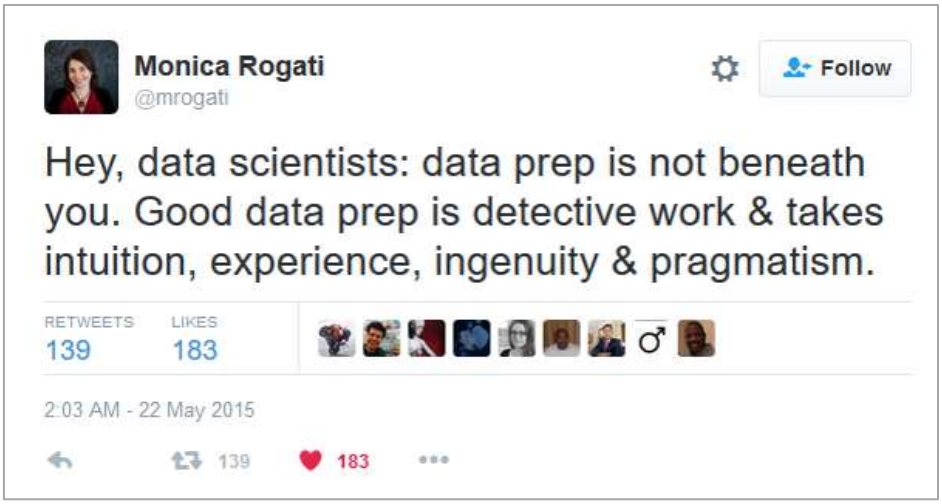

Figure 2.

Monica Rogati's tweet on May 22nd 2015, describing data preparation as detective work.

The goal of this whitepaper is to structure the data prepapration process as much as possible and to build an intuitive web-interface around it.

Data cleaning is often tied with domain knowledge, which can be beyond the data scientist's scope. However, the data scientist can help make the process more structured, automatic, and give it a more intuitive interface.

The Data Cleaning Process

When implementing a data cleaning process / application, we need to make sure that it follows a few basic principles. We want it to run:

- Reliably (complex operations have to run automatically)
- Cross-domain (running on different domains without manual adjustments)
- Interactively (allowing for human supervision)
- From a web browser (no KNIME expertise required)
- On demand (It has to run when and as many times we want)

The workflow described in this whitepaper is implemented with these 5 requirements taken into account.

Data preparation usually involves three stages:

- dimensionality reduction to remove useless and uninformative data columns;
- record cleaning to remove outliers, noisy and generally empty or sparsely populated records;
- and finally transformations such as aggregations to move from the raw data to the data level we need.

The first two steps are part of data cleaning. Here we will concentrate particularly on dimensionality reduction.

Techniques for Dimensionality Reduction

In this project, we implemented some algorithms for dimensionality reduction, as described in another KNIME whitepaper "Seven Techniques for Data Dimensionality Reduction".

Since most algorithms are optimized row-wise, the introduction of a large number of data columns can dramatically affect their performance. Even though newer parallelized versions of existing machine learning algorithms are being published every day, the identification and removal of useless and uninformative data columns can still speed up execution and improve results.

In the Seven Techniques whitepaper, we explored the following techniques to reduce the number of data columns in a data set.

- ***Missing Values***. Data columns with a percentage of missing values higher than a given threshold will be removed.
- ***Low Standard Deviation***. Data columns with standard deviation lower than a given threshold will be removed.
- ***High Correlation***. In pairs of data columns where correlation is higher than a threshold, the first one will be removed.
- *Principal Component Analysis (PCA)*. See reference https://en.wikipedia.org/wiki/Principal_component_analysis.
- *Infrequent Choices in Random Forest shallow Trees*. Here a random forest with shallow tree is implemented and analyzed. Only the data columns most often chosen as candidates are kept.
- *Backward Feature Elimination.*
- *Forward Feature Construction.*

Let's adopt only the first 3 of those techniques for our attempt to develop a more interactive data cleaning application. We will neglect for now the

last 4 techniques, either because they do not allow much interactivity (PCA and random forest) or because they are too slow to be implemented on large data sets (backward feature elimination and forward feature construction).

Let's also add one new dimensionality reduction technique.

- ***Low Skewness***. Data columns with skewness too close to 0, according to a threshold, will be removed.

Outlier Removal

Record cleaning is not within the scope of this whitepaper. However, we are going to show an outlier removal option, just as an example of how easy it could be.

Outlier removal is performed manually by the end user through visual inspection on an interactive scatter plot.

Data Quality Measure

If we decide to remove data columns and/or data rows, we must make sure that we are actually removing noise and not useful information. How do we make sure of that? We measure the data set quality before and after cutting, to see to what extent cutting has affected performances – if at all.

There are many ways to measure data quality, mostly originating from statistical measures calculated on the data set.

A classic measure for example is the Cronbach Alpha. The Cronbach Alpha uses the correlation matrix to measure homogeneity of the data set. Unfortunately, it has been devised to run on small statistical samples and depends heavily on the density of the calculated correlation matrix. As a result, it might fail depending on the size of the data set and on the nature of the correlation matrix.

The most effective way is still machine-learning based. We run a cross-validation procedure on a subset of the entire data set using a simple, less demanding machine-learning algorithm, such as a shallow decision tree. In this case, we quickly reach an estimation of the average error (or average accuracy) describing the quality of the data set.

We apply one or both of these two measures before and after the column and/or row removal operations. If the average error from the cross-validation procedure has not increased beyond some tolerance boundaries, then the data dimensionality reduction has not removed any sensible information from the original data set. Similarly, if you use the

Cronbach Alpha, you need to make sure that the Cronbach Alpha values before and after column and/or row removal have not decreased beyond some defined tolerance boundaries.

Data Cleaning through KNIME WebPortal

The goal of this whitepaper is to develop a workflow for data cleaning, and particularly for dimensionality reduction, that is reliable, can be run on a web browser, on demand, be portable across domains, and interactive.

The open source KNIME Analytics Platform already guarantees reliability and running on demand. If correctly configured, the workflow can also run across domains without requiring any adjustments.

To run on a regular schedule, the workflow needs to run on KNIME Server. To run from a web interface it needs to run from KNIME Server including the WebPortal feature.

The first project we tried was a next best offer project on CRM data. The target was to predict the likelihood of a customer buying additional lawyer insurance. The data file was “NBO Input Data.table” and contained a field named “lawyer insurance” describing whether the customer had a lawyer insurance product (1) or not (0).

The workflow is organized in 4 parts.

1. File upload using a File Upload Quickform node. The file to upload can only be of type.table or .csv
2. Selection of target column
3. Calculation of the quality of the input data set. Calculation of some measures for each column to be used later for dimensionality reduction, such as:
    a. Percentage of missing values
    b. Standard deviation
    c. Skewness
    d. Correlation with other columns

    Such measures are then displayed on a web page to be inspected by the end user and used to make a final decision about column removal.

    Additionally, a scatter plot is also created for outlier inspection and removal.
4. Calculation of the quality of the final data set after selected columns and/or rows have been removed.

Each one of these parts needs a web-based user interface for KNIME WebPortal, which translates into a sequence of wrapped nodes in the workflow, containing Quickform and/or Javascript-based nodes.

Let's now have a look at the sequence of web pages and later, at the sequence of wrapped nodes that generated them.

Step 1. Select and Read the Data File

After you have logged into KNIME WebPortal with your credentials, on the left you will see the list of the workflows available for execution, given the access rights for your account.

Select the workflow you want to execute, in this case the workflow called Data_Cleaning_WebPortal_v2.0, and click Start in the frame on the right, as indicated in figure 3.

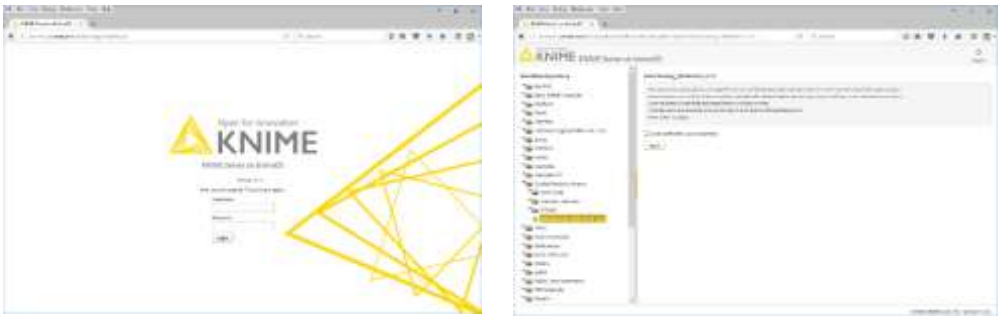

Figure 3.

KNIME WebPortal: login page (left) and Start page for the selected workflow (right).

The first interaction of the workflow with the end user is to ask for the input file, i.e. the file containing the examples to build the predictive model.

A short text describes the kind of file that is expected, maybe also the use case, and the final objective of the workflow.

Using the Upload button the end user can navigate the local machine, select, and upload the requested file (figure 4). The data files are in a folder data in the workflow folder. To access it just use the knime:// protocol as:

```
knime://knime.workflow/data/NBO_input_5_final.table.
```

In the same folder you can find another file accessible as:

```
knime://knime.workflow/data/malariahts_data.table
```

to be used later.

The workflow then continues in the background with reading the file and importing the data into KNIME Analytics Platform.

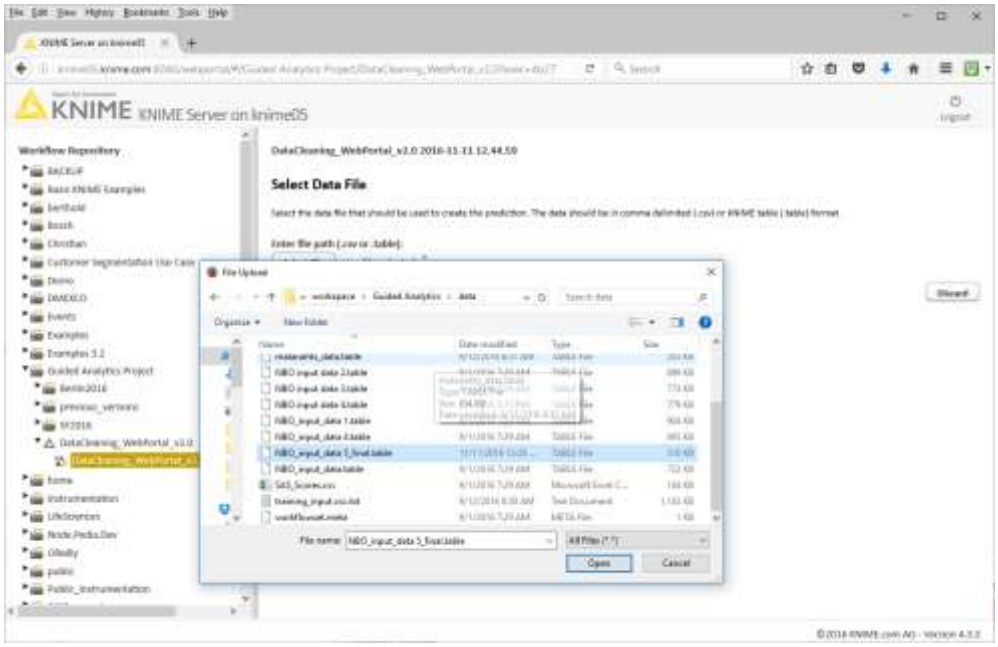

Figure 4.

File Upload dialog on the KNIME WebPortal.

Step 2. Select the Column to be the Prediction Target

Since the goal is to train a model, we will use one column from the data set as the prediction target and the remaining columns as input features. Then in the next web page of this guided execution, the end user is presented with the selection of the prediction target.

In this first run we are dealing with CRM data and the goal is to predict the likelihood of each customer buying additional lawyer insurance.

The column “lawyer insurance” is populated with 0s and 1s, representing respectively the fact that the customer has (1) or does not have (0) a lawyer insurance already. We use these examples as training set to build the model. Therefore, the target variable is this lawyer insurance data column.

Here the dialog asks you to select the column containing the target values; we select the “lawyer insurance” column.

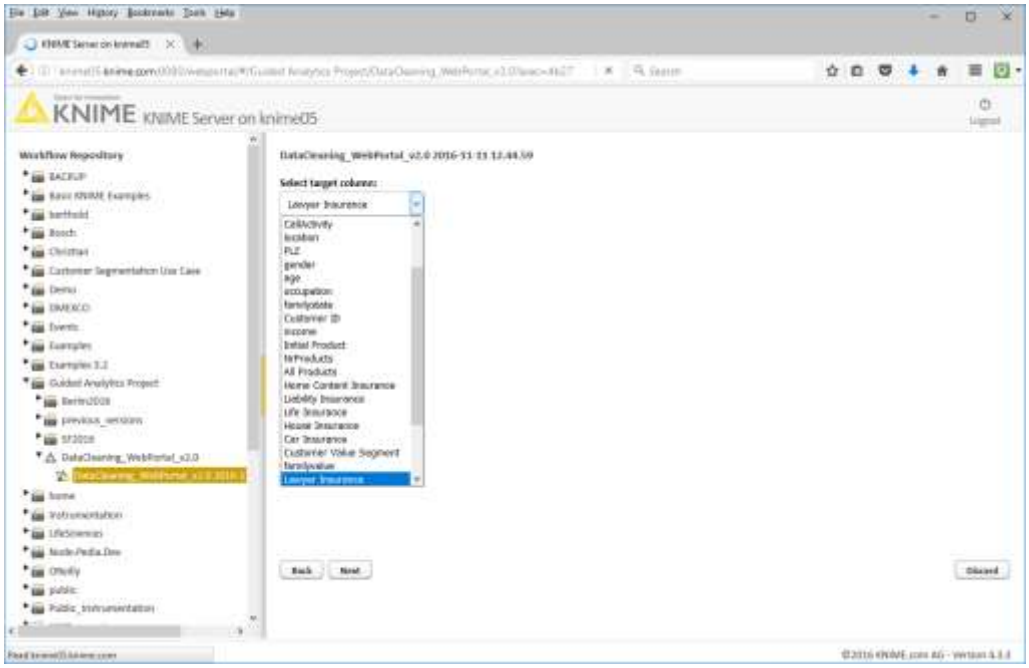

Figure 5.

Selecting the "lawyer insurance" column as the target variable to train the upcoming model.

Step 3. Quality of Original Data Set and Interactive Dimensionality Reduction

This third guided execution step (web page) is the heart of this workflow. Here the web page presents:

- A measure of the quality of the original data set (figure 6)
- Tables including lists of columns (figures 7, 8, and 9) that show the:
    - highest percentage of missing values
    - lowest standard deviation
    - skewness closest to 0
    - highest correlation with another column in the data set

Data set quality is measured through the average error in percent from a cross-validation session, with 10 folds, and using a shallow decision tree for a quick evaluation.

The quality of our data set actually seems to be fantastic! The average error coming out of the cross-validation procedure lies below 1%! This is a very easy problem to solve then or… there is an error.

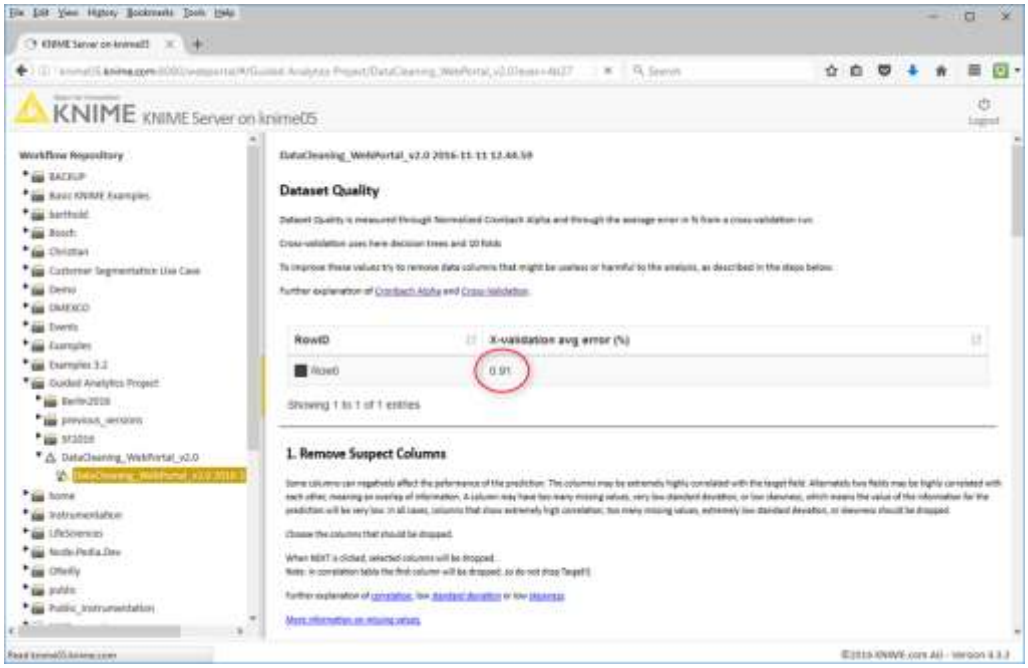

Figure 6.

Measure of data set quality as average error percentage from a cross-validation session using a shallow decision tree. Normalized Cronbach Alpha has not been used.

Caution. If the data set quality looks too good to be true, it probably is.

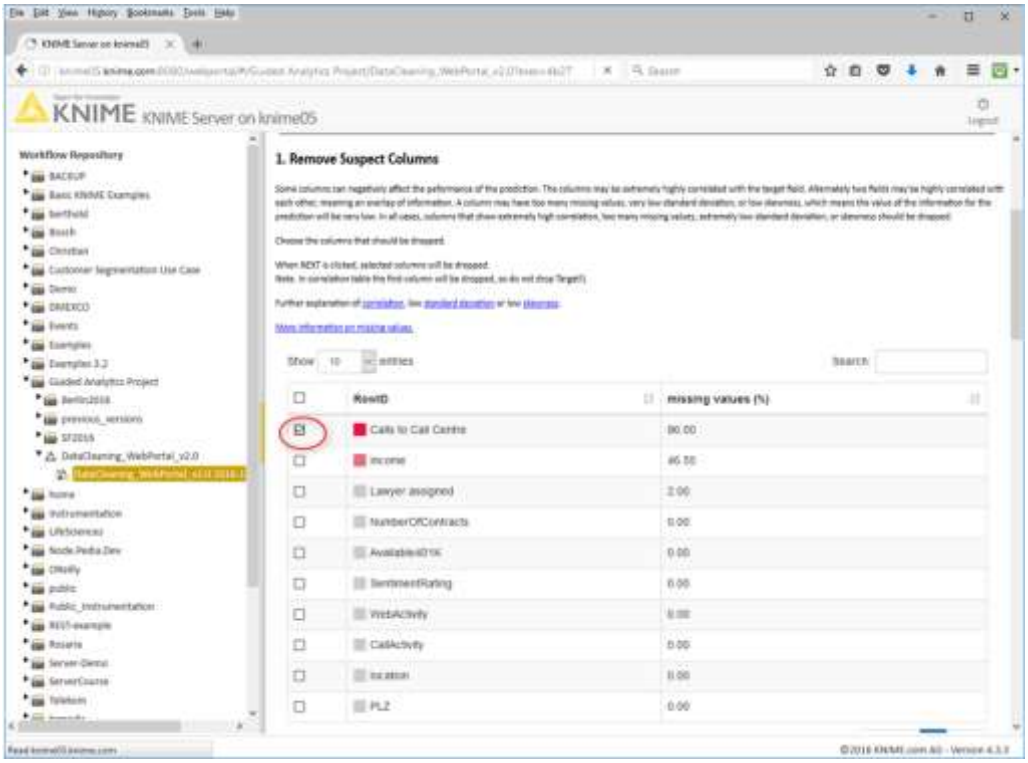

Figure 7.

Input columns with highest percentage of missing values.

Under the measure of data set quality, a number of tables describes how much information is contained in the least informative of the input columns. Each table row reports a column from the data set and its

corresponding measure of useful information. These rows can be selected. Data columns in selected table rows will be removed from the data set. We hope that removal of the least informative columns will increase the quality of the data set.

Figure 7 shows the list of input columns with the highest percentage of missing values. Column “Calls to Call Centre” has 96% missing values, i.e. it is pretty much empty. It is safe to assume that it will contribute very little to the model training. It is therefore selected for removal.

Figure 8 shows the list of input columns with the lowest standard deviation. The “Entity Number” column has 0 standard deviation, i.e. it contains only one value repeated over all rows. I do not remember exactly what this column was measuring. However, it is safe to assume that it will contribute very little to the model training. It is also selected for removal.

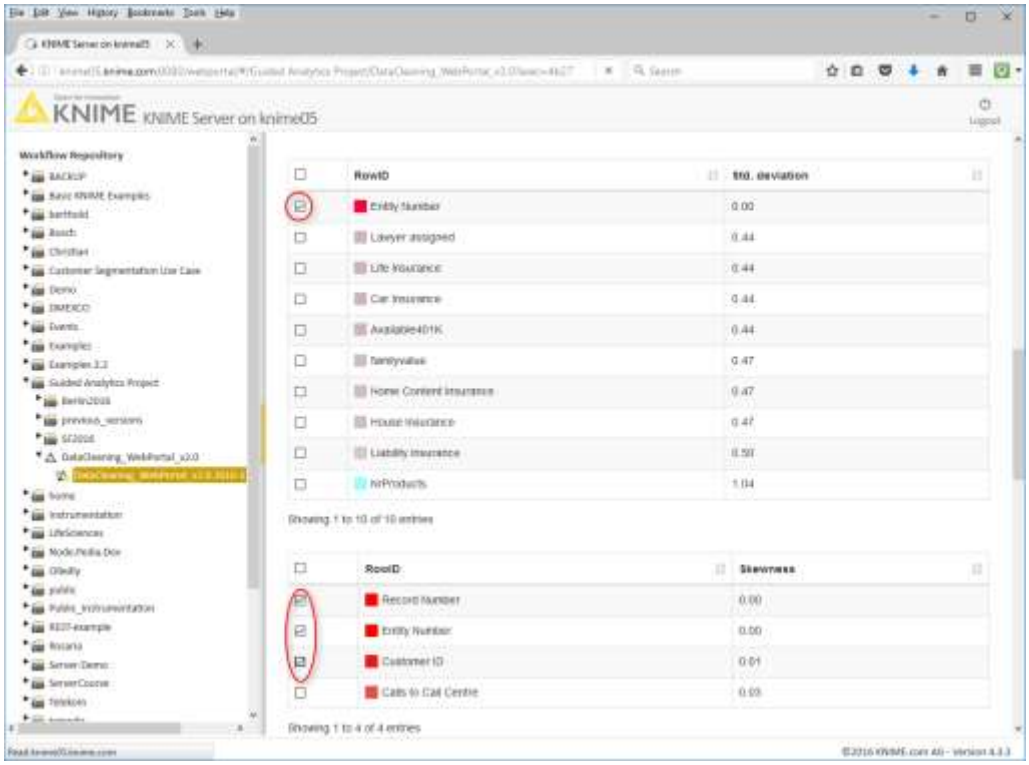

Figure 8.

Input columns with lowest standard deviation and closest to 0 skewness.

The lower part of figure 8 shows the list of input columns with skewness closest to 0. The columns “Record Number”, “Entity Number”, and “Customer ID” have 0 skewness. While I still do not really recall what “Entity Number” contained, I know that “Customer ID” and “Record Number” are unique IDs, respectively, for each customer and for each database record, with no other associated information. The only effect that such IDs can produce on the training procedure is overfitting. So, they are all selected for removal.

Figure 9 shows the list of input columns with the highest correlation to another input column. The columns “Lawyer assigned” and “Target” have a correlation of 1, i.e. they contain the same or almost the same values. “Target” column was the “lawyer insurance” column after renaming. “Lawyer assigned” contains the name of the lawyer assigned to the customer AFTER he bought a lawyer insurance. Right! A value in the “lawyer assigned” column is kind of a giveaway for having already bought a lawyer insurance. In a more realistic and current scenario, the “lawyer assigned” column would not contain any value before the customer bought an additional lawyer insurance. This column is an artefact and has to go.

When a row is selected in this last table, the first column is the one to be removed. Pay attention not to remove the Target column, otherwise you have no more target values on which to train the upcoming model!

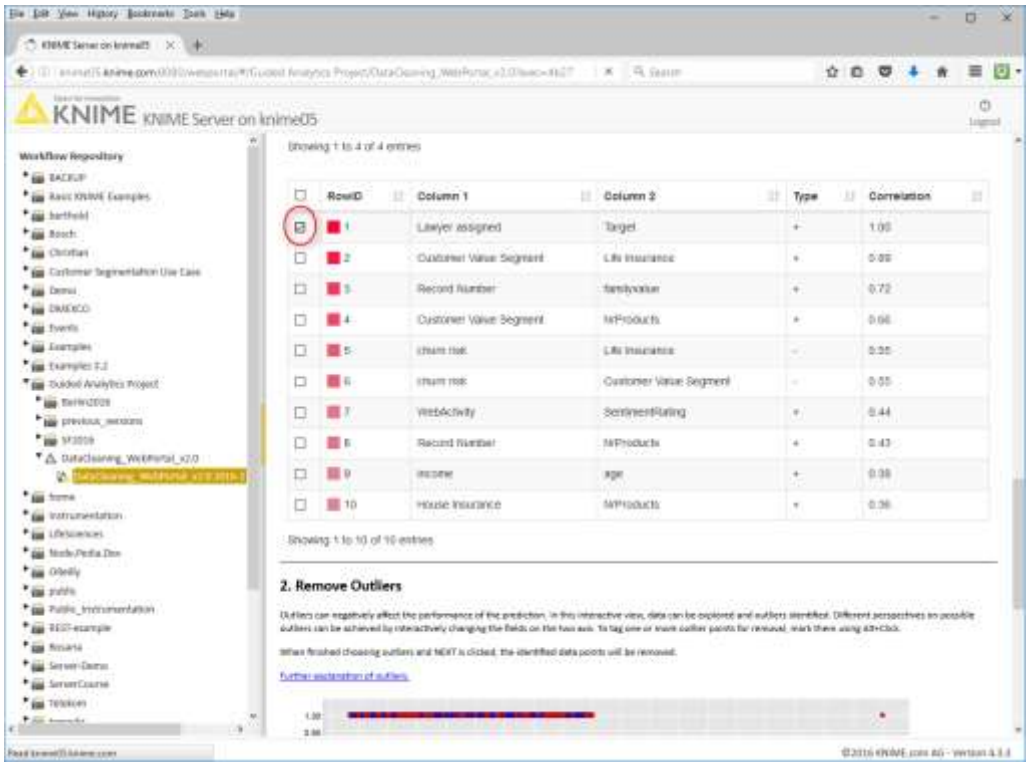

Figure 9.

Input columns with the highest correlation value to another input column.

The goal of this whitepaper is mainly to translate techniques for dimensionality reduction into a useful, intuitive, and interactive web interface. However, here we will also give you a little peek into the cleaning of data rows, through an interactive scatter plot that eliminates outliers in the data.

Figure 10 displays the scatter plot at the end of the same web page with age versus income. You can see that a few people appear to be 200 year old. Due to the isolated position of the points in the scatter plot and our knowledge of human maximum life expectancy, we can assume that this information is bogus, we can label those data rows as outliers, and we can select them via Alt-rectangle drawing to be removed.

Caution. The time needed to draw a scatter plot is of course longer, the larger the input data set is. Remove this step in the web-guided execution, if the data set you are using has a prohibitive size!

After pressing Next, all selected data columns and rows will be removed.

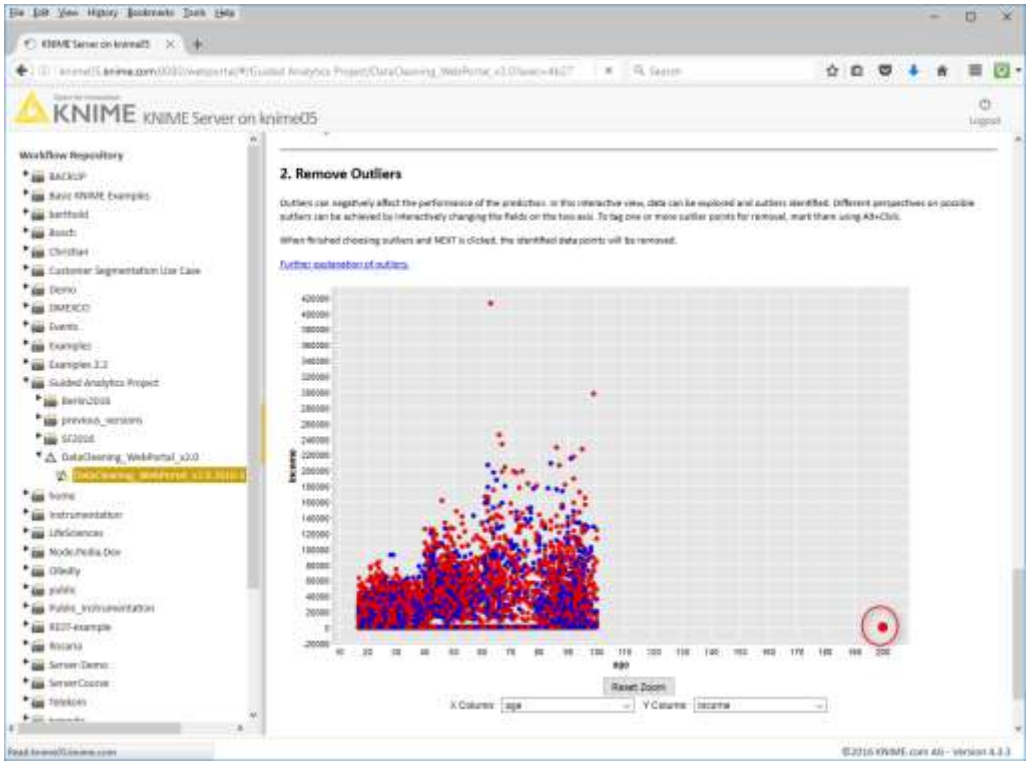

Figure 10.

Scatter Plot displaying age vs. income. A few people declared to be 200 year old. Likely, this information is not true and those data points can be selected for removal.

Step 4. Quality Measure of New Data Set

After cleaning up the data set to remove uninformative data columns, we re-execute the cross-validation procedure and we re-calculate the measure of the data set quality.

The next web page shows the final data set quality, which will hopefully - but not necessarily – be higher; the final number of data columns; and the final number of data rows (figure 11).

In our particular case, the original data set showed an average cross-validation classification error below 1%. Right! Remember the “lawyer assigned” data column? This data column only contained values if lawyer insurance had already been bought. Basically, it was a hidden copy of the target variable. Well, without that copy column, the target values become harder to predict. The cross-validation procedure produces an average error of 43%. Definitely worse, but more realistic.

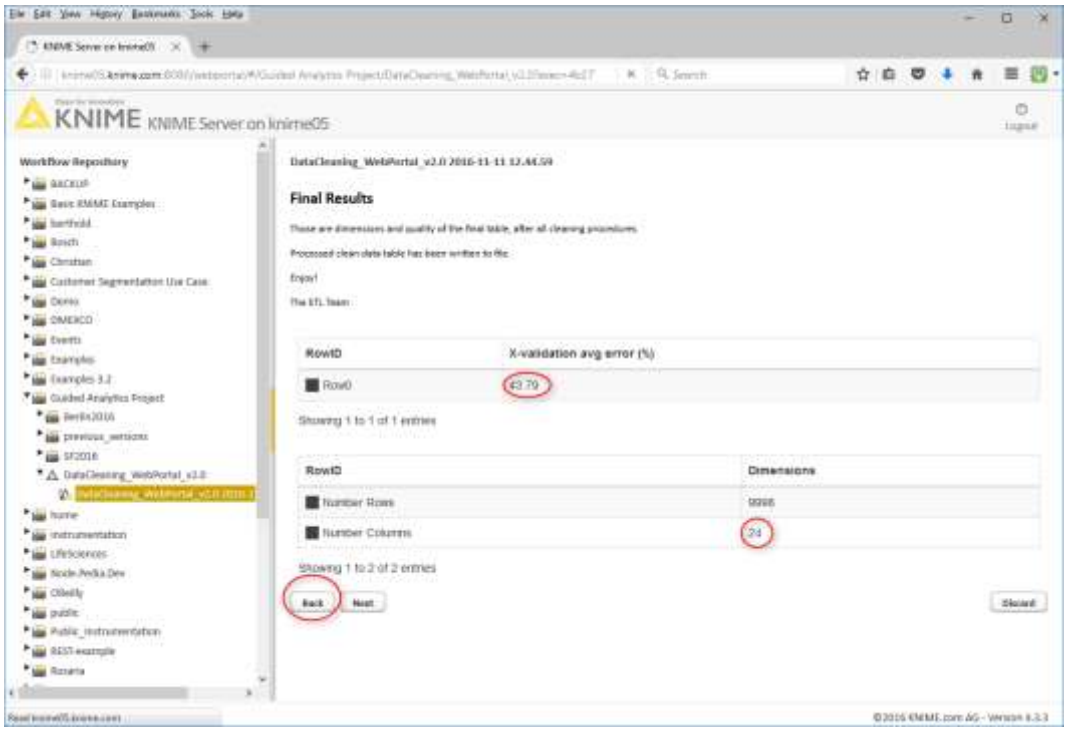

Figure 11.

Measure of quality of final data set. 43% average classification error from the cross-validation procedure. We are very far from the original <1% average error. This was due to an input column with close correlation to the target column.

Clicking Next saves the final data set to a file and brings the workflow execution to end (figure 12).

If the end user is not happy with the results and would like to refine them, the Back button takes him/her back to the column selection page. Here, old columns can be retained and new columns can be selected, for a new data cleaning experiment.

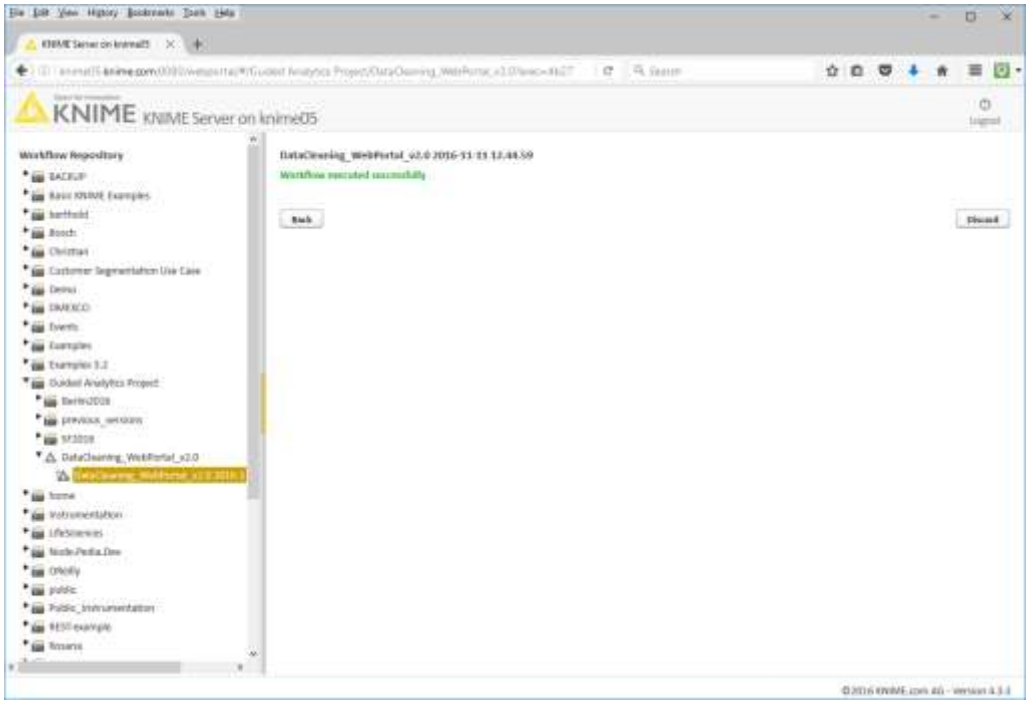

Figure 12.

The End. This is the conclusive web page of the web-guided execution of our workflow.

Data Cleaning through KNIME Workflow

The workflow responsible for those web-guided steps, as described above, is shown in figure 13. There the annotations describe which groups of nodes are responsible for which web-steps. Each group contains at least one wrapped node. Each wrapped node contains at least one Quickform node or one Javascript-based node. Each one of these wrapped nodes generates one of the above web pages.

1. “File Upload” wrapped node generates the web page with the file upload dialog.
2. “Select Target” wrapped node generates the web page with target selection.
3. “Dimensionality Reduction” wrapped node creates the page with the lists of candidate columns for removal.
4. Finally, “Summary” wrapped node creates the last page with the quality measure of the final data set.

The workflow is available for download from the EXAMPLES Server under **50_Applications/ 25_DataCleaning_WebPortal**

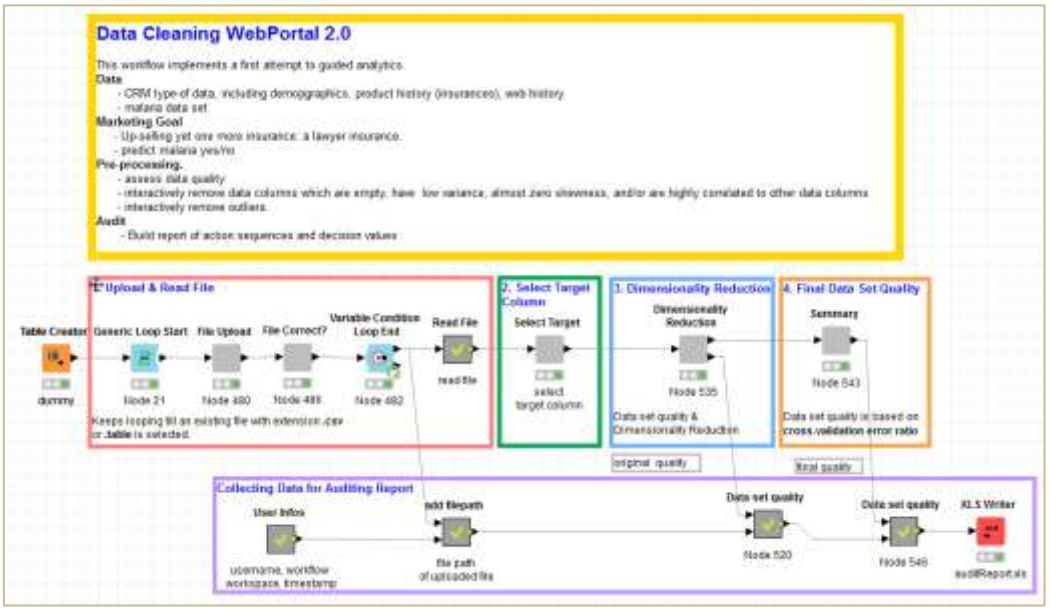

Figure 13.

The full underlying workflow, generating the sequence of web pages from figure 4 to figure 11.

Below is the sub-workflow contained in wrapped node “Dimensionality Reduction”. This node creates the web-step #3, where all data columns are examined in terms of the information they carry.

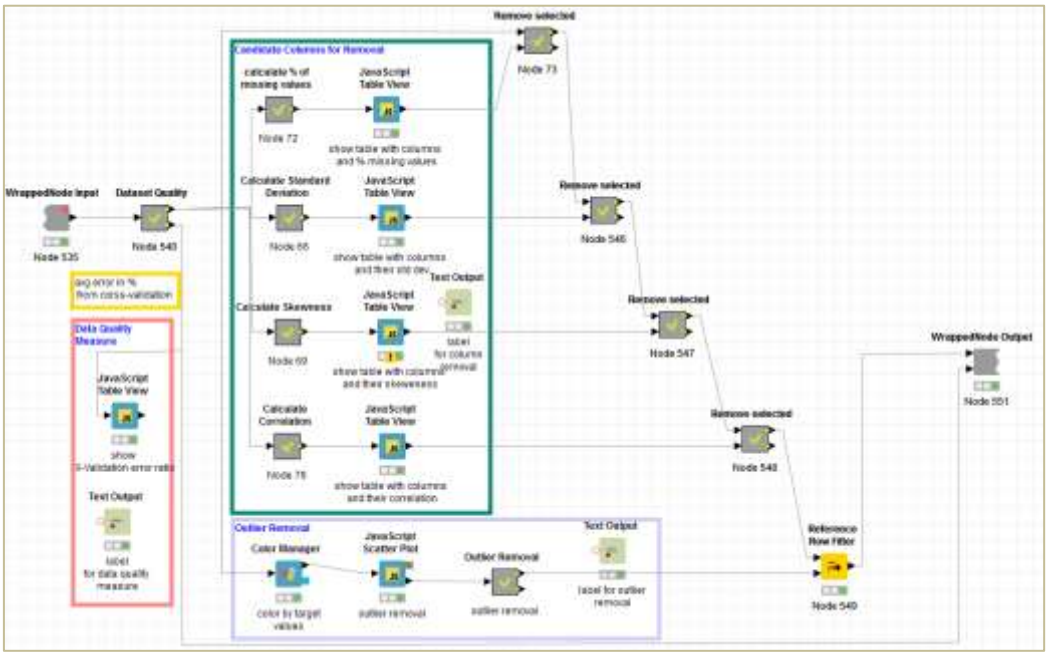

Figure 14.

Sub-workflow in wrapped node named “Dimensionality Reduction” which generates web step #3 (figure 6-10).

The Javascript Table View node in the red frame on the left generates the table with the quality measure of the original data set.

The Javascript Table View nodes in the green frame in the center produce the tables that contain candidate columns for removal.

Finally, the Javascript Scatter Plot node in the purple frame produces the interactive scatter plot on the web page for outlier removal.

The Text Output Quickform node writes text on the web page, either as ASCII text or as HTML formatted text. This node is used to write explanations for what appears on the page.

All those pieces are combined together through a layout tool. The layout tool is activated by one of the last buttons on the right in the tool bar (figure 15). This button opens an editor containing a table-like JSON matrix, with rows and columns. Each node is identified through its nodeID and is located inside a cell uniquely identified by a row and a column of the matrix. The layout button is only active for sub-workflows of open wrapped nodes.

Note. Usually, groups of Javascript nodes come with their own Text Output Quickform node.

Hint. Option “Node” -> “Show Node ID” in the top menu shows the nodeID for each one of the nodes in the wrapped node.

The lower part of the workflow saves all decisions into a table in a database. This is always useful for auditing purposes: to be able to reconstruct exactly all steps and decisions of the cleaning process.

If we pay attention to use *.table files and to not hard-code the data column set in all column-operating nodes, then the workflow should be portable to a different data set.

Of course, if we use a CSV file, the file structure in the File Reader configuration window should be redefined, in the presence of a new file. If we hard-code the column names in some of the column-operating nodes, the workflow will fail for the missing columns.

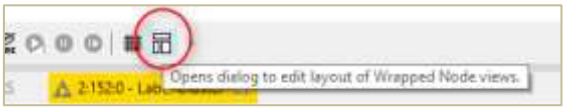

Figure 15 (left).

Layout button in tool bar.

Figure 16 (right).

Table-like JSON structure to layout GUI pieces in a web page from Quickform and Javascript based Visualization nodes.

Let us suppose to have taken extra-care in not hard-coding the column names in any of the nodes. For example, enabling the option “enforce exclusion” on a common set of columns - even an empty one - or by enabling the option “include all columns”, where possible, includes all columns by default, independently of the data set structure.

Let us also suppose that we always use *.table files. A *.table file includes all configuration settings, making the configuration of the Table Reader node independent of the file structure.

Once that has been taken care of, we can now upload the malariahts_data.table file from KNIME WebPortal during workflow execution. This data set contains malaria observations. According to the experts, the target column is Pf3D7_ps_hit = 1(yes) / 0(no).

The sequence of steps 1 to 4 is now repeated on this new file. At step 3, we discover many versions of the primary_readout columns and we discover that most of them are just copies. Removing them improves the final classification performance.

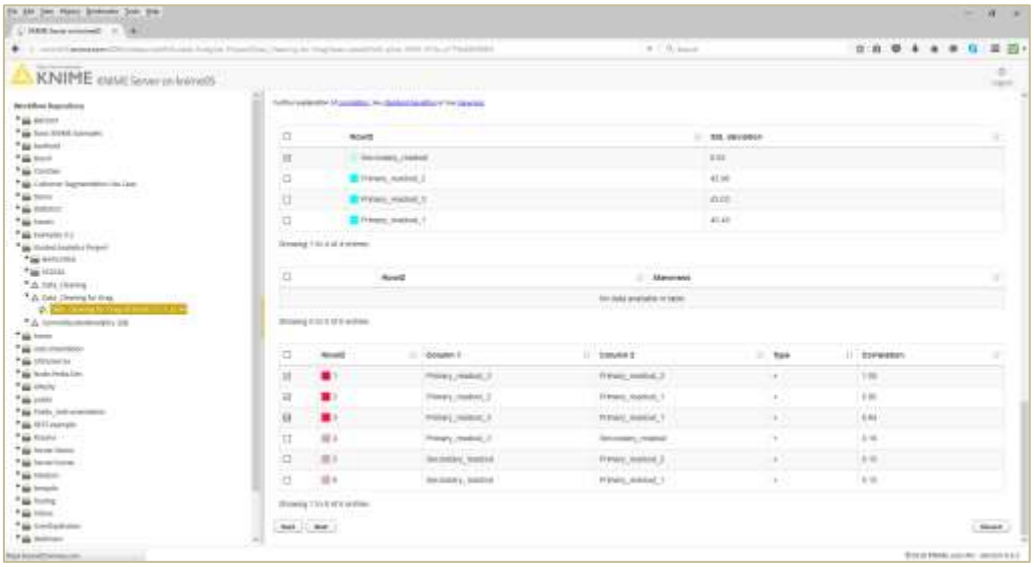

Figure 17.

Skewness and correlation tables when running the workflow on the Malaria dataset. Primary_readout columns are highly correlated.

Conclusions

This whitepaper applies some of the techniques described in the KNIME whitepaper "Seven Techniques for Data Dimensionality Reduction". It applies them and adds an appealing, intuitive, step guided web interface on KNIME WebPortal. This makes the workflow a useful instrument not only in the hands of domain experts, but data analysis beginners, and users, too.

The workflow is available for download from the EXAMPLES Server under **50_Applications/ 25_DataCleaning_WebPortal**.