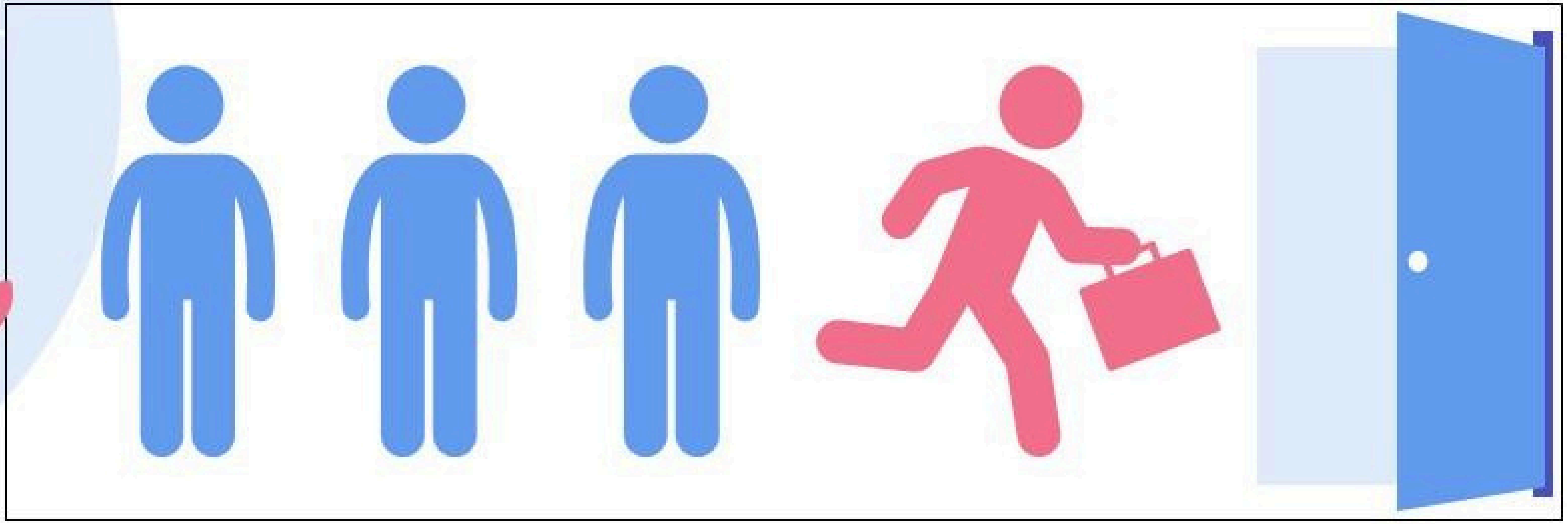


Customer Churn Prediction Using Machine Learning

By Verma SURYA
Busines Analytics
Neoma Business School

Building a Prediction Model

- **Step-by-Step Process**

1. **Data Reading**

- Source: Collect customer data (CSV, Excel, or database).
- Features: Include relevant attributes such as customer ID, contract details, and churn status.

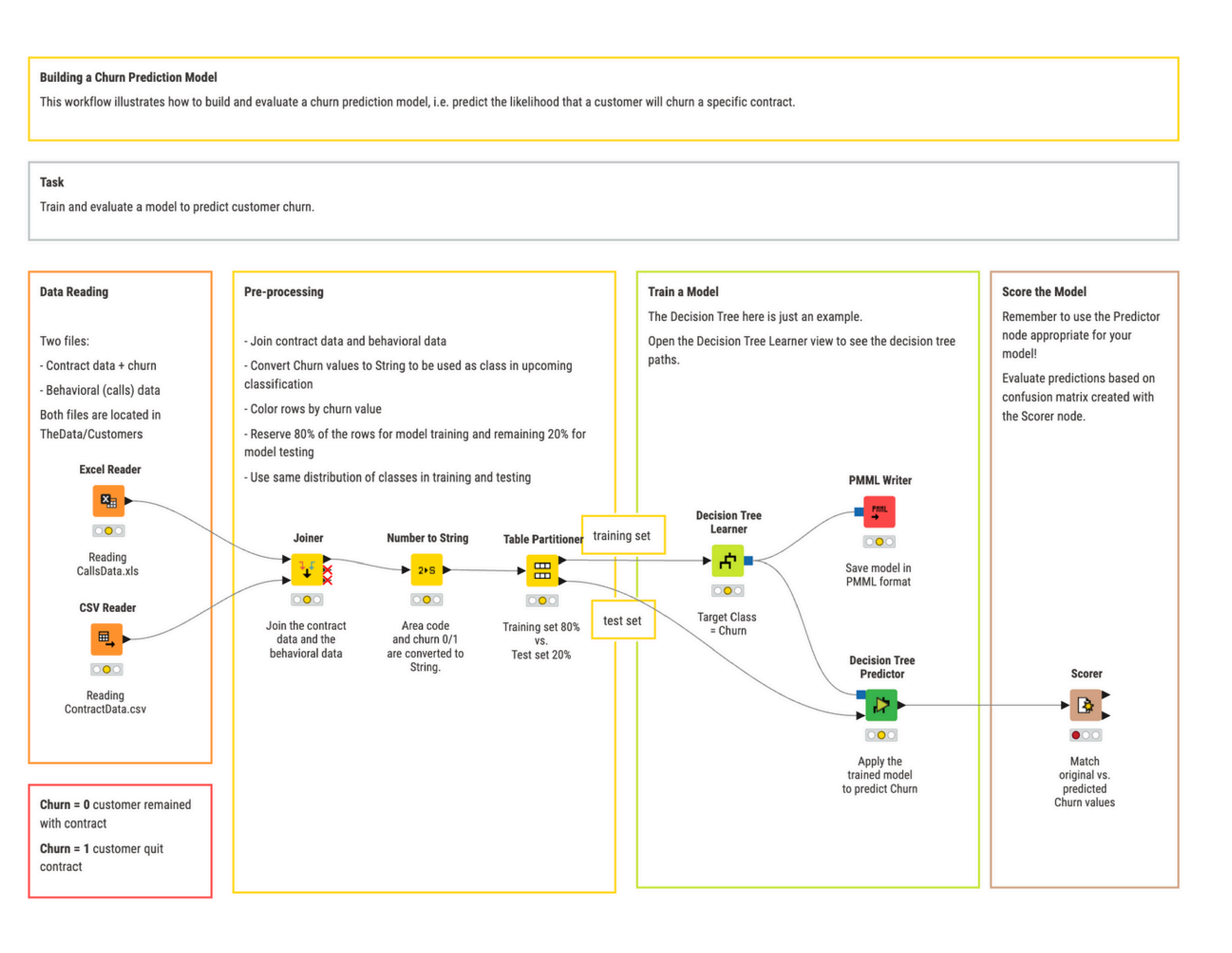
- **2. Pre-processing**

- Data Cleaning: Remove duplicate and handle missing values.
- Label Encoding: Convert churn tatus to binary (0 = remained, 1 = left).

Splitting Data: Divide data into training and test sets (e.g., 70% training, 30% test).

Feature

- Selection: Choose relevant
- features for modeling.



Building a Churn Prediction Model

This workflow illustrates how to build and evaluate a churn prediction model, i.e. predict the likelihood that a customer will churn a specific contract.

Task

Train and evaluate a model to predict customer churn.

Data Reading

Two files:

- Contract data + churn

- Behavioral (calls) data

Both files are located in TheData/Customers

Churn = 0 customer remained with contract

Churn = 1 customer quit contract

Pre-processing

- Join contract data and behavioral data

- Convert Churn values to String to be used as class in upcoming classification

- Color rows by churn value

- Reserve 80% of the rows for model training and remaining 20% for model testing

- Use same distribution of classes in training and testing

Train a Model

The Decision Tree here is just an example.

Open the Decision Tree Learner view to see the decision tree paths.

Score the Model

Remember to use the Predictor node appropriate for your model!

Evaluate predictions based on confusion matrix created with the Scorer node.

Excel Reader
Reading CallsData.xls
CSV Reader
Reading ContractData.csv
Joiner
Join the contract data and the behavioral data
Number to String
2▸S
Area code and churn 0/1 are converted to String.
Table Partitioner
Training set 80% vs. Test set 20%
training set
test set
Decision Tree Learner
Target Class = Churn
PMML Writer
PMML
Save model in PMML format
Decision Tree Predictor
Apply the trained model to predict Churn
Scorer
Match original vs. predicted Churn values

3. Model Training

- Algorithm: Use a Decision Tree or a similar classifier.
- Training: Fit the model using the training dataset.

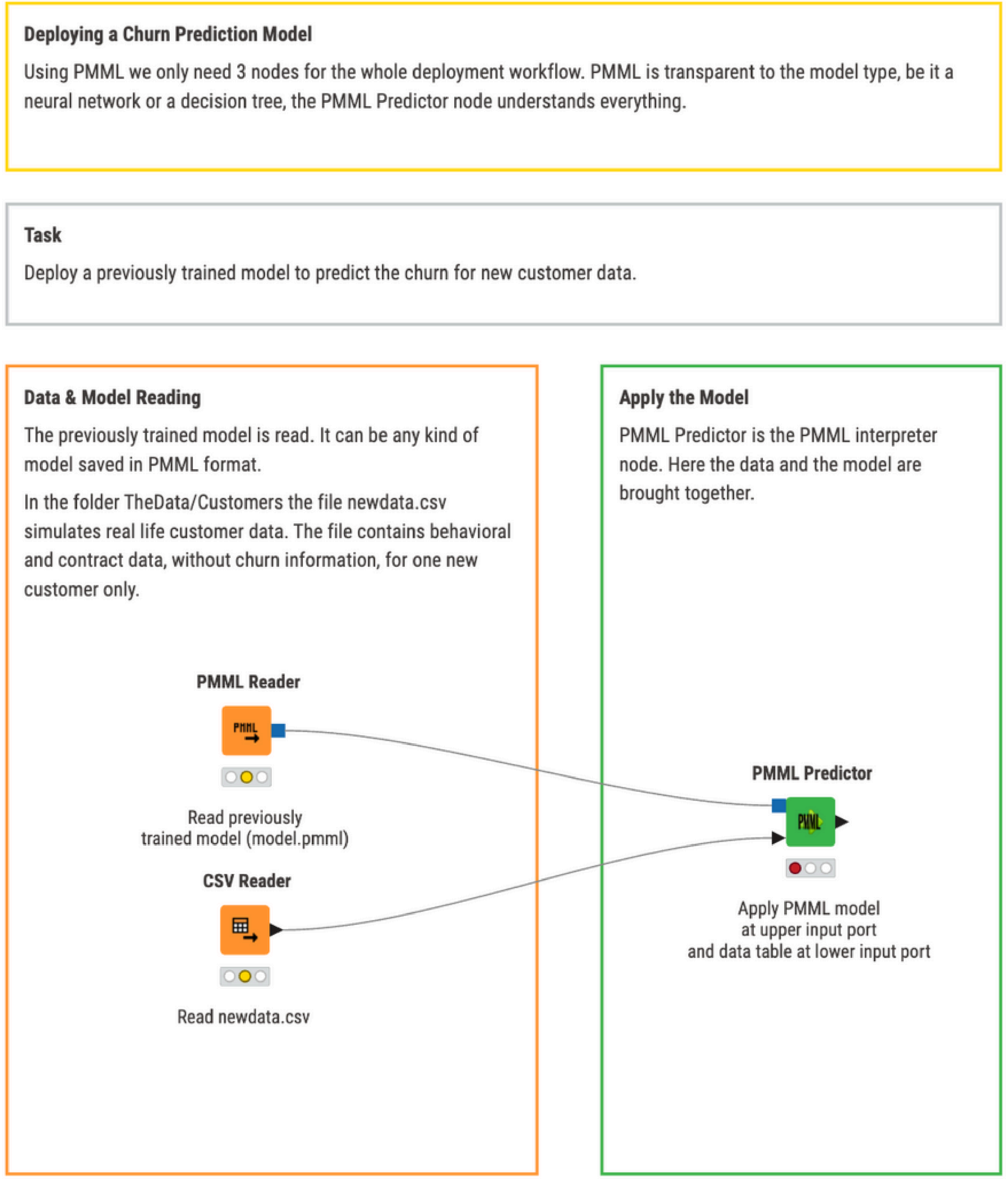
4. Model Testing

- Validation: Test the model on the unseen test dataset.
- Prediction: Predict churn
- status for test data.
-

5. Model Evaluation

- Scoring: Compare predictions
- with actual outcomes.
- Metrics: Use accuracy, precision, recall, or F1-score to evaluate performance.
-

Deploying the churn Prediction Model



Deploying a Churn Prediction Model

Using PMML we only need 3 nodes for the whole deployment workflow. PMML is transparent to the model type, be it a neural network or a decision tree, the PMML Predictor node understands everything.

Task

Deploy a previously trained model to predict the churn for new customer data.

Data & Model Reading

The previously trained model is read. It can be any kind of model saved in PMML format.

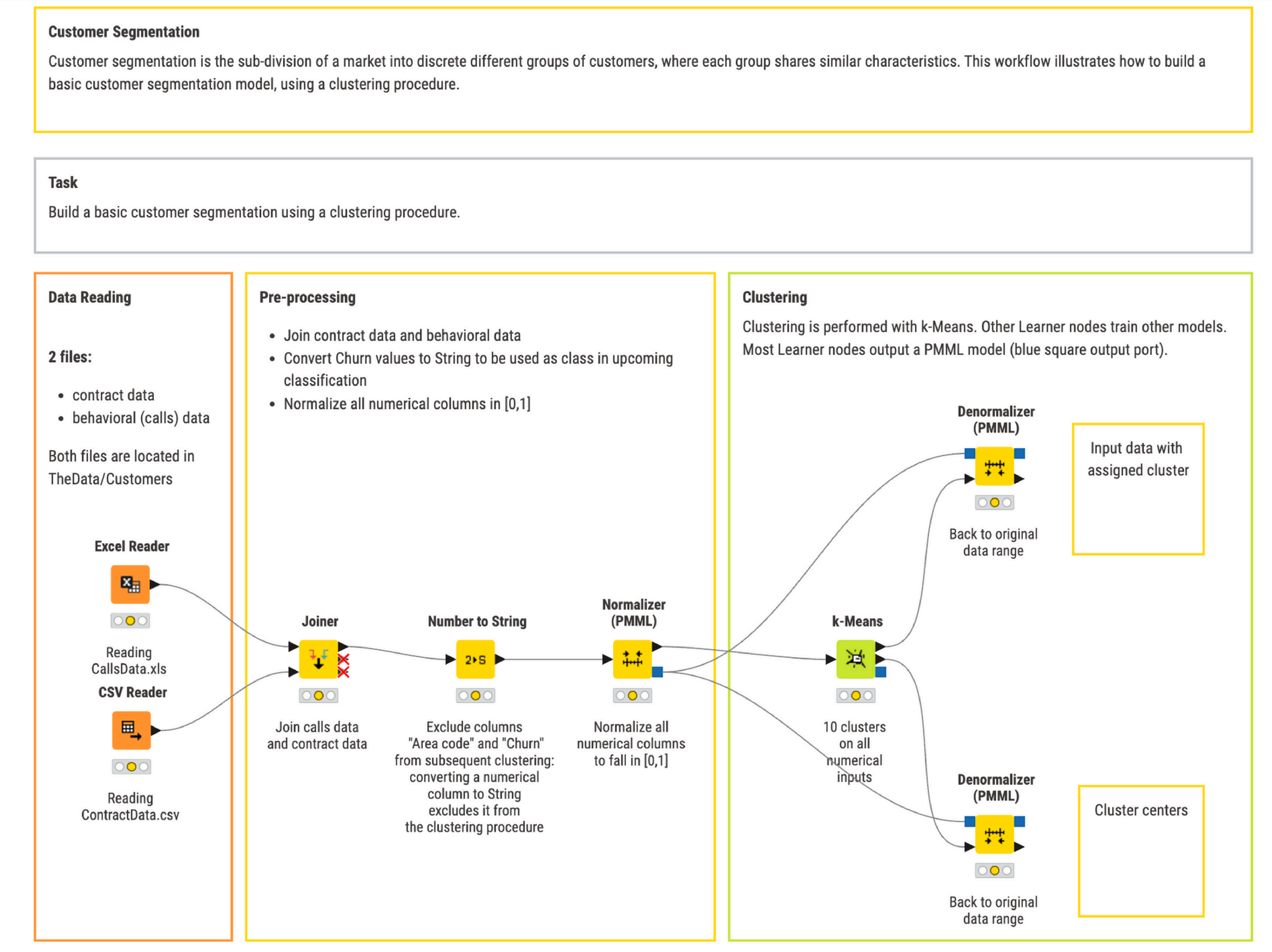
In the folder TheData/Customers the file newdata.csv simulates real life customer data. The file contains behavioral and contract data, without churn information, for one new customer only.

Apply the Model

PMML Predictor is the PMML interpreter node. Here the data and the model are brought together.

PMML Reader
PMML
Read previously trained model (model.pmml)
CSV Reader
Read newdata.csv
PMML Predictor
PMML
Apply PMML model at upper input port and data table at lower input port

Basic Customer Segmentation



Customer Segmentation

Customer segmentation is the sub-division of a market into discrete different groups of customers, where each group shares similar characteristics. This workflow illustrates how to build a basic customer segmentation model, using a clustering procedure.

Task

Build a basic customer segmentation using a clustering procedure.

Data Reading

2 files:

- contract data
- behavioral (calls) data

Both files are located in TheData/Customers

Pre-processing

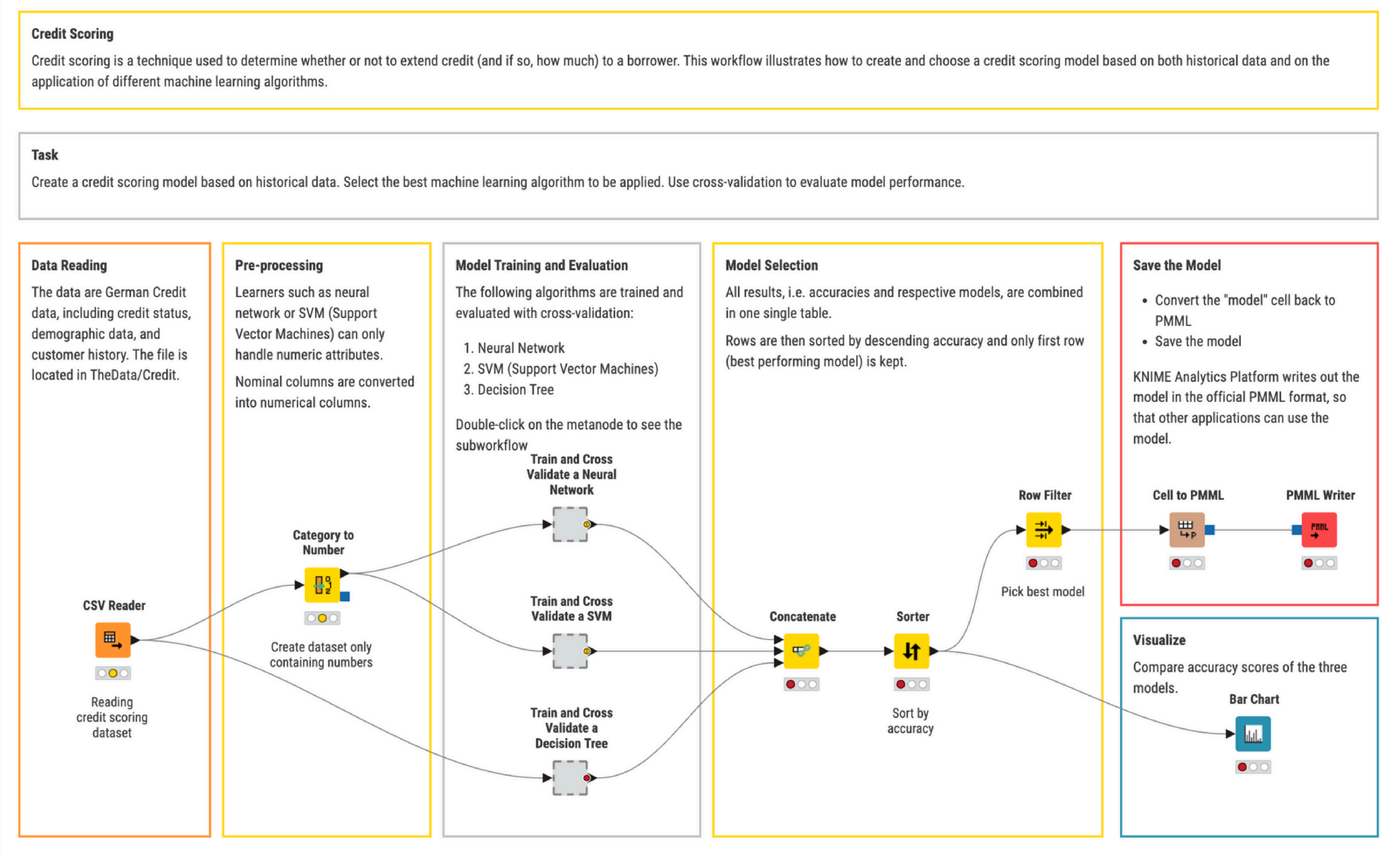
- Join contract data and behavioral data
- Convert Churn values to String to be used as class in upcoming classification
- Normalize all numerical columns in [0,1]

Clustering

Clustering is performed with k-Means. Other Learner nodes train other models. Most Learner nodes output a PMML model (blue square output port).

Excel Reader
Reading CallsData.xls
CSV Reader
Reading ContractData.csv
Joiner
Join calls data and contract data
Number to String
Exclude columns "Area code" and "Churn" from subsequent clustering: converting a numerical column to String excludes it from the clustering procedure
Normalizer (PMML)
Normalize all numerical columns to fall in [0,1]
k-Means
10 clusters on all numerical inputs
Denormalizer (PMML)
Back to original data range
Input data with assigned cluster
Denormalizer (PMML)
Back to original data range
Cluster centers

Building a Credit Scoring Model



Credit Scoring

Credit scoring is a technique used to determine whether or not to extend credit (and if so, how much) to a borrower. This workflow illustrates how to create and choose a credit scoring model based on both historical data and on the application of different machine learning algorithms.

Task

Create a credit scoring model based on historical data. Select the best machine learning algorithm to be applied. Use cross-validation to evaluate model performance.

Data Reading

The data are German Credit data, including credit status, demographic data, and customer history. The file is located in TheData/Credit.

Pre-processing

Learners such as neural network or SVM (Support Vector Machines) can only handle numeric attributes.

Nominal columns are converted into numerical columns.

Model Training and Evaluation

The following algorithms are trained and evaluated with cross-validation:

1. Neural Network
2. SVM (Support Vector Machines)
3. Decision Tree

Double-click on the metanode to see the subworkflow

Model Selection

All results, i.e. accuracies and respective models, are combined in one single table.

Rows are then sorted by descending accuracy and only first row (best performing model) is kept.

Save the Model

- Convert the "model" cell back to PMML
- Save the model

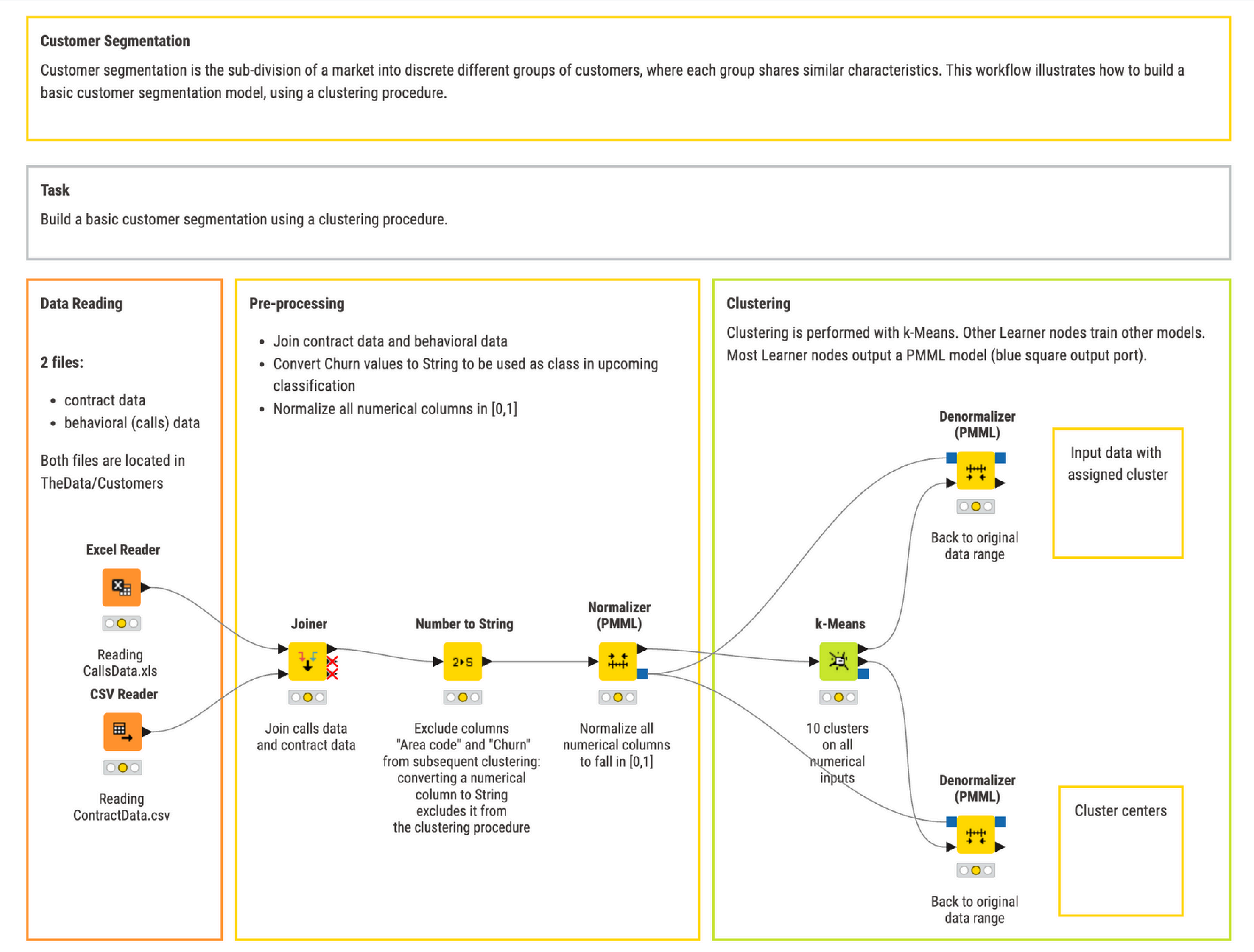
KNIME Analytics Platform writes out the model in the official PMML format, so that other applications can use the model.

Visualize

Compare accuracy scores of the three models.

CSV Reader
Reading credit scoring dataset
Category to Number
Create dataset only containing numbers
Train and Cross Validate a Neural Network
Train and Cross Validate a SVM
Train and Cross Validate a Decision Tree
Concatenate
Sorter
Sort by accuracy
Row Filter
Pick best model
Cell to PMML
PMML Writer
Bar Chart

Basic Customer Segmentation use case



Customer Segmentation

Customer segmentation is the sub-division of a market into discrete different groups of customers, where each group shares similar characteristics. This workflow illustrates how to build a basic customer segmentation model, using a clustering procedure.

Task

Build a basic customer segmentation using a clustering procedure.

Data Reading

2 files:

- contract data
- behavioral (calls) data

Both files are located in TheData/Customers

Pre-processing

- Join contract data and behavioral data
- Convert Churn values to String to be used as class in upcoming classification
- Normalize all numerical columns in [0,1]

Clustering

Clustering is performed with k-Means. Other Learner nodes train other models. Most Learner nodes output a PMML model (blue square output port).

Input data with assigned cluster

Cluster centers

Excel Reader
Reading CallsData.xls
CSV Reader
Reading ContractData.csv
Joiner
Join calls data and contract data
Number to String
Exclude columns "Area code" and "Churn" from subsequent clustering: converting a numerical column to String excludes it from the clustering procedure
Normalizer (PMML)
Normalize all numerical columns to fall in [0,1]
k-Means
10 clusters on all numerical inputs
Denormalizer (PMML)
Back to original data range
Denormalizer (PMML)
Back to original data range