# Customer Segmentation comfortably from a Web Browser

## Combining Data Science and Business Expertise

**Rosaria Silipo**
Rosaria.Silipo@knime.com

**Christian Albrecht**
Christian.Albrecht@knime.com

## Summary

If you work on sales or CRM analytics, you have for sure dealt with the problem of customer segmentation at least once. If you are a data scientist, you have for sure dealt with the problem of including domain and business experts' knowledge in your workflow at least once.

**This whitepaper addresses these exact two problems:**

- Create a Customer Segmentation analytics heart
- Create a Web User Interface to inject business experts' knowledge into the final results

The customer segmentation analytics heart can be implemented through a series of rules with the Rule Engine node, if we have an expert at hand; or through binning on a particular customer aspect, such as money or loyalty, if we know what aspect is important to our business; or through a clustering technique if we have no knowledge of the business and the data we are analyzing. Since this whitepaper has to be general and apply to a number of different business cases and since the authors have only a limited knowledge on just a few of them, we decided to proceed blindly and use a clustering technique to implement our customer segmentation heart.

Next, in order to unite the distance-based clusters and the experts' knowledge, we built a wizard on the KNIME WebPortal to visually investigate and appropriately label the resulting clusters.

Example data are referring to telco customer behaviors in terms of call usage across the 24 hours. Our business expert then seats comfortably in front of the computer, opens a web browser, visualizes the clusters on a scatter plot one after the other, and based on his knowledge and experience labels each one of them appropriately.

This is indeed the first step to a more complex guided analytics workflow. The idea behind is to allow business experts to interact with the analytics running in the background. Notice that the analytics details are not shown and cannot be changed, but the results are refined through the business expert's contribution.

*The workflow is available for download from the EXAMPLES Server under* **50_Applications/24_Customer_Segmentation_UseCase**

## Table of Contents

## What is Customer Segmentation and Who is Responsible For It?

Customer segmentation has undoubtedly been one of the most implemented applications in data analytics since the birth of customer intelligence and CRM data.

The concept is easy. Group your customers together based on some criteria, such as revenue creation, loyalty, demographics, buying behavior, or any combination of these criteria and more. And, if you, like us, are completely in the dark as to which criteria can be used to identify groups of customers, you can always (blindly) run a clustering procedure.

Indeed, the definition and interpretation of such grouping criteria is more the work of a modern business analyst, who is (or should be) a domain expert. The data scientist can implement either these criteria or a blind clustering procedure.

Data scientists and modern business analysts should then work closely together to achieve and automatize a comprehensive description of the company's groups of customers.

Two problems arise from this conclusion.

1. We need to implement a customer segmentation frame that can accommodate a self-adjusting procedure, such as clustering or a pre-defined set of rules as translated from the business analysts' knowledge.
2. Modern business analysts need an interactive way to inject their knowledge into the customer segmentation frame without ever opening the underlying data processing workflow.

## The Basic Workflow for Customer Segmentation

Our lack of knowledge of the business domain often makes us utterly incapable of defining business rules for meaningful customer segmentation. In this case, we need to resource to a clustering procedure.

## Available Clustering Algorithms

There are many clustering procedures and KNIME Analytics Platform makes most of them available under the category Analytics/Mining/Clustering in the Node Repository panel.

The most commonly known and used is the k-Means algorithm. The k-Means algorithm associates patterns according to the minimum distance criterion and calculates the prototypes of the new clusters as the average of all data points included. The distance used is, in most cases, the Euclidean distance and this generates spherical clusters around their prototypes. In KNIME Analytics Platform the k-Means clustering procedure is implemented by the k-Means node.

Other nodes are available to implement other clustering procedures, such as the nearest neighbors, DBSCAN, hierarchical clustering, SOTA, etc … They all use different similarity measures and aggregation algorithms and end up with differently shaped clusters on the same data.

Once the clustering algorithm/node has been chosen, we start to put together the workflow.

## Data

We are using the same telco data set used for the churn prediction use case (https://www.knime.org/knime-applications/churn-prediction) which contains 2 files. One file has the contract data, while the other contains the operational (cell phone usage) data for each customer. Each record is uniquely identified by the cellular number and area code. Both files are read, one by a File Reader and one by an Excel Reader node, and then joined together to obtain a summary record for each customer.

## Pre-processing

Clustering algorithms work on distances, i.e. they take into account only numerical features. In order to expand the number of features to be included in the clustering or, contrarily, to exclude some features from the clustering, we need to work on their type.

a. ***Type Conversions: Number to String and String to Number***

   Sometimes, string type columns contain just numbers: either pure numbers saved in a string-type column; or string values easily transformable into numbers, such as “35years”. In this case, it is

very easy to reduce the value to a number with a String to Number node, optionally preceded by a String Manipulation node.

On the other hand, there will sometimes be numerical fields, which should not be used for clustering as they carry no information about the customer. An example are the Customer ID labels. While Customer IDs sometimes happen to be just numbers, they are uniquely generated by a system and usually have nothing to do with the customer demographics, the behavior, or any other group of customer features. In this case, such columns should be excluded from the clustering procedure. We simply convert them into String columns with the Number To String node.

***b. Discretization***

Another easy way to transform a nominal column into a numerical column is to use discretization. Basically, we associate a number to each nominal value. An example is the equivalence between a star ranking system and the corresponding judgements: 5 stars = “very good”, 1 star = “very bad”. The 5 text evaluations can be transformed into 1, 2, 3, 4, 5 number of stars.

***Normalization***

Again, clustering algorithms are based on distances calculated across numerical data values. The range of the columns thus plays a major role in making one data column more influential than the other. For example, data column “age” with range [1,100] will outweigh data column score in range [0,1]. To avoid that, all numerical data columns have to be normalized to fall into the same range, usually [0,1].

After string manipulation, conversion, discretization, and normalization, we apply the k-Means algorithm to produce a predefined number of data clusters. Let’s default to 10 clusters.

## Final Workflow for Customer Segmentation

The basic workflow for customer segmentation consists of only 3 steps: data reading, data pre-processing, and k-Means clustering.

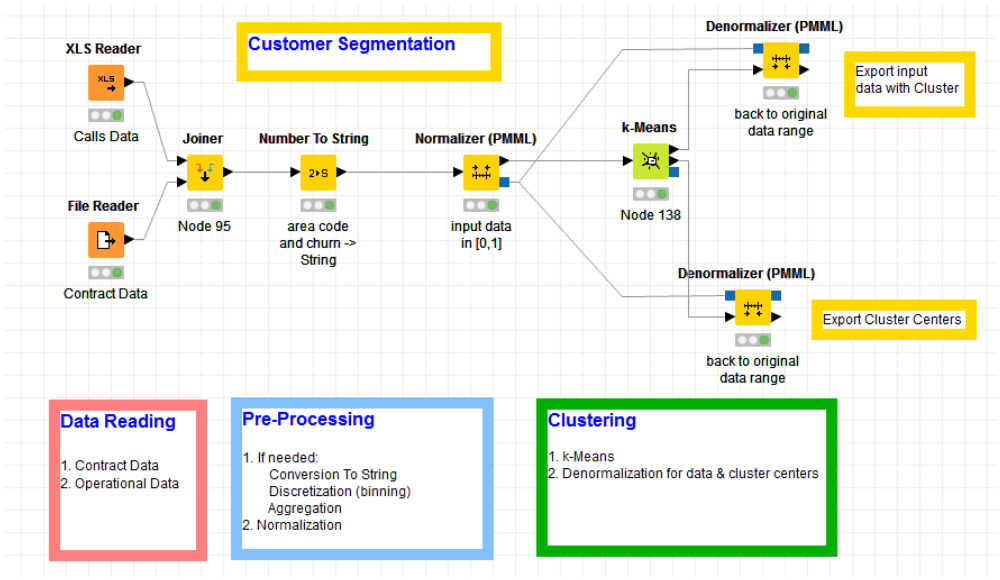



**Figure 1.**

*Workflow for Basic Customer Segmentation with Clustering.*

The heart of the workflow is the k-Means node. It is the k-Means node that groups the data and produces clusters and cluster representative prototypes.

**Note.** Such a grouping can be changed easily by using a binning procedure (Auto-Binner, Numeric Binner, and Binner(Dictionary) nodes) or even a manually defined set of rules (Rule Engine node).

This workflow generates two data outputs: the cluster centers and the original data rows with cluster labels (see last column in Figure 2).

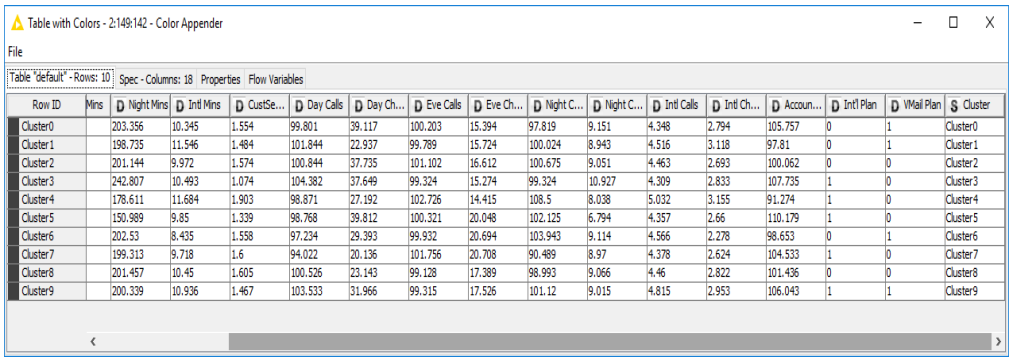


Table with Colors - 2:149:142 - Color Appender

File

Table "default" - Rows: 10 | Spec - Columns: 18 | Properties | Flow Variables

| Row ID | Mins | Night Mins | Intl Mins | CustSe... | Day Calls | Day Ch... | Eve Calls | Eve Ch... | Night C... | Night C... | Intl Calls | Intl Ch... | Accoun... | Int'l Plan | VMail Plan | Cluster |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Cluster0 | | 203.356 | 10.345 | 1.554 | 99.801 | 39.117 | 100.203 | 15.394 | 97.819 | 9.151 | 4.348 | 2.794 | 105.757 | 0 | 1 | Cluster0 |
| Cluster1 | | 198.735 | 11.546 | 1.484 | 101.844 | 22.937 | 99.789 | 15.724 | 100.024 | 8.943 | 4.516 | 3.118 | 97.81 | 0 | 1 | Cluster1 |
| Cluster2 | | 201.144 | 9.972 | 1.574 | 100.844 | 37.735 | 101.102 | 16.612 | 100.675 | 9.051 | 4.463 | 2.693 | 100.062 | 0 | 0 | Cluster2 |
| Cluster3 | | 242.807 | 10.493 | 1.074 | 104.382 | 37.649 | 99.324 | 15.274 | 99.324 | 10.927 | 4.309 | 2.833 | 107.735 | 1 | 0 | Cluster3 |
| Cluster4 | | 178.611 | 11.684 | 1.903 | 98.871 | 27.192 | 102.726 | 14.415 | 108.5 | 8.038 | 5.032 | 3.155 | 91.274 | 1 | 0 | Cluster4 |
| Cluster5 | | 150.989 | 9.85 | 1.339 | 98.768 | 39.812 | 100.321 | 20.048 | 102.125 | 6.794 | 4.357 | 2.66 | 110.179 | 1 | 0 | Cluster5 |
| Cluster6 | | 202.53 | 8.435 | 1.558 | 97.234 | 29.393 | 99.932 | 20.694 | 103.943 | 9.114 | 4.566 | 2.278 | 98.653 | 0 | 1 | Cluster6 |
| Cluster7 | | 199.313 | 9.718 | 1.6 | 94.022 | 20.136 | 101.756 | 20.708 | 90.489 | 8.97 | 4.378 | 2.624 | 104.533 | 1 | 0 | Cluster7 |
| Cluster8 | | 201.457 | 10.45 | 1.605 | 100.526 | 23.143 | 99.128 | 17.389 | 98.993 | 9.066 | 4.46 | 2.822 | 101.436 | 0 | 0 | Cluster8 |
| Cluster9 | | 200.339 | 10.936 | 1.467 | 103.533 | 31.966 | 99.315 | 17.526 | 101.12 | 9.015 | 4.815 | 2.953 | 106.043 | 1 | 1 | Cluster9 |

**Figure 2.**

*Output Data from Workflow in Figure 1. Notice the Cluster Assignment in the last Column to the right.*

## Refining Customer Segments with Experts' Knowledge

### Scatter Plot for all Data Points colored by Cluster

The simple cluster prototype, however, is not very generous with information. Even a domain expert cannot gather much knowledge from the simple cluster average values.

A graphical exploration of the customer distribution, for example through a scatter plot, could help the modern business analyst in the task of identifying and appropriately labelling each cluster. The scatter plot below shows used-evening-minutes vs. used-daily-minutes for all customers in the dataset, colored by cluster.

This scatter plot has been built using the Javascript Scatter Plot node, which offers a plot view and allows to introduce some degree of interactivity to be introduced into the plot (through the tab "View Controls" in its configuration dialog). Interactivity, for example, could amount to editable title, data point selection and export, or changing the data columns to display in the plot.

The 10 colors for the 10 clusters in the plot come from a Color Manager node preceding the scatter plot node (Figure 3).

Messy, huh? As powerful as the interactive view can be, we cannot rescue much information about each single cluster, besides the purple and the green ones in the front position.


**Figure 3.**

*Scatter Plot Day Minutes vs. Evening Minutes on full Data Set. Data Points are colored by Cluster.*

## One Scatter Plot for each Cluster: Looping

Let's change approach then. Let's visualize the same scatter plot for only one cluster at a time, cluster after cluster. The scatter plot drawing node is still the Javascript Scatter Plot node. However, now we loop on all clusters, using a Group Loop Start and a Loop End (2 Ports) node. At each iteration, we visualize the scatter plot of a different cluster.

Here the data points belonging to the current cluster are in color against a gray background formed by all the remaining data points. Below is the scatter plot for Cluster 0 in red on top of all other points in gray.

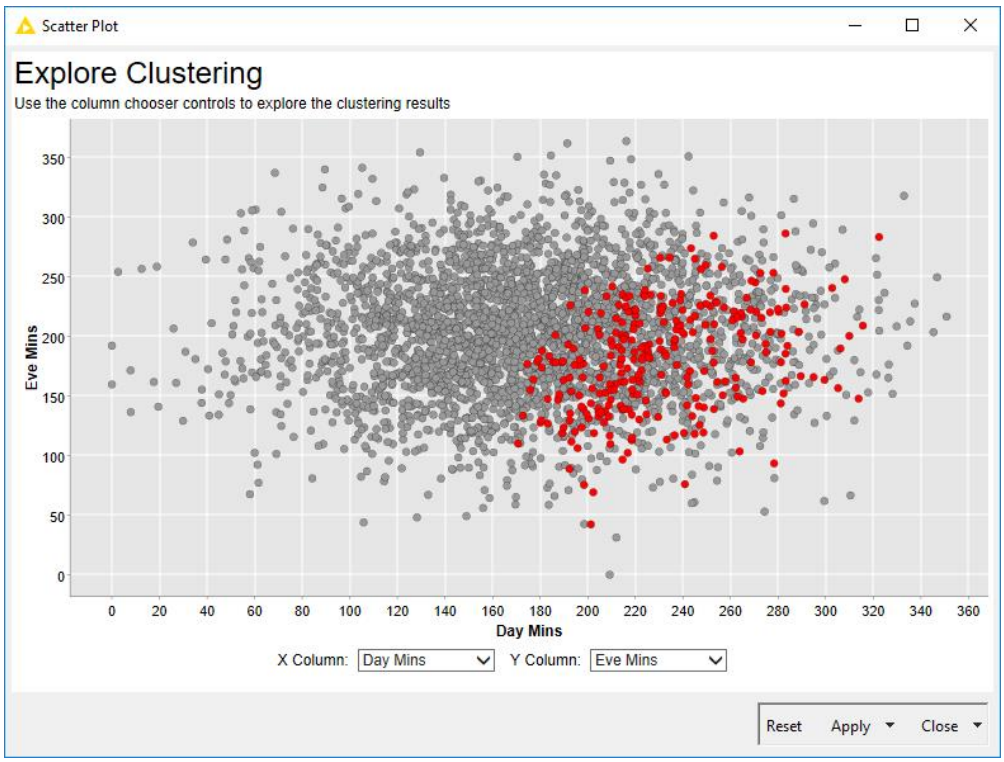



**Figure 4.**

*Scatter Plot of Day Minutes vs. Evening Minutes for one Cluster only. Only the Data Points belonging to the selected Cluster are colored. The remaining Points are gray in the background.*

Visual inspection is now much easier: cluster 0 covers the daily callers. We could probably label this cluster “Daily Callers” or something like that. Actually, let’s do that.

Let’s introduce a labelling function for the business analyst to give a name or write notes for what he/she sees.

That is, let’s introduce a String Input Quickform node to insert the cluster name in a text box.

## Comfortably from a Web Browser

Modern business Analysts are domain experts, but not necessarily data analytics or even KNIME experts. Life would be much easier for everybody involved if such visual inspection and labeling could be conducted comfortably from a web browser without having to deal with workflows or clustering techniques or similar things. Yes, life would be much easier!

Let’s move the whole workflow to the KNIME Server. The advantage of the Javascript Scatter Plot node, as all Javascript based visualization

nodes, is that what you see in the KNIME view is what you see in a web browser view. Exactly the same, down to the smallest interactive control.

Let's now add a few explanation texts that will guide the business analysts through the various steps of the analysis … et voilà the workflow has become a fully guided web browser based application.

If you are the domain expert business analyst, just open a web browser, log in to the KNIME Server (you will need credentials for that), select this customer segmentation application, and follow the steps.

Step 1. Start the workflow

Step 2. Select the parameters

Step 3. Explore the data

Step 4. Label the clusters, one by one

Step 5. Review the analysis including the assigned labels

**Step 1. Start the workflow.**

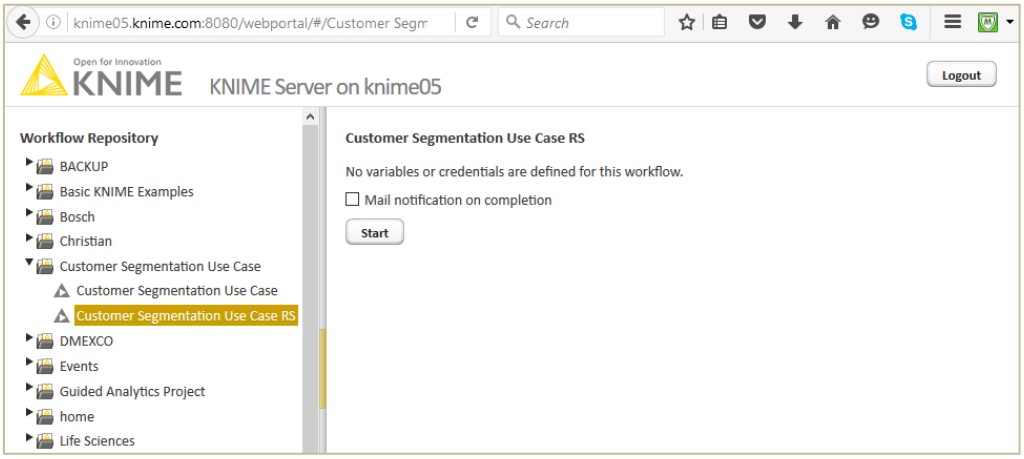



**Figure 5.**

*Step 1: Start the Workflow.*

**Step 2. Select the parameters**

- Select a reasonable number of clusters. Usually a reasonable number will be between 3 and 10: not too big otherwise the loop lasts forever and not too small to make sure we cover all of the groups of customers. However, for a sensible selection of the number of clusters, different businesses follow different rules.

- Select the data columns to use for the analysis. In this data set we have a few features that are all related with each other: minutes spent on the phone, # of calls, and related charges. In Figure 6, our modern business analyst has chosen only the minutes, but he could have chosen either more or less features and explored the different results. This experimental interactive character is exactly the goal of this application.

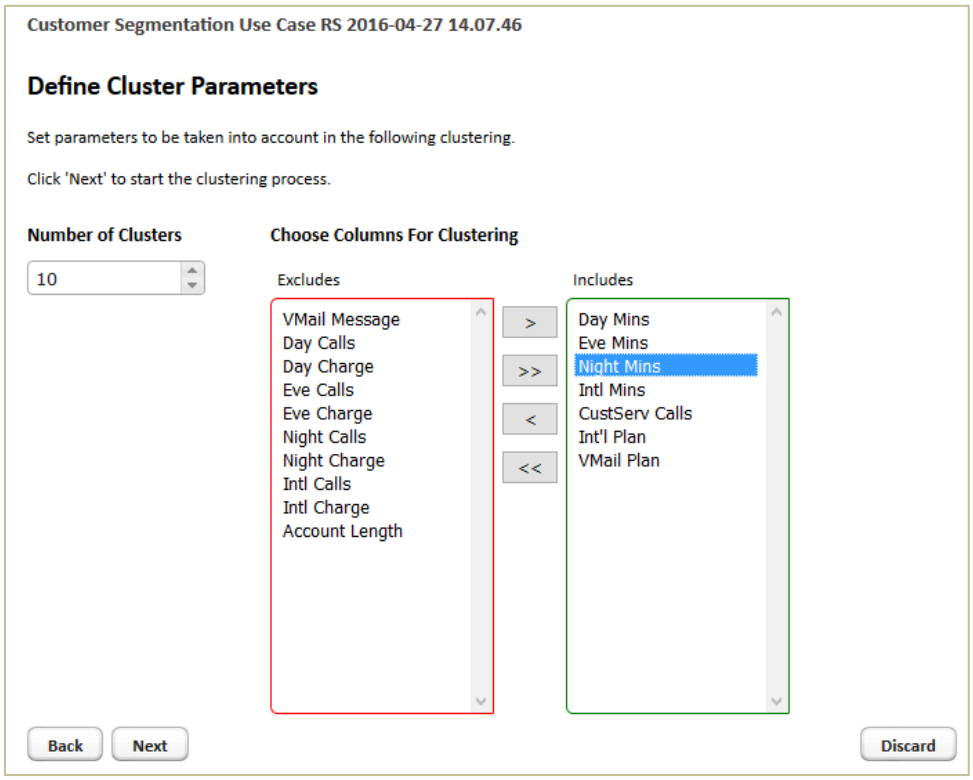



**Figure 6.**

*Step 2: Select Number of Clusters and Numerical Features for Clustering.*

### Step 3. Explore the data

The first interactive visualization frame covers all clusters. This is the moment to abstract yourself from the tiny details, see the big picture, maybe even change perspective from time to time, and choose different feature for the scatter plot.

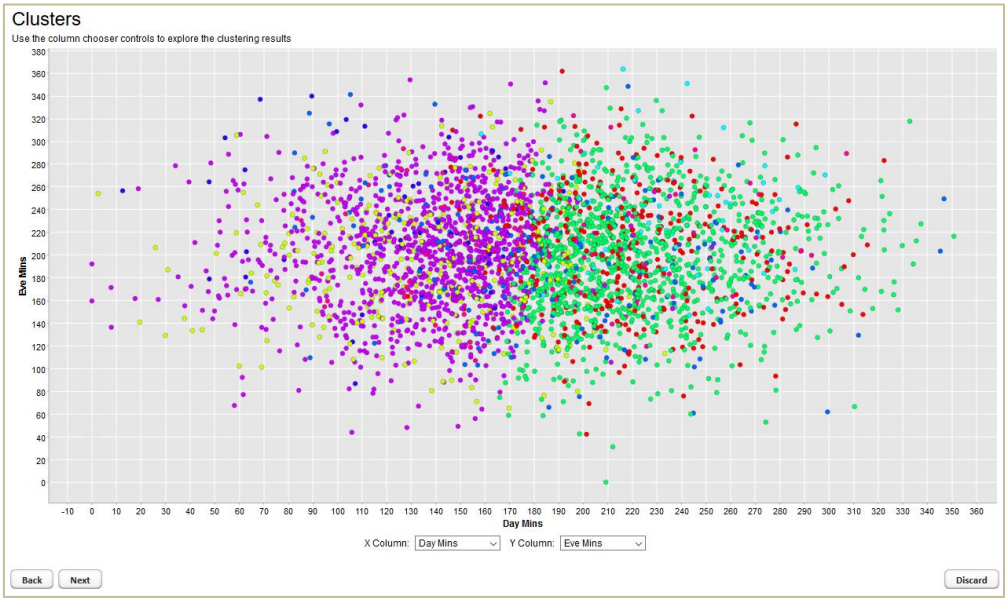



**Figure 7.**

*Step 3: Explore all Data through a Scatter Plot (like in Figure 3 but this time on a Web Browser).*

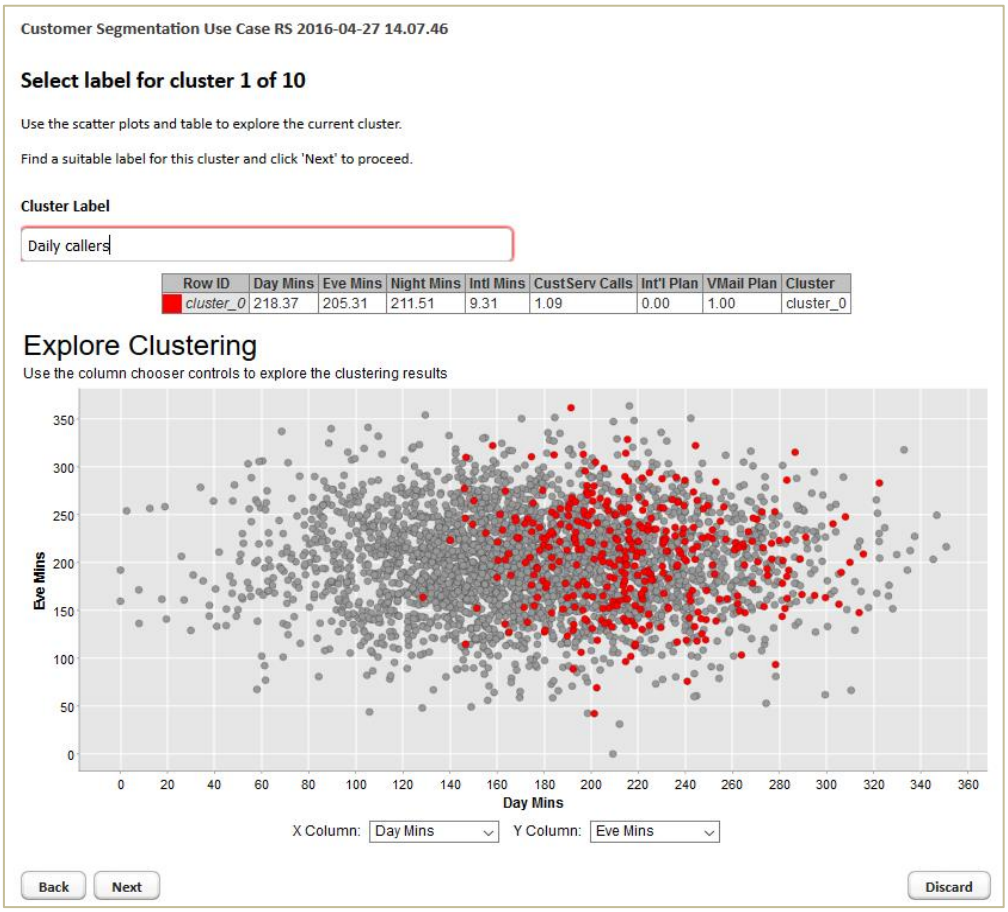


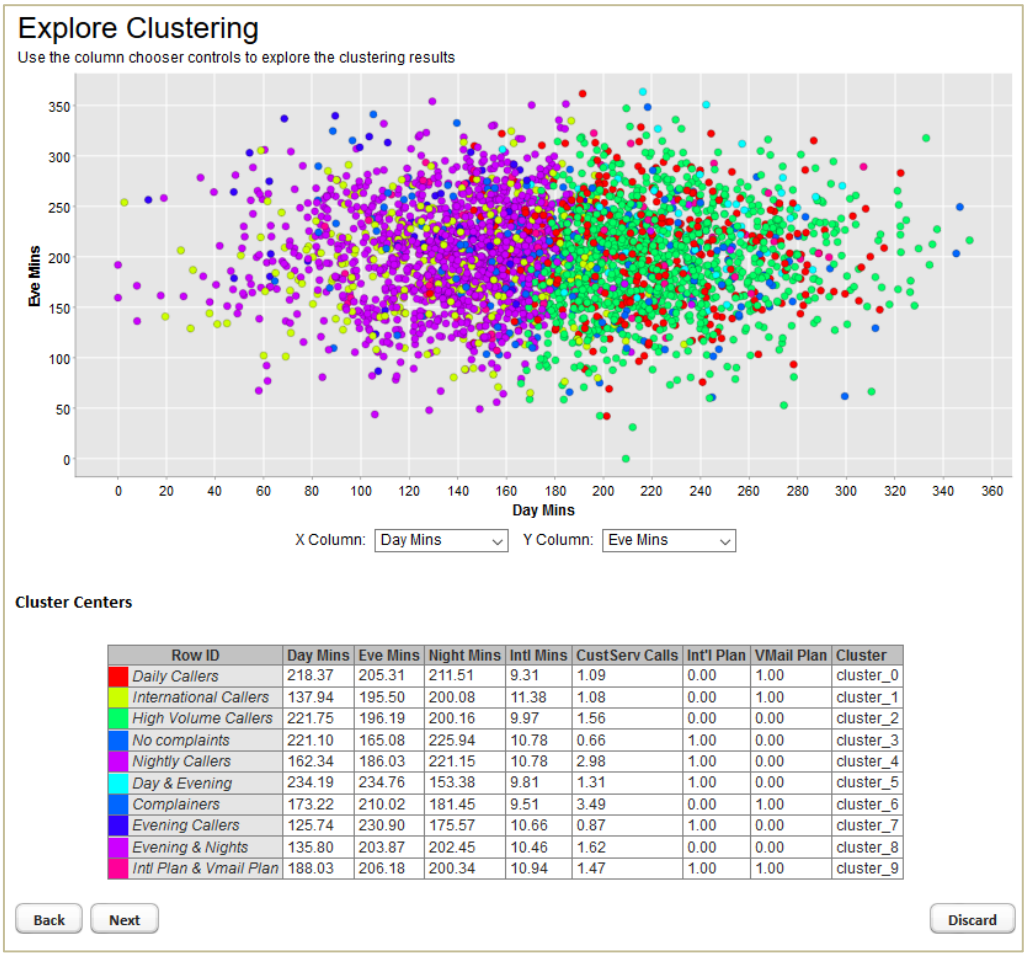

**Figure 8.**

*Step 4: For each Cluster inspect Data Point Positions and label the current Cluster accordingly.*

### Step 4. Label the clusters one by one

Let's go into detail now and let's explore the clusters one by one. For each cluster, first let's inspect its data points in a scatter plot; then let's create and assign a meaningful label.

Figure 8 shows again cluster 0 with its daily callers.




| Row ID | Day Mins | Eve Mins | Night Mins | Intl Mins | CustServ Calls | Int'l Plan | VMail Plan | Cluster |
|---|---|---|---|---|---|---|---|---|
| Daily Callers | 218.37 | 205.31 | 211.51 | 9.31 | 1.09 | 0.00 | 1.00 | cluster_0 |
| International Callers | 137.94 | 195.50 | 200.08 | 11.38 | 1.08 | 0.00 | 1.00 | cluster_1 |
| High Volume Callers | 221.75 | 196.19 | 200.16 | 9.97 | 1.56 | 0.00 | 0.00 | cluster_2 |
| No complaints | 221.10 | 165.08 | 225.94 | 10.78 | 0.66 | 1.00 | 0.00 | cluster_3 |
| Nightly Callers | 162.34 | 186.03 | 221.15 | 10.78 | 2.98 | 1.00 | 0.00 | cluster_4 |
| Day & Evening | 234.19 | 234.76 | 153.38 | 9.81 | 1.31 | 1.00 | 0.00 | cluster_5 |
| Complainers | 173.22 | 210.02 | 181.45 | 9.51 | 3.49 | 0.00 | 1.00 | cluster_6 |
| Evening Callers | 125.74 | 230.90 | 175.57 | 10.66 | 0.87 | 1.00 | 0.00 | cluster_7 |
| Evening & Nights | 135.80 | 203.87 | 202.45 | 10.46 | 1.62 | 0.00 | 0.00 | cluster_8 |
| Intl Plan & Vmail Plan | 188.03 | 206.18 | 200.34 | 10.94 | 1.47 | 1.00 | 1.00 | cluster_9 |

**Figure 9.**

*Step 5: Summary of all Clusters with the assigned Labels.*

### Step 5. Final Summary with assigned Labels

After inspecting and labelling all clusters, the business analyst reaches a final summary page where all clusters, their assigned labels, and their cluster centers are shown and saved.

## The Underlying Workflow

There is, of course, a workflow running behind this whole browser-based customer segmentation wizard!

The basic workflow shown in Figure 1 becomes now the first half on the left of this new workflow (Figure 10). This first half also includes a wrapped node, named “Define Cluster Parameters”, with Quickform nodes for the selection of the number of clusters and of the numerical features to be used for clustering.

The second half of the workflow, on the right, implements the browser-based visualization. The first wrapped node, named “Display Cluster Result”, produces a number of scatter plot to visualize the whole data set, where data points are colored by cluster.

Then there is the loop. Do you recognize it? It loops on all clusters, visualizes many scatter plots like the one in Figure 8, and allows for labeling. The last wrapped node on the right shows the summary page (Figure 9) and writes the final results to a file.

*The workflow is available for download from the EXAMPLES Server under* **50_Applications/24_Customer_Segmentation_UseCase**

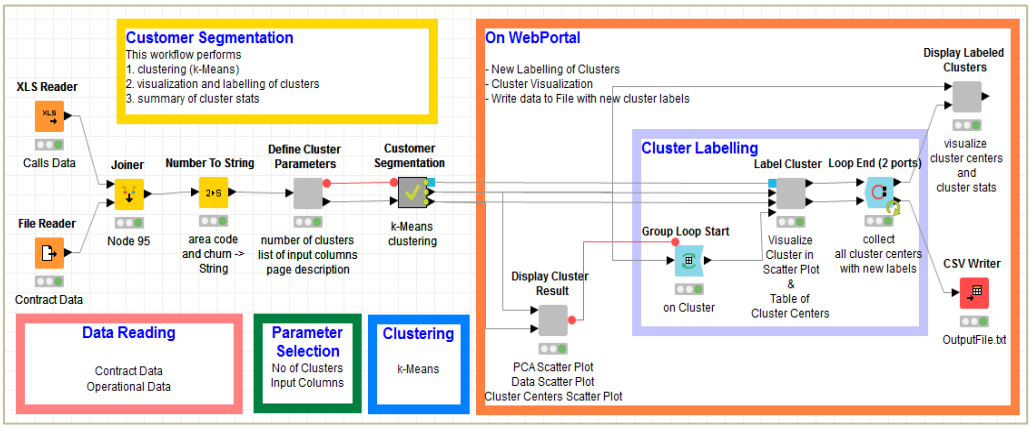



**Figure 10.**

*The full underlying Workflow, performing a Customer Segmentation through Clustering and allowing for inspection and labelling of each Cluster from a Web Browser.*

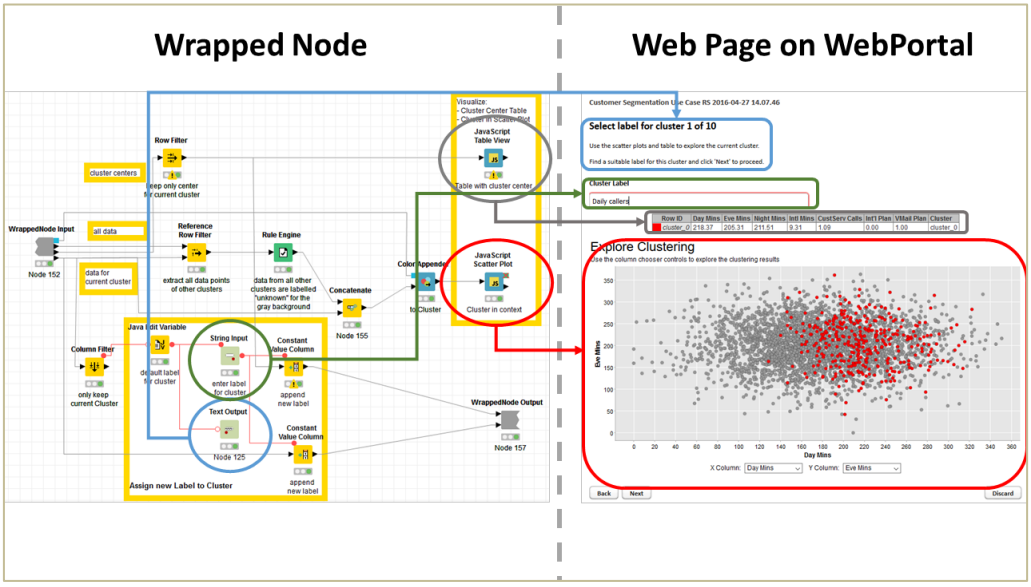



**Figure 11.**

*From Wrapped Node to Web Page. Content of Wrapped Node named "Label Cluster" on the left and resulting Web Page on WebPortal on the right.*

## From Wrapped Node to Web Page

All wrapped nodes have a similar functionality. That is to contain a number of nodes to:

a. Clean up the workflow appearance
b. Implement isolated specific tasks
c. Build a GUI on a web page by combining Quickform and Javascript based Visualization nodes

The first two tasks are common to the wrapped nodes and the metanodes. The last item is really only specific to wrapped nodes. Let's check it out, for example, for the wrapped node named "Label Cluster" inside the loop in Figure 10.

This wrapped node contains two Quickform nodes (String Input and Text Output) and two Javascript based nodes (Javascript Table View and Javascript Scatter Plot). Let's take Figure 8 as a reference and let's see how these nodes relate to the different UI pieces on the web page.

- The Text Output node creates the top explanation text
- The String Input node creates the text box for the cluster labelling, right below the explanation text
- The Javascript Table Viewer node creates the one-row table with the cluster center features
- The Javascript Scatter Plot node creates the scatter plot with the colorful cluster points against the gray ones as background

All these User Interface 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 12). This button opens an editor containing a table-like JSON structure, with rows and columns. Each node is identified through its nodeID and is located inside a cell uniquely identified by one row and one column.

The layout button is only active for sub-workflows of open wrapped nodes.

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

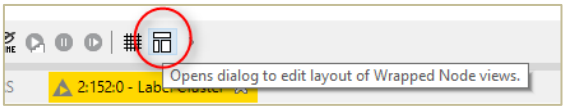



**Figure 12.**

*Layout Button in Tool Bar.*


**Figure 13.**

*Table-like JSON structure to layout GUI pieces in a Web Page from Quickform and Javascript based Visualization Nodes.*

## Conclusions

This whitepaper addresses two problems: the analytics problem of building a customer segmentation workflow and the team cooperation problem of including other experts' knowledge into the data analytics part.

These two parts, customer segmentation and segment visualization, are completely separated in the workflow built for this application. This allows for better maintenance, since different components can be changed without affecting the rest of the workflow. For example, the customer segmentation was implemented through a clustering technique for lack of better business knowledge. However, the clustering node can be easily changed with a binner node or a Rule Engine node, if such knowledge becomes available.

Interaction between data analysts and domain and business experts is implemented through a sequence of web pages: a wizard on the KNIME WebPortal. In this way, our business experts just open the web browser and step by step refine the customer segmentation results based on their own knowledge and experience.

*The workflow is available for download from the EXAMPLES Server under* **50_Applications/24_Customer_Segmentation_UseCase**.