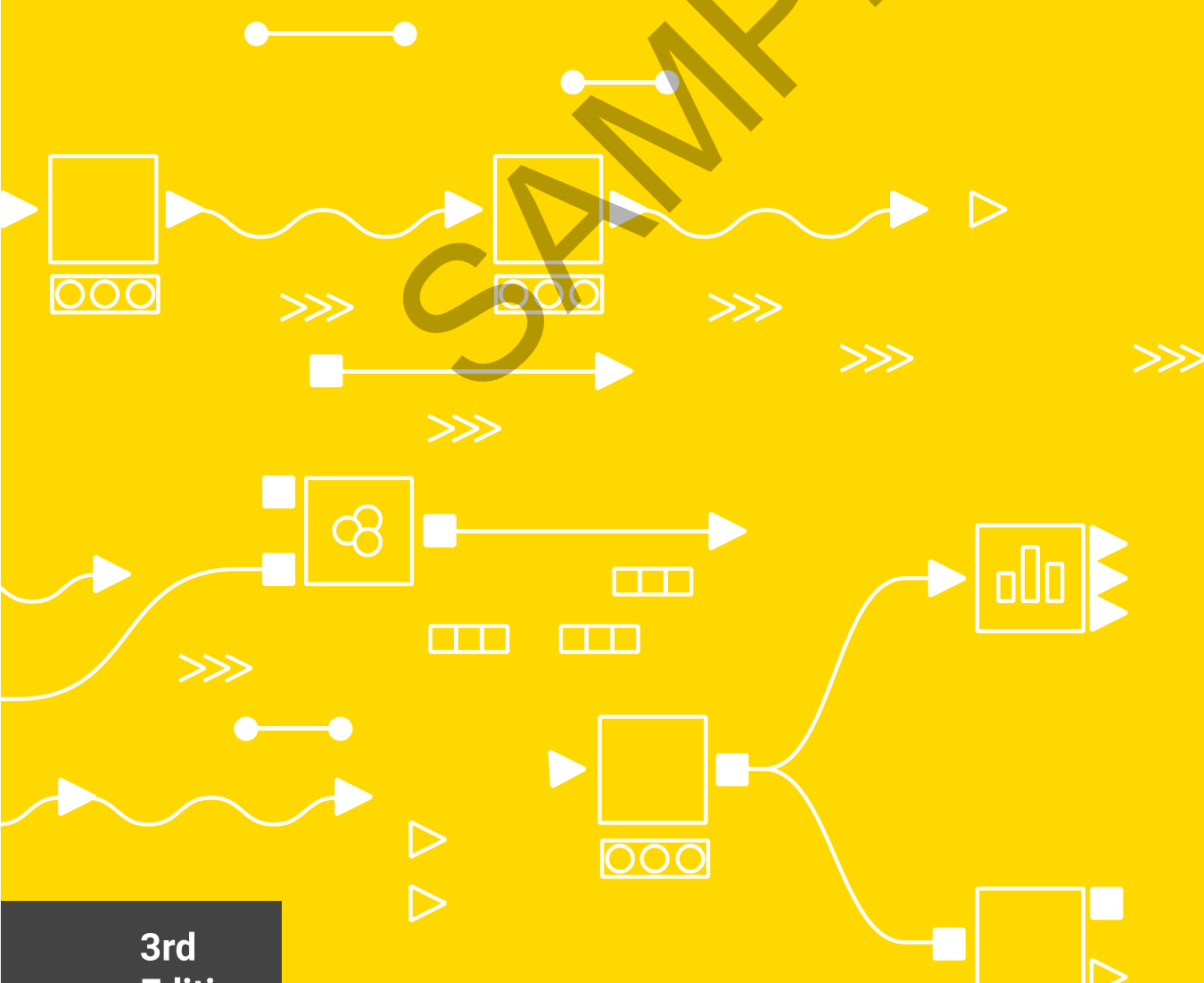


A Collection of Case Studies

# Practicing Data Science



**3rd  
Edition**

Copyright©2021 by KNIME Press

All rights reserved. This publication is protected by copyright, and permission must be obtained from the publisher prior to any prohibited reproduction, storage in a retrieval system, or transmission in any form or by any means, electronic, mechanical, photocopying, recording or likewise.

This book has been updated for **KNIME 4.5**.

For information regarding permissions and sales, write to:

KNIME Press  
Hardturmstrasse 66  
8005 Zurich  
Switzerland

[knimepress@knime.com](mailto:knimepress@knime.com)

SAMPLE

# Table of Contents

Introduction.....	9
Chapter 1. Customer Intelligence .....	10
1.1. Churn Prediction.....	10
What You Need.....	10
The Workflow to Train the Model.....	10
Which Model?.....	10
Deployment .....	11
1.2. Customer Segmentation .....	13
Customer Segmentation Strategies.....	13
What You Need.....	13
Basic Workflow for Customer Segmentation with Clustering Procedure.....	14
Model Deployment .....	15
Refining Customer Segments with Business Knowledge by Domain Experts .....	15
Chapter 2. Retail and Supply Chain.....	18
2.1. Market Basket Analysis with the Apriori Algorithm .....	18
What You Need.....	18
Workflow to Build Association Rules with the Apriori Algorithm.....	18
Deployment .....	20
References .....	21
2.2. Movie Recommendations with Spark Collaborative Filtering.....	23
What You Need.....	23
A general dataset with movie ratings by users .....	23
Movie preferences by current user.....	23
A Spark Context.....	24
Workflow to Build the Recommendation Engine with Collaborative Filtering.....	26
Deployment .....	27
References .....	29
Chapter 3. Banking and Insurance.....	30
3.1. Credit Risk Assessment .....	30
What You Need.....	30
Workflow to Predict Delinquency .....	30
Data Preprocessing .....	31
Linear Correlation Map.....	32
SMOTE Algorithm .....	33
Model Evaluation .....	34
Deployment .....	35

References .....	35
Chapter 4. Web & Social Media.....	37
4.1. Document Classification: Spam vs. Ham .....	37
What You Need.....	37
Workflow .....	37
Deployment .....	38
References .....	39
4.2. Topic Detection: What Is It All About?.....	40
Text Summarization .....	40
Topic Detection .....	41
What You Need.....	42
The Workflow .....	42
Deployment .....	43
References .....	43
4.3. Sentiment Analysis: What's With the Tone?.....	44
Introduction .....	44
Sentiment Analysis: The Techniques .....	44
What You Need.....	44
The Workflow .....	44
NLP based Sentiment Analysis .....	44
ML based Sentiment Analysis .....	46
Deployment .....	46
References .....	47
4.4. Find the Influencers.....	48
Introduction .....	48
The Workflow .....	49
Data Access.....	49
The Matrix of Nodes and Interactions.....	49
Drawing the Chord Plot.....	50
More Formal Network Analysis Technique .....	52
Conclusion.....	52
4.5. Sentiment & Influencers .....	53
Introduction .....	53
What You Need.....	53
The Workflow .....	54
Influence Scores.....	54
Sentiment Analysis.....	55

Putting It All Together .....	56
So, How Did We Do? .....	58
4.6 Digging up Hillary Clinton’s Past: An Interactive Tour of her Email Dataset .....	59
Hillary Clinton’s email dataset.....	59
Questions about Hillary Clinton’s emails .....	59
Training Workflow - Workflow Development Stage 1: .....	60
Deployment Workflow - Workflow Development Stage 2: .....	61
Our own conclusions.....	63
Chapter 5. Web Analytics.....	64
5.1. ClickStream Analysis.....	64
Introduction .....	64
What You Need.....	64
The Workflow .....	66
Preprocessing .....	66
Data Preprocessing .....	66
Data Preprocessing for Visualization.....	67
Visualization.....	69
What We Found .....	71
User Activity According to Age and Gender .....	71
Category Popularity during the Week .....	72
Purchases During the Day and Week .....	74
Click Patterns.....	75
Summary .....	75
Chapter 6. IoT.....	76
6.1. Bike Restocking Alert with Minimum Set of Input Features .....	76
What You Need.....	76
Training Workflow .....	77
Data Preprocessing .....	77
Backward Feature Elimination.....	78
Model Training and Evaluation .....	79
Deployment Workflow .....	80
6.2. Taxi Demand Prediction using Random Forest on Spark .....	82
Introduction .....	82
What You Need.....	82
Preprocessing .....	82
Data Exploration .....	83
Line Plot.....	83

Auto-correlation .....	84
The Training Workflow .....	86
Testing the Model .....	87
The Deployment Workflow .....	88
Summary .....	89
References .....	90
6.3. Anomaly Detection .....	90
What You Need .....	92
Training 313 AR Models .....	93
Deployment .....	94
Testing Results .....	96
Chapter 7. Life Sciences .....	97
7.1. DNA Sequence Similarity Search with BLAST .....	97
What You Need .....	97
Analyzing 270,000 Year Old DNA .....	97
The Workflow .....	98
Handling Asynchronous REST Operations .....	98
BLAST Result .....	99
Deployment .....	100
7.2 Automatic Tagging of Disease Names in Biomedical Literature .....	101
Introduction .....	101
What We Need .....	101
The Workflow .....	102
1. Dictionary and Corpus Creation .....	102
Dictionary creation (Disease Names) .....	102
Corpus creation .....	102
2. Model Training and Evaluation .....	103
Comparison with input dictionary .....	104
3. Co-occurrence of Tagged Disease Names .....	106
Co-occurrence network .....	106
Subgraph .....	106
Summary .....	107
Deployment .....	108
7.3 Creating and Deploying a Self-testing Prediction Service .....	109
What You Need .....	109
The Prediction Workflow .....	109
Preparing the Workflow for Testing .....	111

Deployment - Making it a Web Service.....	113
Conclusion.....	115
Chapter 8. Cybersecurity.....	116
8.1. Fraud Detection.....	116
What You Need.....	116
The Workflow.....	117
Reading.....	117
Partitioning.....	117
Training the Model.....	117
Evaluating the Model on a Class Unbalanced Test Set.....	118
Find a Better Prediction Threshold.....	118
Deployment Workflow and Real Time Performance.....	118
8.2. Fraud Detection Using a Neural Auto-encoder.....	120
Introduction.....	120
What You Need.....	120
Credit Card Transactions Dataset.....	120
The Auto-encoder.....	120
The Anomaly Detection Rule.....	121
The KNIME Keras Deep Learning Extension.....	122
Installation.....	122
Training the Auto-encoder.....	122
Data Preprocessing.....	122
Building the Auto-encoder Architecture.....	123
Training & Testing the Auto-encoder.....	124
Optimizing the Threshold.....	124
The Final Workflow.....	125
Deployment.....	126
Conclusions.....	126
Chapter 9. Text Generation.....	128
9.1. Neural Machine Translation with RNN.....	128
What You Need.....	128
The Encoder-Decoder RNN Structure.....	129
The Training Workflow.....	129
Defining the Network Structure.....	130
Text Preprocessing and Encoding.....	132
Training and Editing the Network.....	132
The Deployment Workflow.....	133

Conclusion..... 134

References ..... 135

9.2. Product Naming with Deep Learning for Marketers and Retailers – Keras Inspired..... 136

What You Need..... 136

The Training Workflow ..... 137

Defining the Network Structure ..... 138

    The Basic Structure ..... 138

    Introducing Temperature..... 138

    The Dropout Layer ..... 139

Preprocessing and Encoding ..... 139

Training the Network..... 139

The Deployment Workflow ..... 140

Summary ..... 141

References ..... 141

SAMPLE



# Introduction

We could start a philosophical discussion about what data science is. But that is not that kind of book. This book is a collection of experiences in data science projects.

There are many declinations of data science projects: with or without labeled data; stopping at data wrangling or involving machine learning algorithms; predicting classes or predicting numbers; with unevenly distributed classes, with binary classes, or even with no examples at all of one of the classes; with structured data and with unstructured data; using past samples or just remaining in the present; with real time or close to real-time time execution requirements or for acceptably slower performances; showing the results in shiny reports or hiding the nitty and gritty behind a neutral IT architecture; and last but not least with large budgets or no budget at all.

In the course of my professional life, I have seen many of such projects and their data science nuances. So much experience - and the inevitably related mistakes - should not be lost. Therefore, the idea of this book: a collection of data science case studies from past projects.

While the general development of a data science project is relatively standard, following, for example, the CRISP-DM cycle, each project often needs the cycle to be customized: that special ingredient added to adapt to the particular data, goals, constraints, domain knowledge, or even budget of the project.

This book is organized by application fields. We start with the oldest area in data science in chapter 1: the analysis of CRM data. We move on to retail stores with recommendation engines. And then discuss projects in the financial industry, about social media, time series analysis in IoT, and close finally with a number of cybersecurity projects.

All examples described in this book refer to a workflow (or two) which are available on the [KNIME EXAMPLES server](#). These reference workflows are dutifully reported at the beginning of each section. Please notice that all example workflows have been simplified for these use cases. All optimizations, model comparisons, model selections, and other experiments, are not shown here, to better focus on the conclusive details of each project only.

We will update this book as frequently as possible with the descriptions and workflows from the newest, most recent data science projects, as they become available.

We hope this collection of data science experiences will help grow the practical data science skills in the next generation of data scientists.

Rosaria Silipo

# Chapter 1. Customer Intelligence

## 1.1. Churn Prediction

By Rosaria Silipo

Access workflow on [hub.knime.com](https://hub.knime.com)

Or from: `EXAMPLES/50_Applications/18_Churn_Prediction`

Every company has CRM data. Even though a dataset from a CRM system is usually not that large, interesting applications can be developed using customer data from within the company. Churn prediction is one such application.

### What You Need

To predict the likelihood of your current customers churning, you need data from previous customers with their churn history. Typically, customer information in your CRM system concerns demographics, behavioral data, and revenue information. At the time of renewing contracts, some customers did, and some did not, i.e. they churned. These example customers, both the ones who churned and the ones who did not, can be used to train a model to predict which of the current customers are at risk of churning.

The dataset we used for this example was originally available as a free download from the [Iain Pardoe website](#). Currently, accessing the data is still possible because it is embedded in the workflow. The dataset consists of two files. The first file includes contract data for 3333 telco customers and the second file includes operational data for the same customers. The contract data contains, among various attributes, a churn field: churn = 0 indicates a renewed contract; churn = 1 indicates a closed contract.

### The Workflow to Train the Model

This is a binary classification problem. We want to predict which customer will churn (churn = 1) and which customer will not (churn = 0). That is:

attr 1, attr 2, ..., attr n => churn (0/1)

After rejoining the two parts of the data, contractual and operational, converting the churn attribute to a string for the upcoming machine learning algorithm, and coloring data rows in red (churn=1) or blue (churn=0) for purely esthetical purposes, we trained a machine learning model to predict churn as 0 or 1 depending on all other customer attributes.

### Which Model?

Which model shall we train? KNIME Analytics Platform offers a large variety of machine learning models to choose from. For this example, we trained a decision tree because of its appealing tree visualization. However, we could have used any other available machine learning algorithm for nominal class-like predictions, for example, Random Forests, Gradient Boosted Trees, or even deep learning. Given the small size of the dataset, though, we preferred not to overkill the application with an overly complex model.

Whatever machine learning algorithm you end up with, you always need to train it and evaluate (test) it. For this reason, the [Partitioning](#) node is required to split the data into one dataset for training and another one for testing. We chose a proportion with 80% training data vs. 20% test data.

The [Decision Tree Learner](#) node was fed with the training set (80% of the data). To train the decision tree (Decision Tree Learner node), we specified:

- the column with the class values to be learned (in this case: Churn)
- an information (quality) measure
- a pruning strategy (if any)
- the depth of the tree through the minimum number of records per node (higher number → shallower tree)
- the split strategies for nominal and numerical values.

At the end of the training phase, the “View” option in the node context menu showed the decision path throughout the tree to reach the leaves with churning and non-churning customers. After training, the decision tree model was saved to a file in [PMML](#) format.

At this point, we needed to evaluate the model before running it on real data. For the evaluation, we used the test set (the remaining 20% of data) to feed a [Decision Tree Predictor](#) node. This node applies the model to all data rows one by one and predicts the likelihood of that customer to churn given his/her contractual and operational data ( $P(\text{Churn}=0/1)$ ). Depending on the value of such probability, a predicted class will be assigned to the data row (Prediction (Churn) =0/1).

The number of times that the predicted class coincides with the original churn class is the basis for any measure of model quality as it is calculated by the [Scorer](#) node. The Scorer node offers a number of quality measures, like accuracy or Cohen’s Kappa. Notice that the customers with churn=0 are many more than the customers with Churn=1. Thus, in this case, the Cohen’s Kappa produces a more realistic measure of the model performance.

Notice also that the Scorer node – or any other scoring node – allows you to evaluate and compare different models. A subsequent Sorter node would allow you to select and retain only the best performing model.

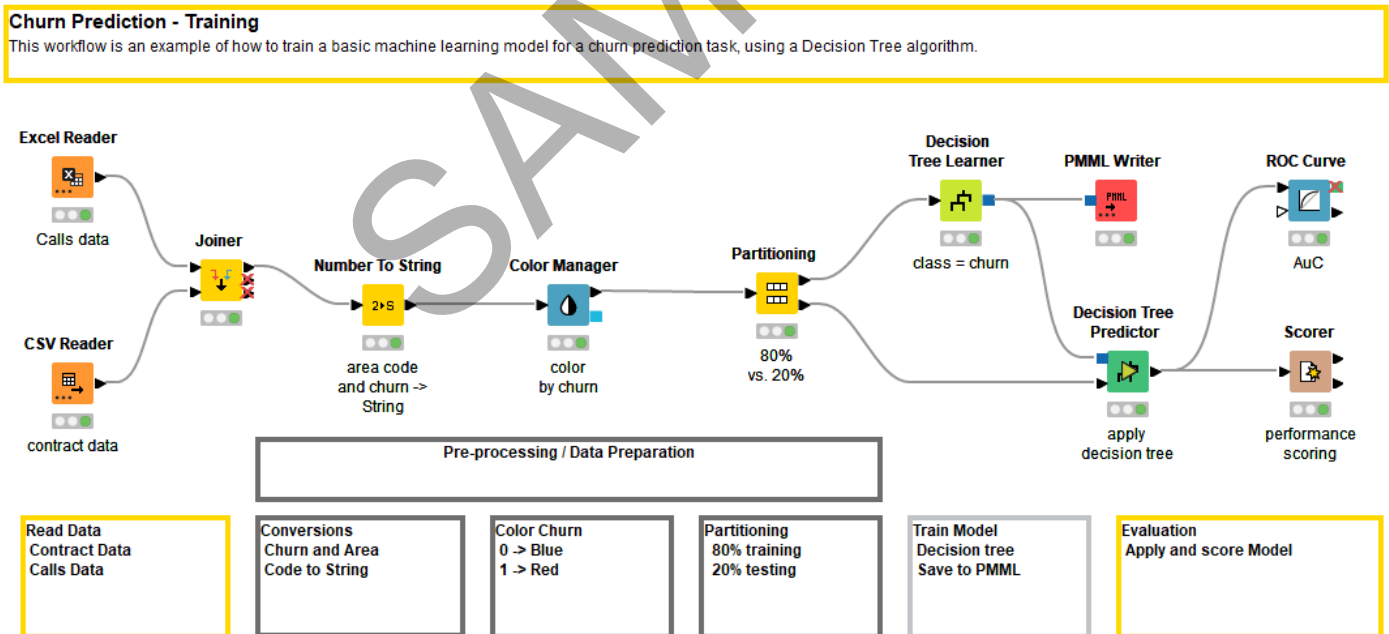


Figure 1. Training a Decision Tree model to predict the likelihood of customers to churn. Any other machine learning algorithm able to deal with binary classification can also be used instead of a decision tree.

## Deployment

Once the model performances are accepted, we moved the model into production for deployment on real data. Here we need only to read the stream of real-life data coming in through a file or database i.e. the data source and apply the generated model.

We then applied a Decision Tree Predictor node to run the model on the real-life input data (**Error! Reference source not found.**). Notice that as the model is structured in PMML format, we could have also used a [PMML Predictor](#) or a [JPMML Classifier](#) node. The output data will contain a few additional columns with the prediction class and the probability distributions for both classes, churn=0 and churn=1, (providing this is specified in the configuration settings of the predictor node).

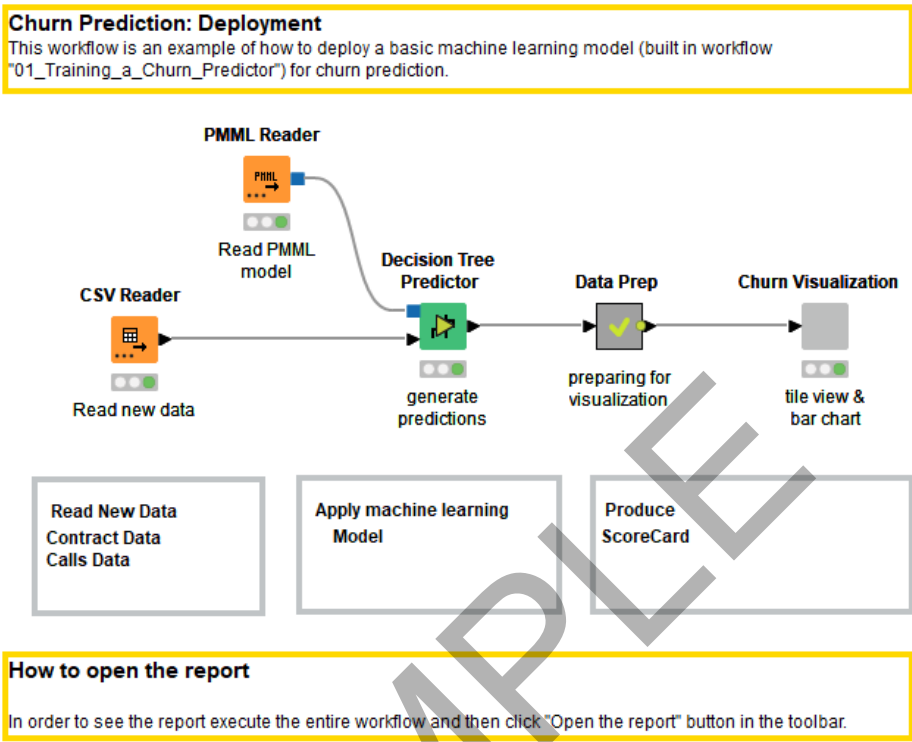


Figure 2 Applying the trained model to predict the likelihood of the current customer churning.

The last part of the production workflow is the result display. In our case, the input is data from one particular customer and the output is the likelihood that this customer is going to churn. We showed this via a speedometer chart in a BIRT report (Figure 3).

This likelihood score can be reassuring (below 40%), somewhat perplexing (above 40% and below 80%), or straightforward alarming (above 80%). Colors in the speedometer have been chosen accordingly to the alarm level. Different customer care actions have been devised for different likelihood scores.

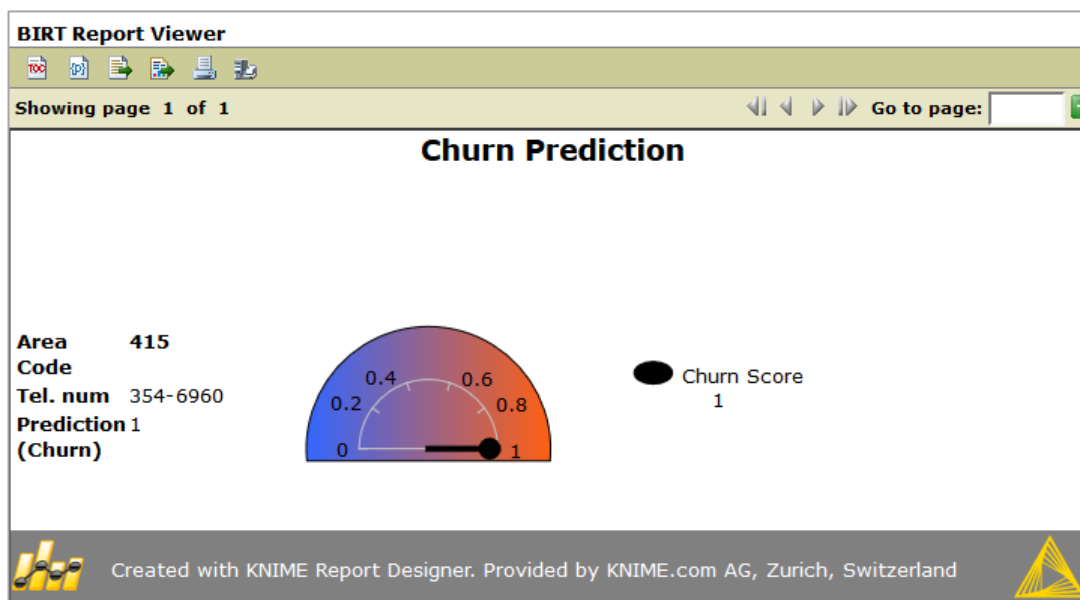


Figure 3. Displaying the likelihood to churn of the current customer via BIRT report.

## 1.2. Customer Segmentation

By Rosaria Silipo and Vincenzo Tursi

Access workflow on [hub.knime.com](https://hub.knime.com)

Or from: *EXAMPLES/50\_Applications/24\_Customer\_Segmentation\_Use\_Case*

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 simple. Group your customers together based on some attribute values, such as revenue creation, loyalty, demographics, buying behavior, etc ..., or any combination of these.

### Customer Segmentation Strategies

The group (or segment) definition can follow many strategies, depending on the degree of expertise and domain knowledge of the data scientist.

1. **Grouping by rules.** Somebody in the company already knows how the system works and how customers tick. It is already known how to group them together with respect to a given task, like for example a campaign. A [Rule Engine](#) node would suffice to implement this set of experience-based rules. This approach is highly interpretable, but not very portable to new analysis fields. In the event of a new goal, new knowledge, or new data the whole rule system needs to be redesigned.
2. **Grouping as binning.** Sometimes the goal is clear and not negotiable. One of the many features describing our customers is selected as the representative one, be it revenues, loyalty, demographics, or anything else. In this case, the operation of segmenting the customers in groups is reduced to a pure binning operation. Here customer segments are built along one or more attributes by means of bins. This task can be implemented easily, using one of the many binner nodes available in KNIME Analytics Platform.
3. **Grouping with no knowledge.** It is often safe to assume that the data scientist does not know enough of the business at hand to build his/her own customer segmentation rules. In this case, if no business analyst is around to help, the data scientist should resolve to a plain blind clustering procedure. The subsequent work of cluster interpretation can be done by the business analyst, who is (or should be) the domain expert.

With this goal in mind, of making this workflow suitable for a number of different use cases, we chose the third option.

There are many clustering procedures, which you can find in KNIME Analytics Platform in the category Analytics/Mining/Clustering of the Node Repository panel: e.g. k-Means, nearest neighbors, DBSCAN, hierarchical clustering, SOTA, etc ... We went for the most commonly used: the k-Means algorithm.

### What You Need

To cluster your current customers into different groups, you need data describing past and present customers. You can see your customers from many points of view: demographics, money flow, shopping behavior, loyalty, number of contracts, purchased products, and probably even more that are more strictly related to your business.

Often raw data do not describe all those customer perspectives. Usually, a pre-processing phase is needed to aggregate and transform the data to move from a number of contracts, for example, to the money flow or

to the loyalty score, from a weblog session to the clickstream history, etc. We will not describe these aggregation procedures here since they can be quite complex and often specific to the particular business. We will assume that our data adequately describe some aspects of our customers.

For more details on preparing customer data, you can check the following posts from the “Data Chef ETL Battles” series published in the KNIME blog:

- [“Customer Transactions. Money vs. Loyalty”, 2017](#)
- [“Energy Consumption Time Series. Behavioral Measures over Time and Seasonality Index from Auto-Correlation”, 2017](#)
- [“A Social Forum. Sentiment vs Influence”, 2018](#)

The dataset we used for this example was originally available as a free download from the [Iain Pardoe website](#). Currently, accessing the data is still possible because they are embedded in the workflow. The dataset consists of two files. The first file includes contract data for 3333 telco customers and the second file includes operational data for the same customers.

## Basic Workflow for Customer Segmentation with Clustering Procedure

The basic workflow for customer segmentation consists of only three steps: data reading, data pre-processing, and k-Means clustering. The clustering procedure - in this case the k-Means algorithm - and its associated normalization / de-normalization transformations represent the segmentation engine; that is the intelligent part of this workflow.

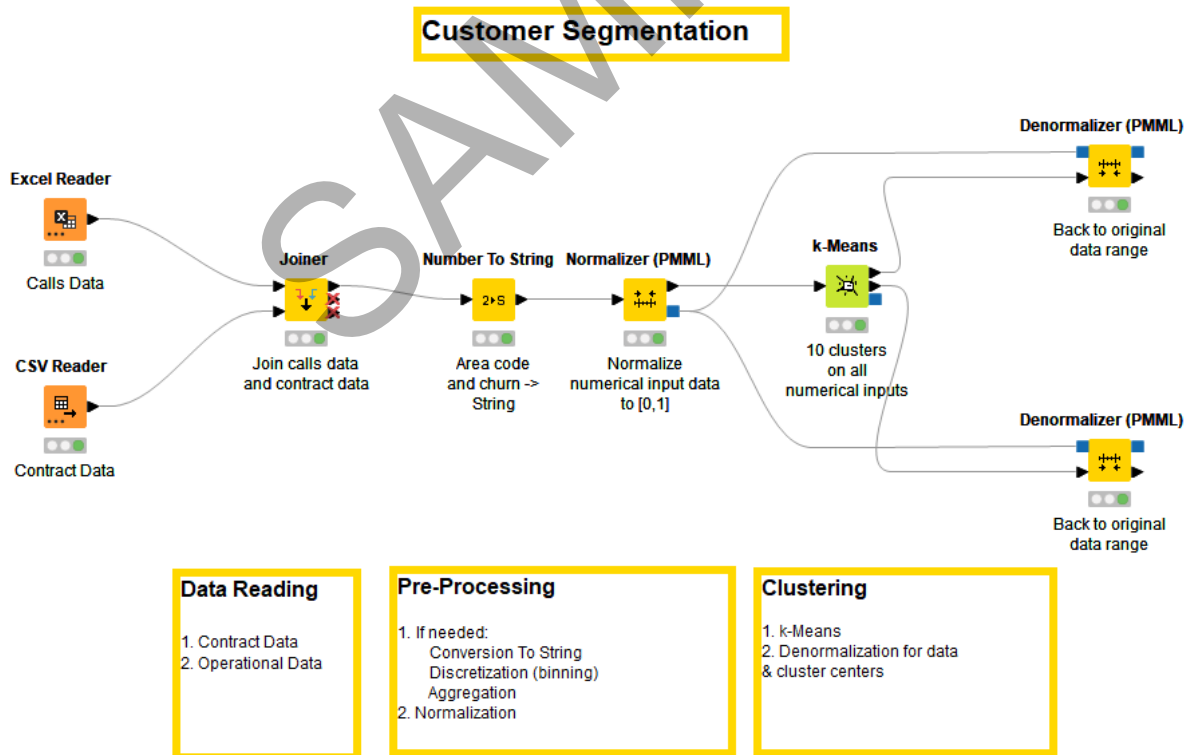


Figure 4. Basic Workflow for Customer Segmentation using k-Means as clustering technique.

Clustering could be replaced by a Rule Engine node (grouping by rules) or by an [Auto-Binner](#) or [Numeric Binner](#) node (grouping as binning) if knowledge becomes available and we decide to change the segmentation strategy.

## Model Deployment

The node to assign each data row to its cluster is the [Cluster Assigner](#) node. Here the distance is calculated between the input data row and all of the cluster centers. The cluster with the minimum distance is assigned to the input customer. Of course, the same pre-processing as performed before running the k-Means workflow is also required in the deployment workflow.

## Refining Customer Segments with Business Knowledge by Domain Experts

A desirable improvement of the previous segmentation workflow could consist of involving business analysts in the process. Modern business analysts have precious knowledge of the data acquisition process and of the business case. Allowing them to interact with the results of the segmentation is often beneficial.

The idea thus is to guide modern business analysts through all phases of the analysis not from within the workflow, but from a web browser.

In the second part of this project, we deployed the k-Means segmentation workflow on the [KNIME WebPortal](#). In this phase a web-based visualization wizard is created by strategically adding a number of Widget and JavaScript nodes to the workflow.

Indeed, on the KNIME WebPortal, workflow execution hops from a component containing Widget or JavaScript nodes to the next, producing at each step a web-based Guided User Interface (GUI) wizard. It is then possible to organize full guidance of the analysis on a web browser: for example, step 1 selects appropriate values, step 2 inspects the plot and readjusts selected values, and so on.

For each step, it is possible to design the GUI through multiple User Interface (UI) items on a web page layout, such as dropdown menus, radio buttons, interactive plots, and more. These UI components, generated by Widget and JavaScript nodes, form a web page, if placed inside a component. The layout of the web page produced by the component is controlled through a matrix layout available via a button in the tool bar at the top of KNIME Analytics Platform's workbench. This button is active only when the component is open in the workflow editor.

The first GUI step of the web-based wizard requires the number of segments and the data columns to be used for the segmentation. Segments (clusters) are then created in the background.

In the second step, a summary of all segments is displayed in a scatter plot and proposed to business analysts for inspection. The scatter plot is interactive, and the business analysts can decide whether or not to change the coordinates for a new inspection perspective, remove outliers, or drill down on a group of points. However, already with only 4 clusters, the scatter plot becomes hard to interpret.



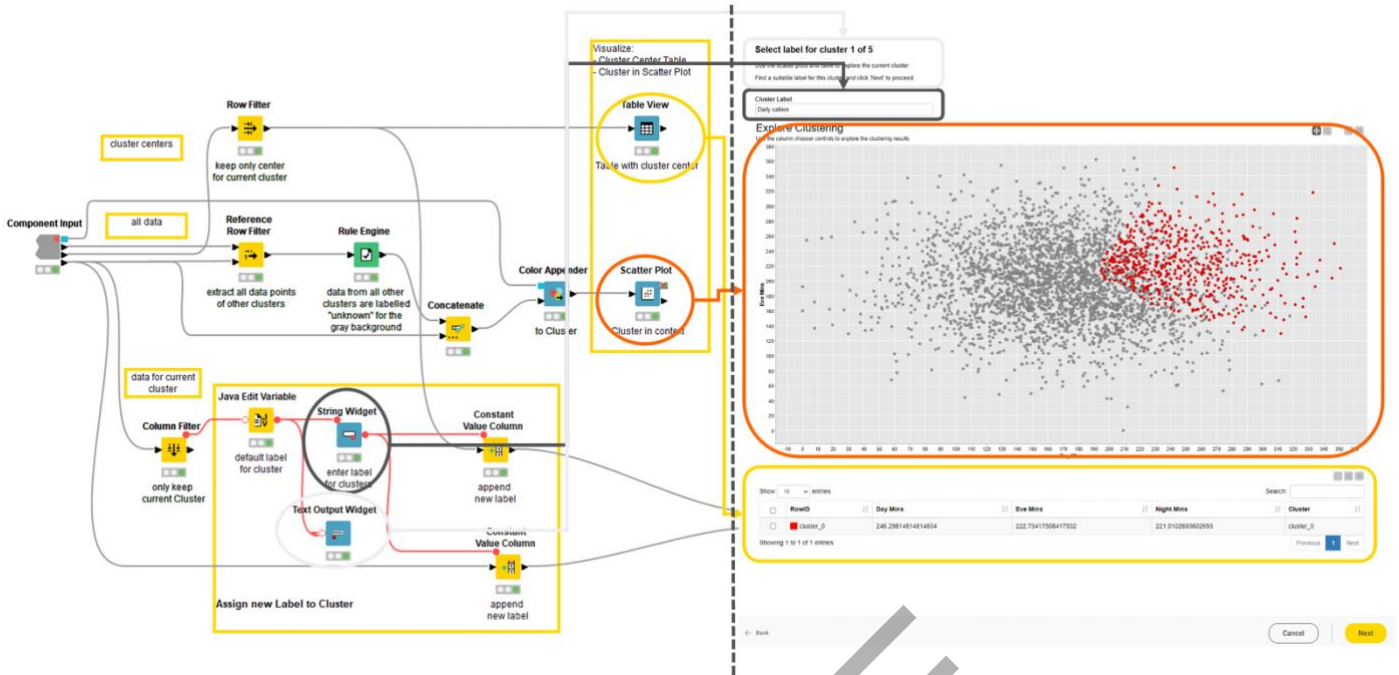


Figure 5. Widget and JavaScript nodes in a component (left) produce the webpage on right when running on KNIME WebPortal.

So, in the next steps the segments are displayed one by one. The data are taken through a loop where at each iteration the data points of one cluster are displayed in color against all other points displayed in gray. At each iteration, the wizard web page reports the scatter plot with the cluster, the table with the cluster centers, and a textbox to enter free text. The textbox allows the business analysts to appropriately label or annotate – for example with calls to action regarding the customer segment under scrutiny.

The complete workflow is shown in the figure below (Figure 6). The loop on the right goes through all clusters one by one. The component, Label Cluster, produces the web page shown in the previous figure. The whole sequence allows the business analysts to annotate and label the customer segments one by one, with the k-Means based segmentation part on the left and the visualization-interaction loop on the right. The content of the component, Label Cluster, and its corresponding web page is shown in the figure above (Figure 5Error! Reference source not found.).

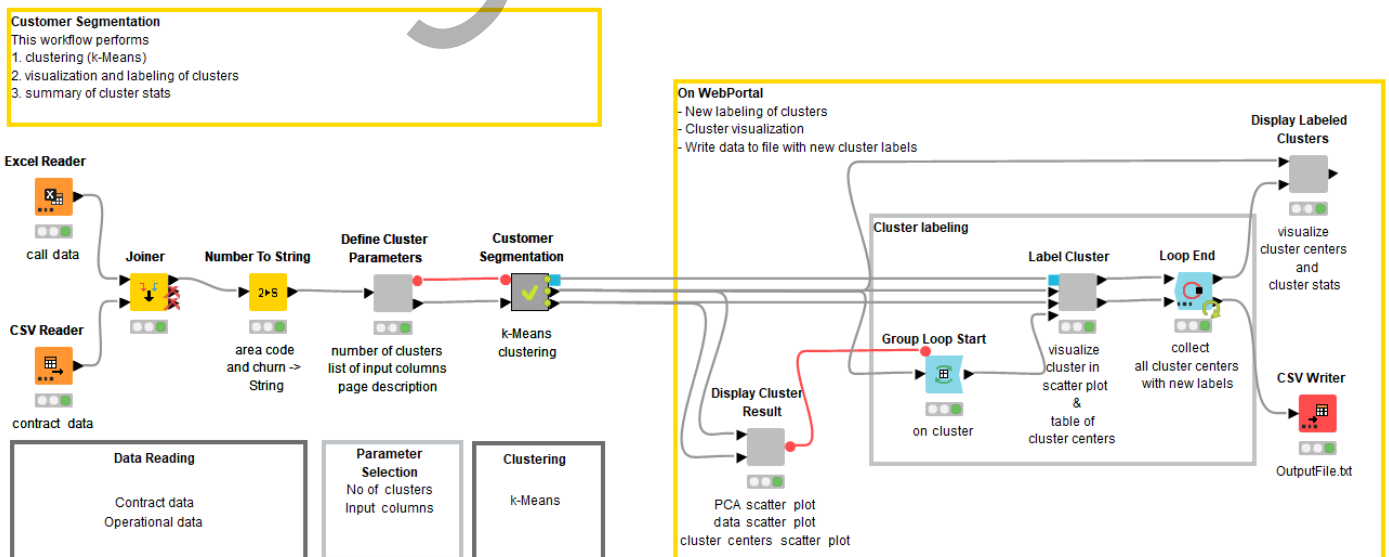


Figure 6. Final Workflow derived from the basic workflow for customer segmentation. The loop on the right goes through all clusters one by one. The component, Label Cluster, produces the web page shown in the previous figure. The whole sequence allows the business analysts to annotate and label the customer segments one by one.



This web-based analytics approach brings together the machine learning background of data analysts and the business domain knowledge of modern business analysts.

The whole project is described in detail in our recent whitepaper, named [“Customer Segmentation Conveniently from a Web Browser. Combining Data Science and Business Expertise”](#) and freely available for download.

The two workflows described in this post – for basic customer segmentation and for Web GUI guided customer segmentation - can be downloaded from the KNIME Hub from the [Examples space, Customer Segmentation Use Case page](#).

SAMPLE