**Churn Prediction Using KNIME Tool**

**Churn prediction** modelling techniques attempt to understand the precise customer behaviours and attributes which signal the risk and timing of customer churn.**Churn prediction** is the use of statistics and data about your customers to try to model who might leave for another service. Customer churn, also known as customer attrition, occurs when customers stop doing business with a company. The companies are interested in identifying segments of these customers because the price for acquiring a new customer is usually higher than retaining the old one.

Workflows and data are available on our [**GitHub**](https://github.com/SMIIT-Projects/Churn-Prediction-Using-KNIME-Tool) Account

**Link:** <https://github.com/SMIIT-Projects/Churn-Prediction-Using-KNIME-Tool>

In this blog, we will predict Customer's Exit Prediction from Bank. That’s way we will create a simple customer churn prediction model using Bank Customer Dataset. For this Churn Prediction model here we used KNIME tool.

**The Churn Prediction Problem**

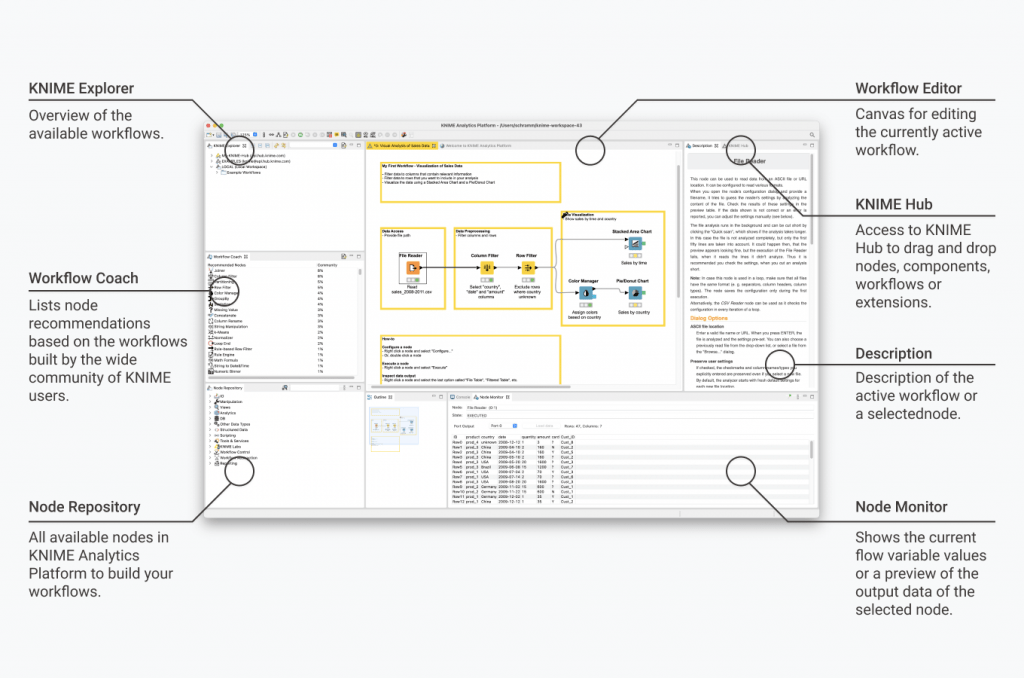
Typical information that is available about Customer ID, Credit Score, Geography, Gender, Age, Tenure, Bank Balance, NumOfProducts, HasCrCard, IsActiveMember, Estimated Salary, and Exited. at the time of Exited Customers, some customers are Exited from Bank and some not Exited: they churn.

This is a prediction problem. Starting with a small training set, where we can see who has Exited and who has not in the past, we want to predict which customer will Exited (Exited = 1) and which customer will not (Exited = 0). Using this Bank’s past dataset we will be predict Classification Model.

In this dataset here we have dependent and independent features like: Exited is a dependent feature and remaining all features are independent features as like it’s mention are above.

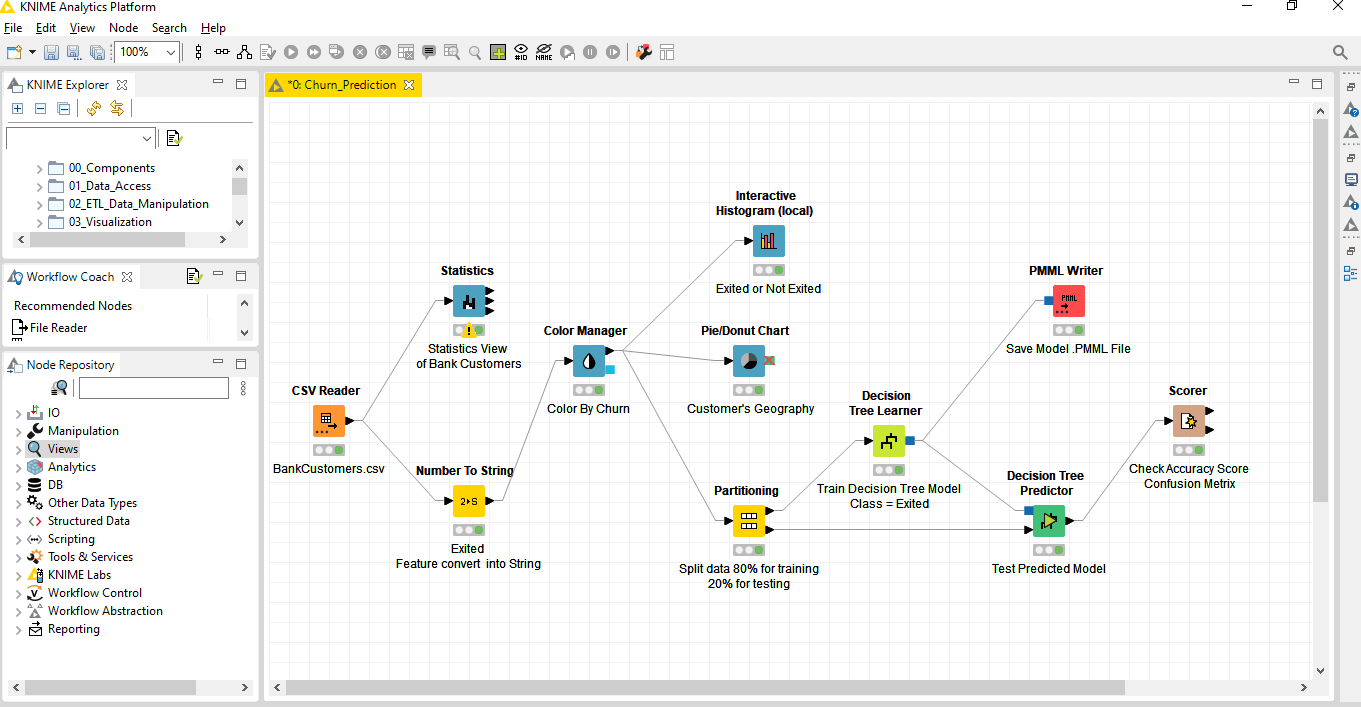
**What is KNIME Analytics Platform?**

KNIME Analytics Platform is the free, open-source software for creating data science. It is helping you discover the potential hidden in your data, mine for fresh insights, or predict new features.



**Build Our First Workflow in KNIME**

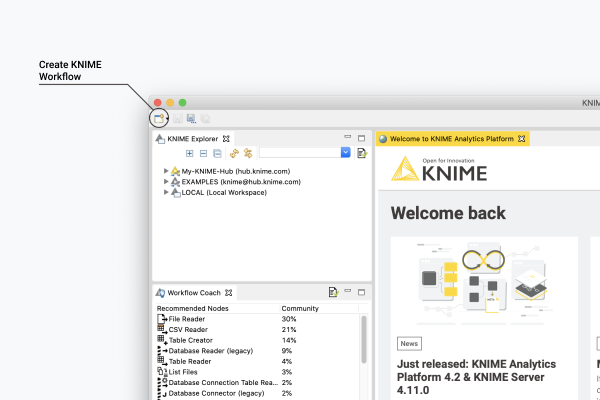
Let’s say that we have some Bank Customers data that we want to process, analyze and visualize. With the following example workflow, we will read, transform and visualize this BankCustomers.csv data.



**Build Customer Churn Prediction Model in KNIME**

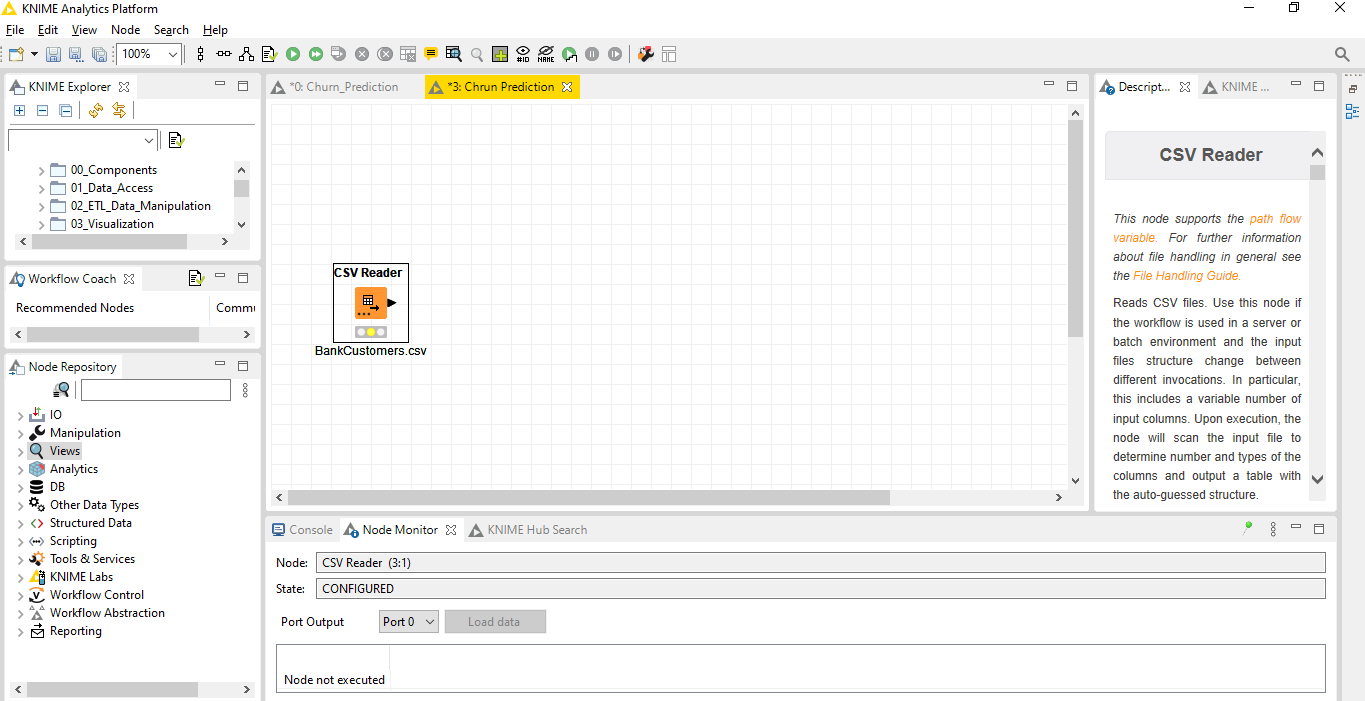
**Step 1 – Download data from our GitHub account and create new workflow**

To get started, first download the BankCustomers.CSV file that contains the data that you are going to use in the workflow. Open your Analytics Platform and create a new, empty workflow by clicking “New” in the toolbar.

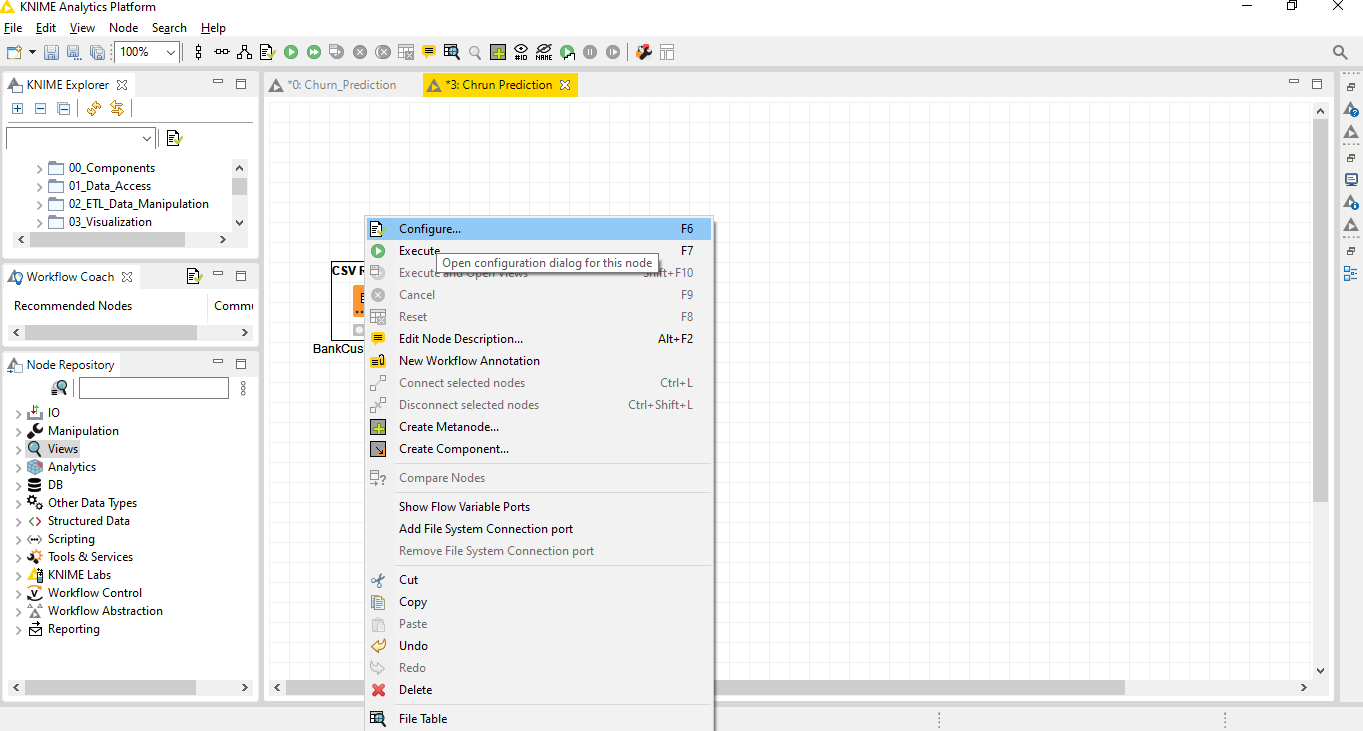


**Step 2 – Drag and drop CSV file into workbench editor**

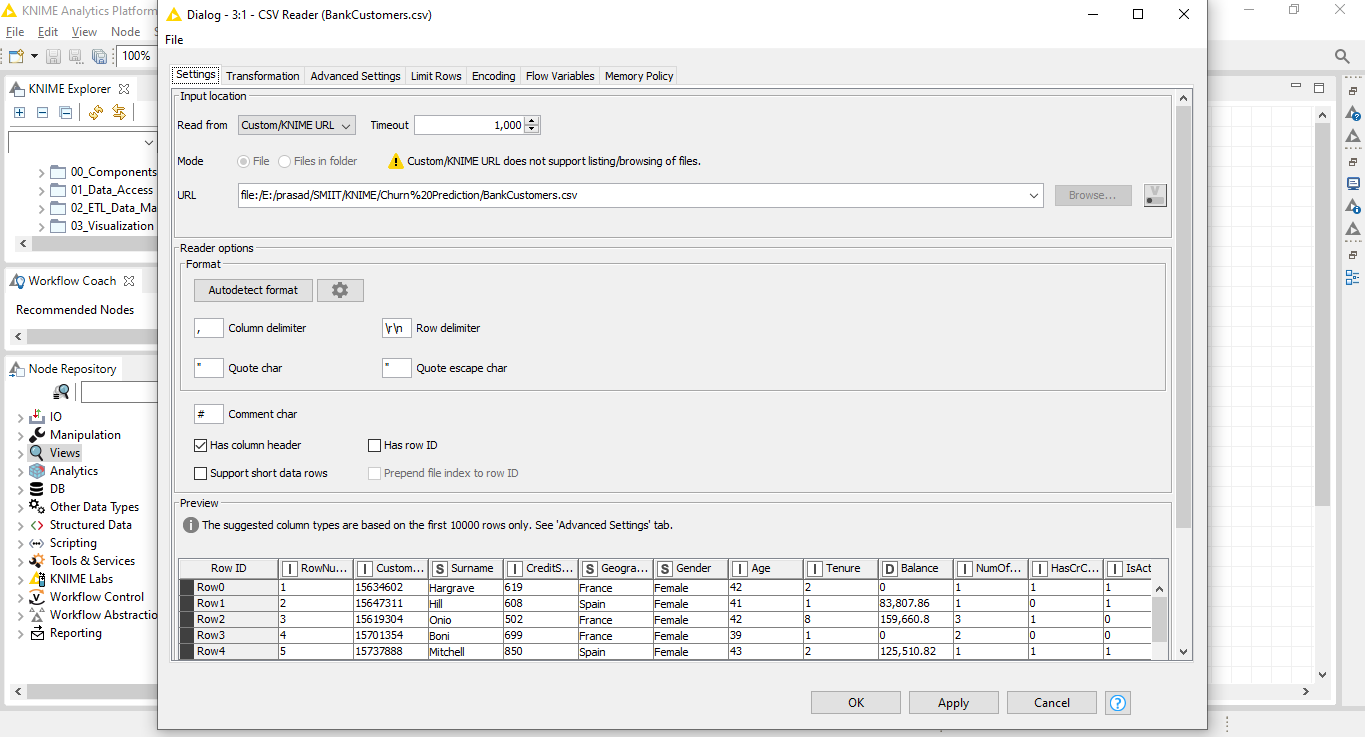
From the download folder, drag and drop the CSV-file into the workbench editor.



A File Reader node will appear on the workflow editor and its configuration dialog will pop-up.



After select configure then will open below dialog box. Here we shown head of the data frame rows and columns. Click on OK.



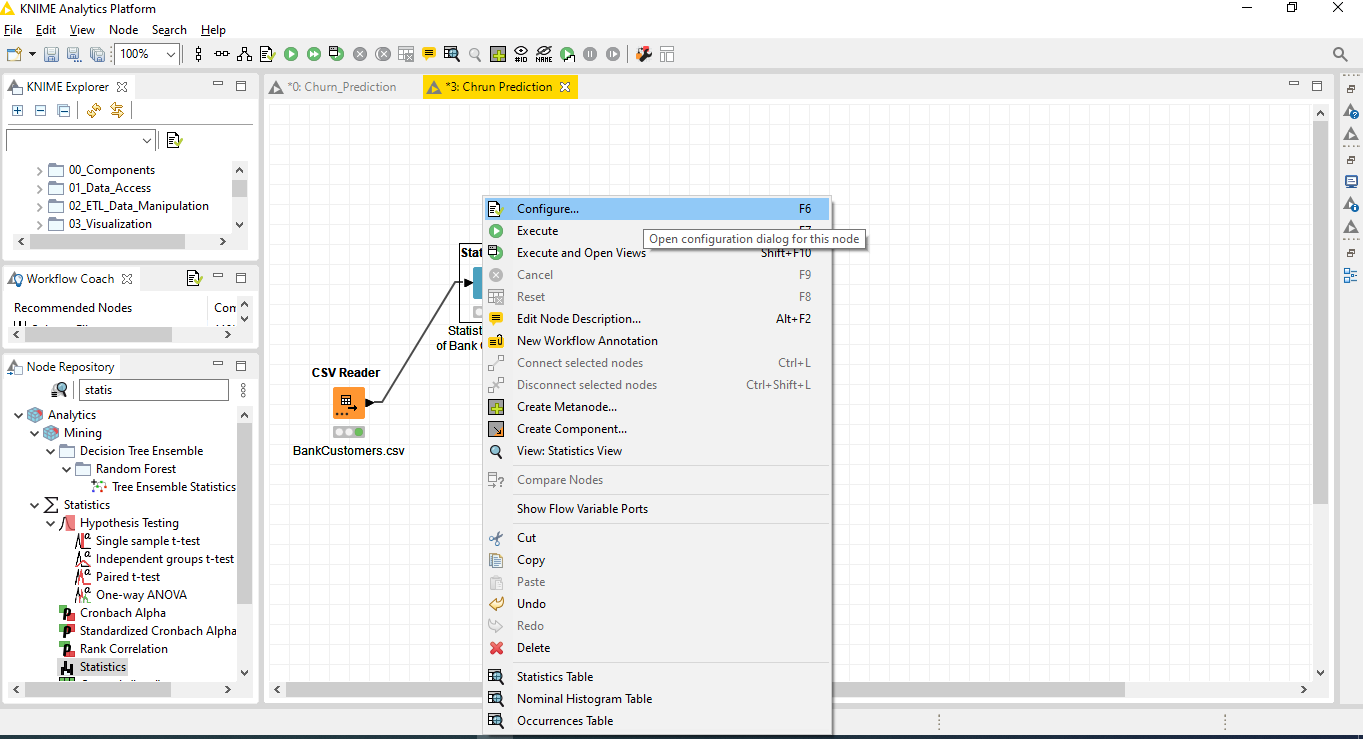
**Step 3 – Statistics Node, Statistics view of the Bank Customers.**

To Visualize some of the columns, by using the Statistics node.

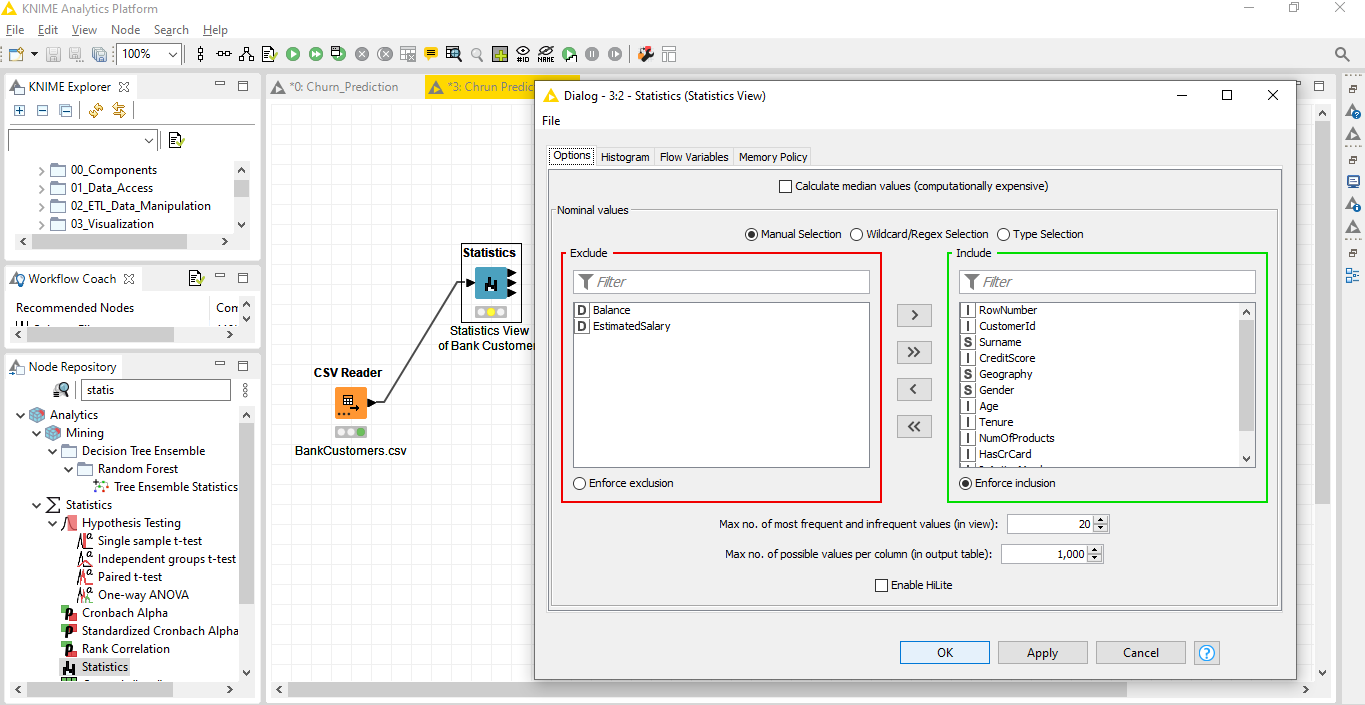
In the node repository panel on the left:

* write “Statistics” in the search field
* drag&drop the Statistics node to your workflow editor

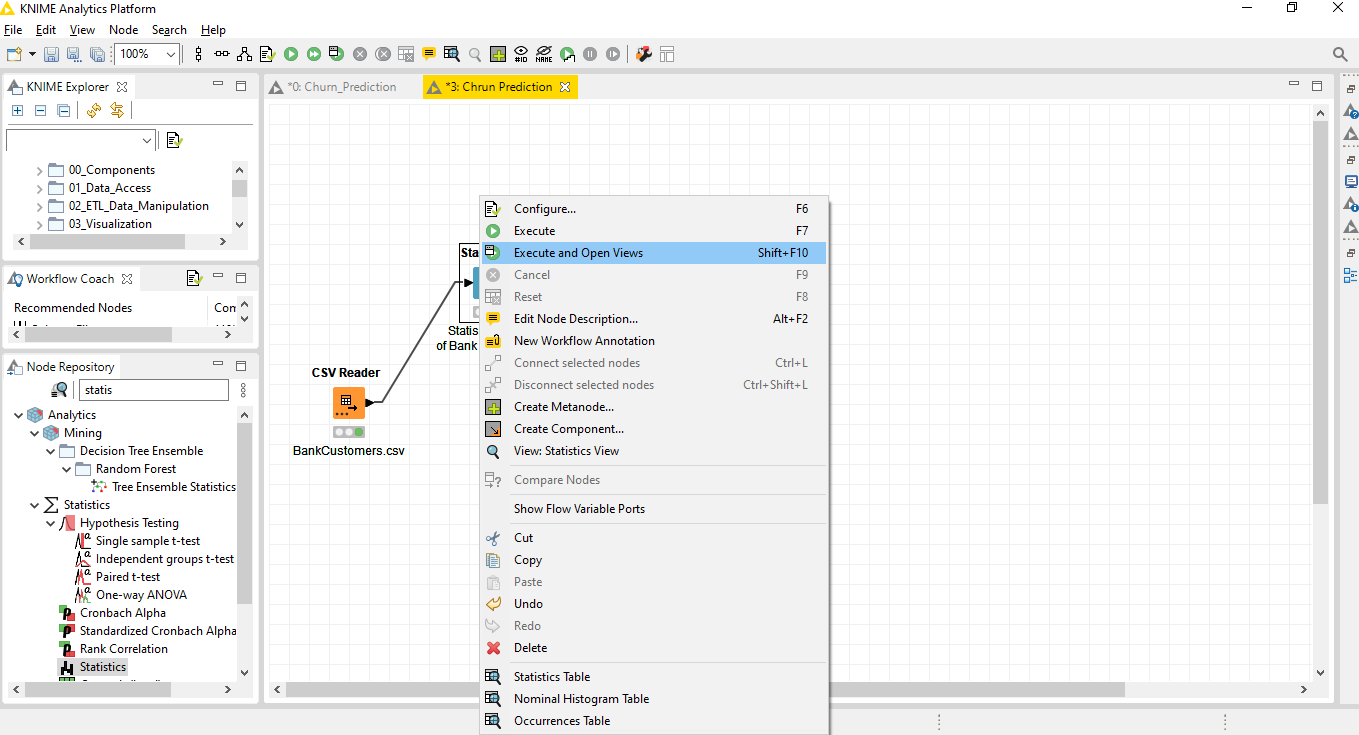
To open the configuration dialog, right-click the node and choose Configure.



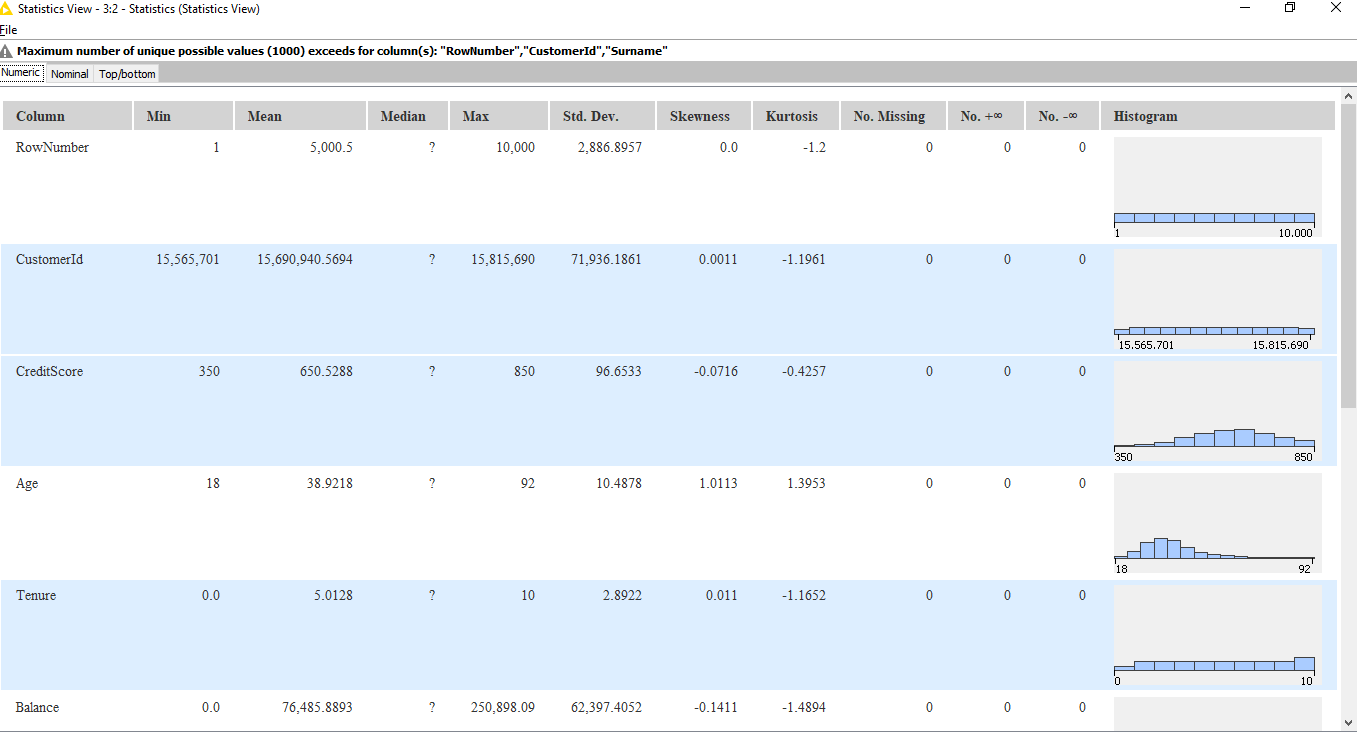
Here, move the “Balance”, “EstimatedSalary” into the Exclude field on the left side of the dialog, then click OK. After executing the node, the Statistics data table is available at the output port of the Statistics node.

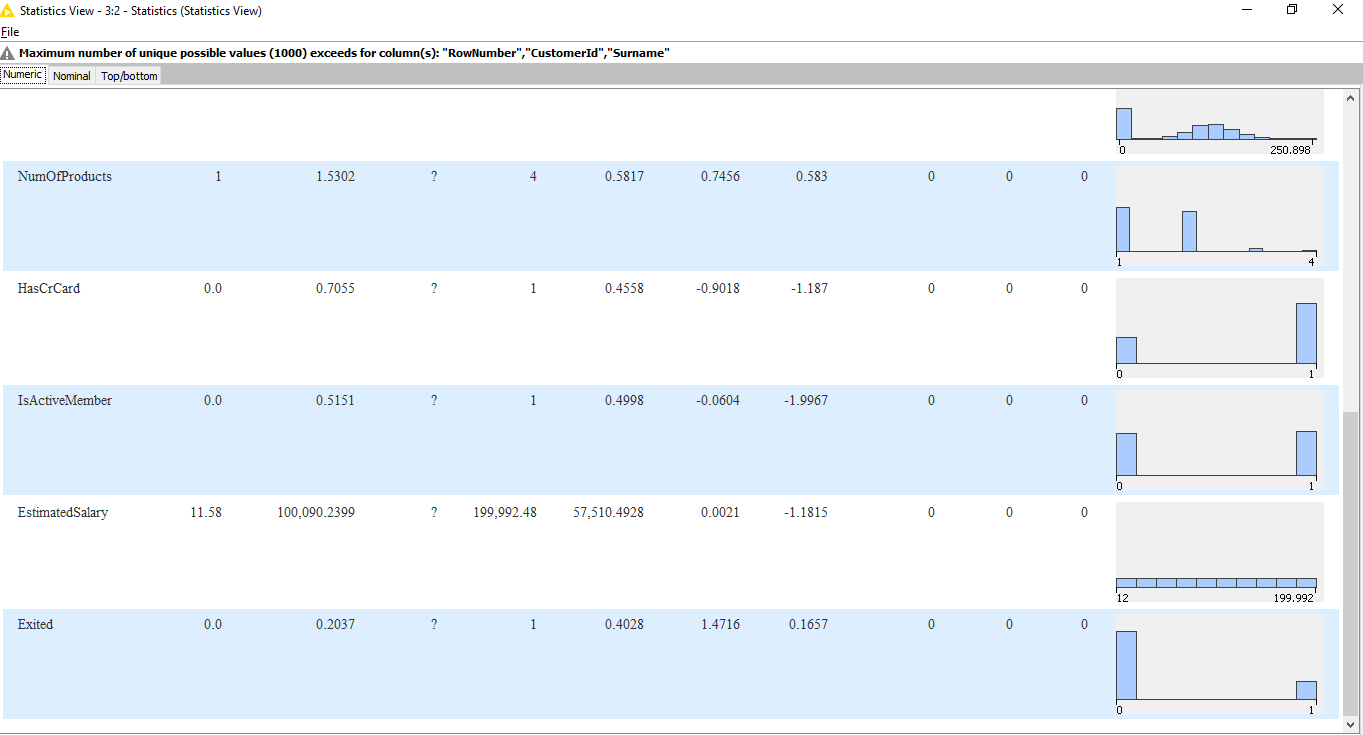


After that right click on Statistics node and then select Execute and open view

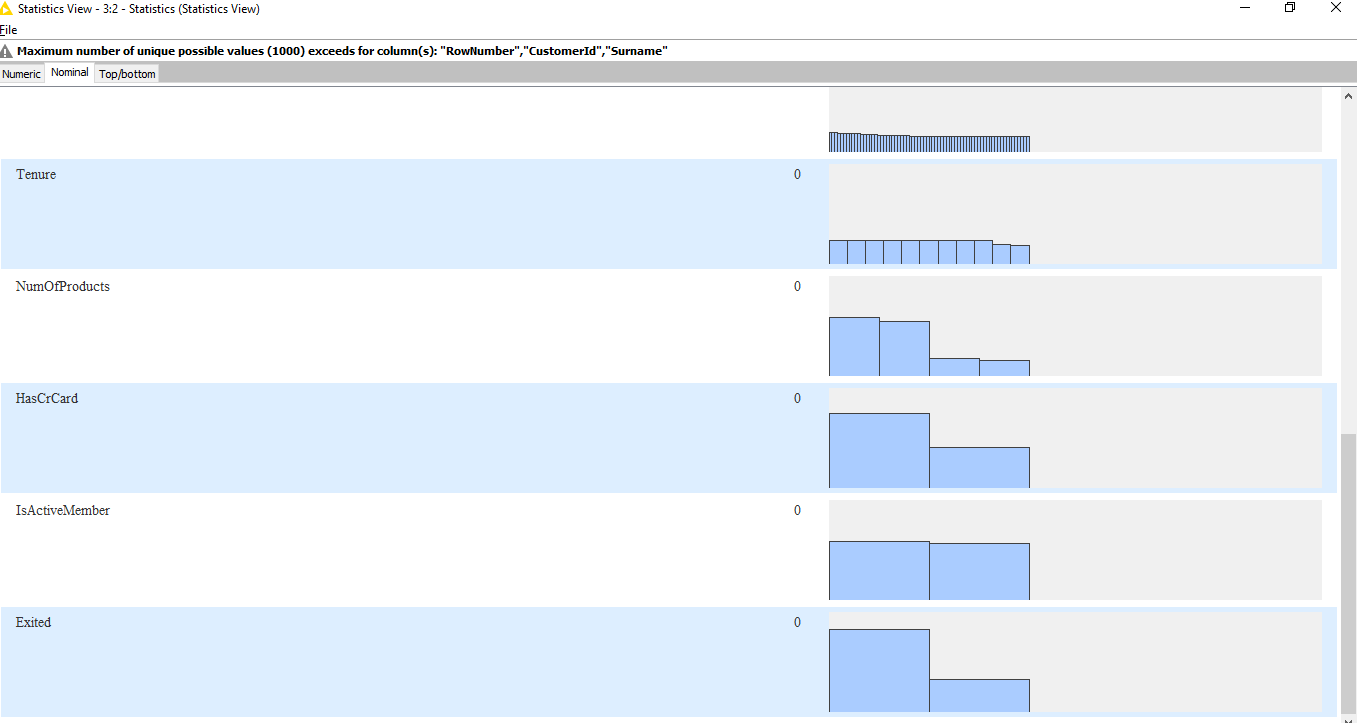
****

Here we shown Numeric view of whole dataset like: Column Name, Min, Mean, Median, Max, Std. Dev, Skewness, Kurtosis, No. Missing, Histogram.

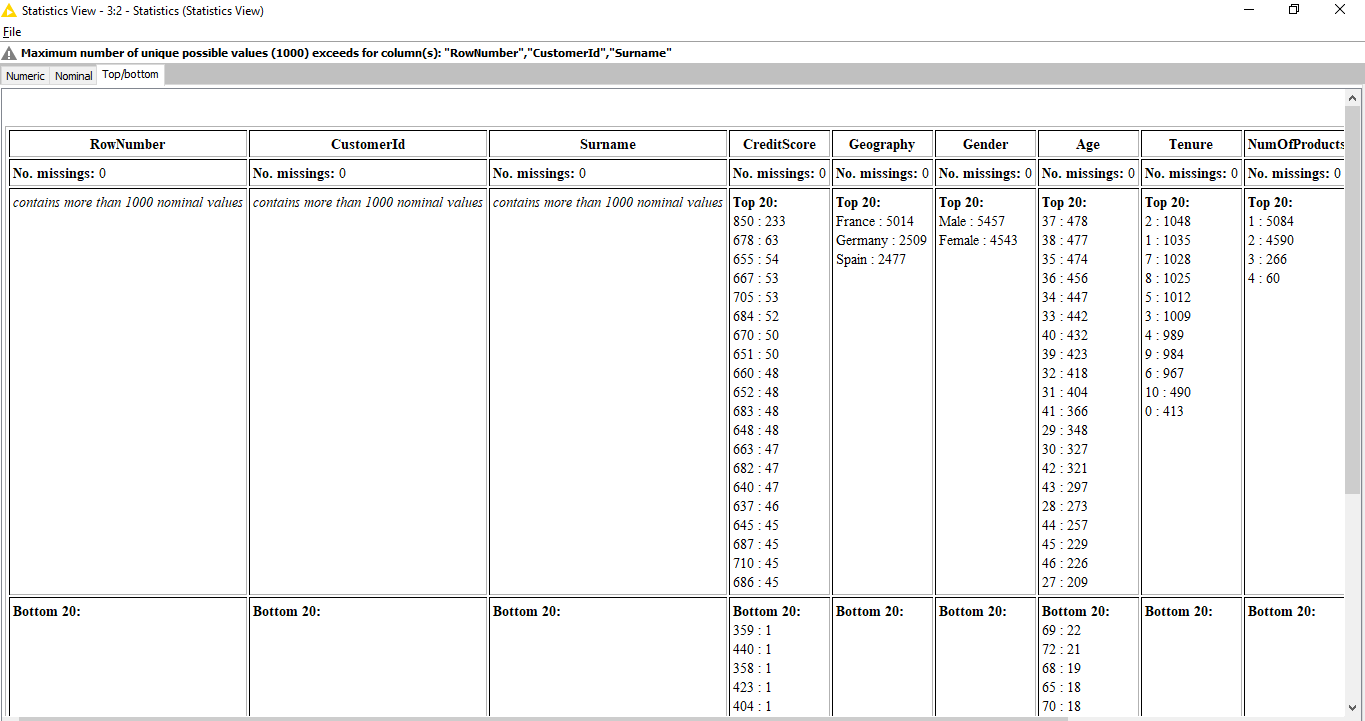
****

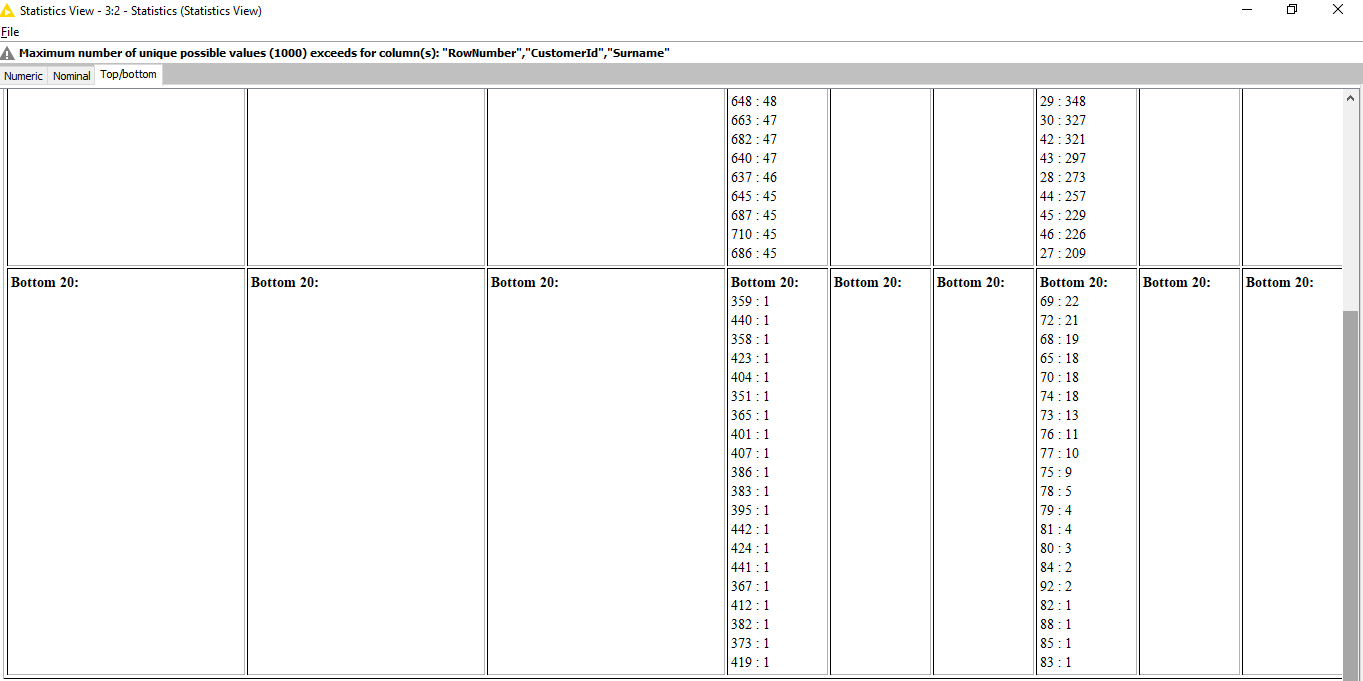
****

**Nominal View of dataset**

****

**Top Bottom View:**

****

****

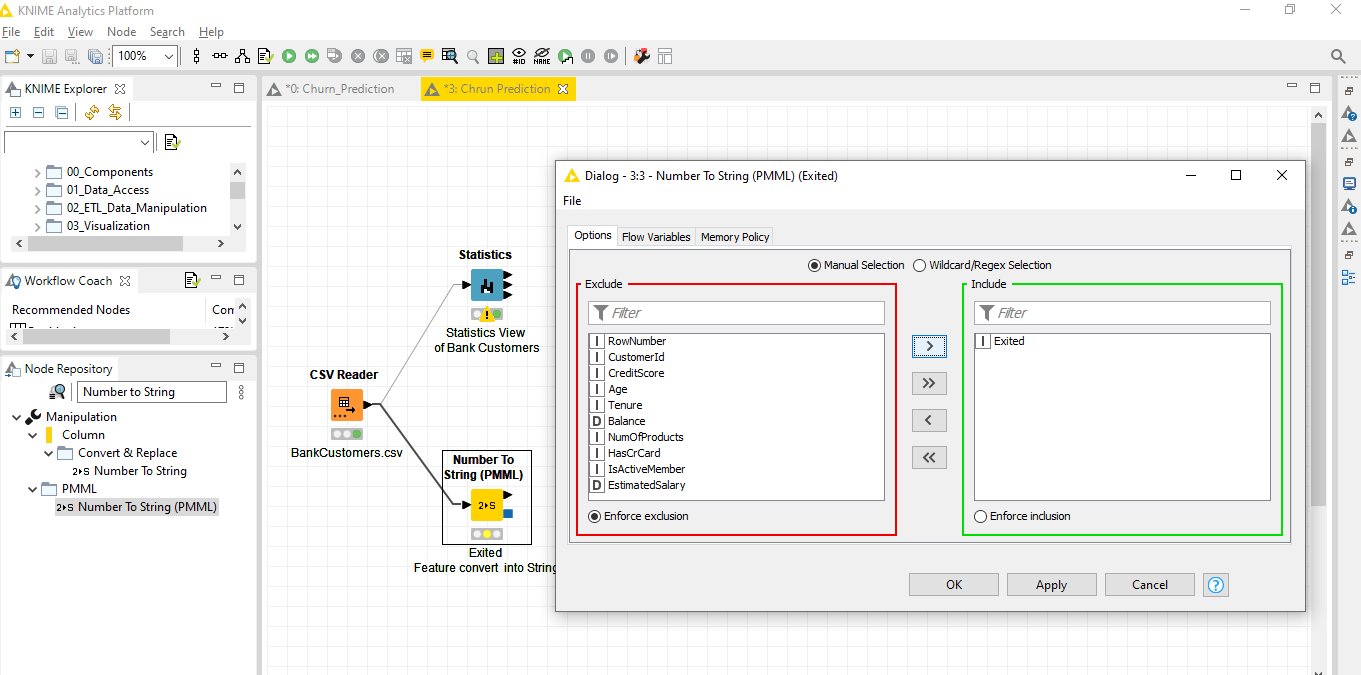
**Step 4– Number to String Node**

To Convert Exited column in to String, by using the Number to String node.

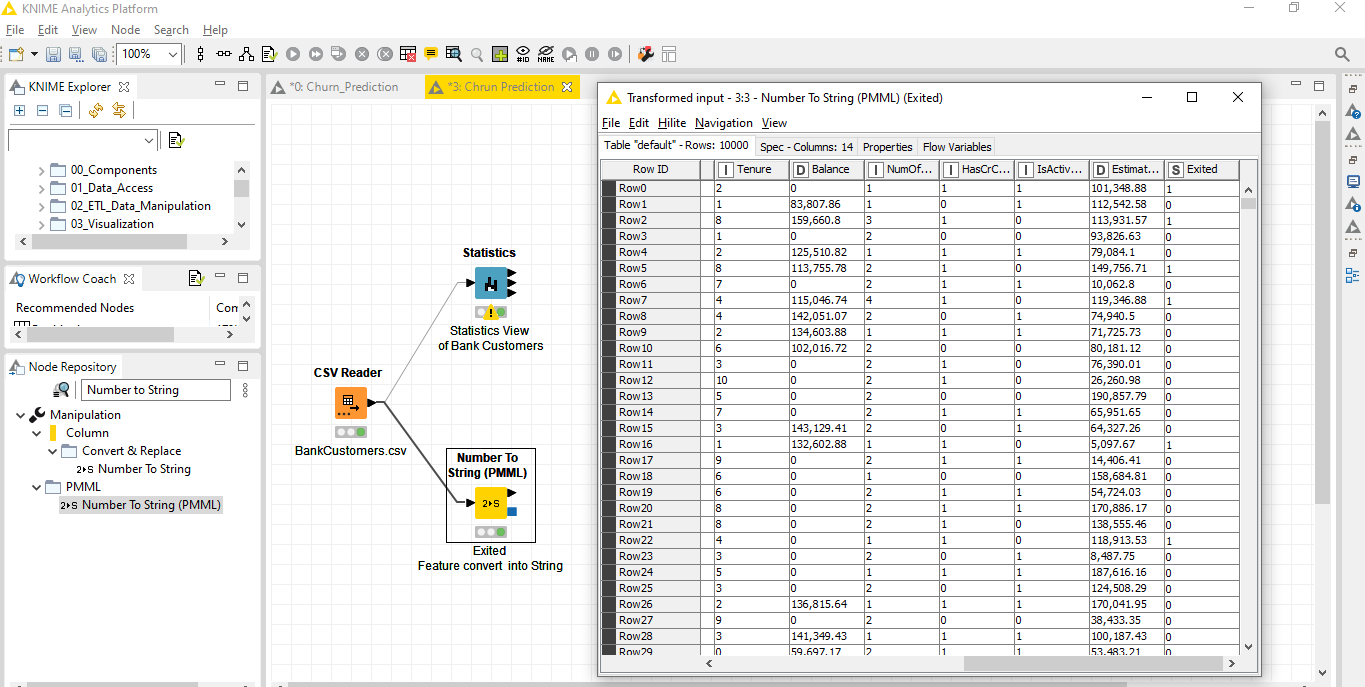
In the node repository panel on the left:

* write “Number to String” in the search field
* drag&drop the node to your workflow editor

To open the configuration dialog, right-click the node and choose Configure. Here, move the “Exited” feature into the Include field on the Right side of the dialog, then click OK.



After executing the node, the transformed input table is available at the output port of this node.

****

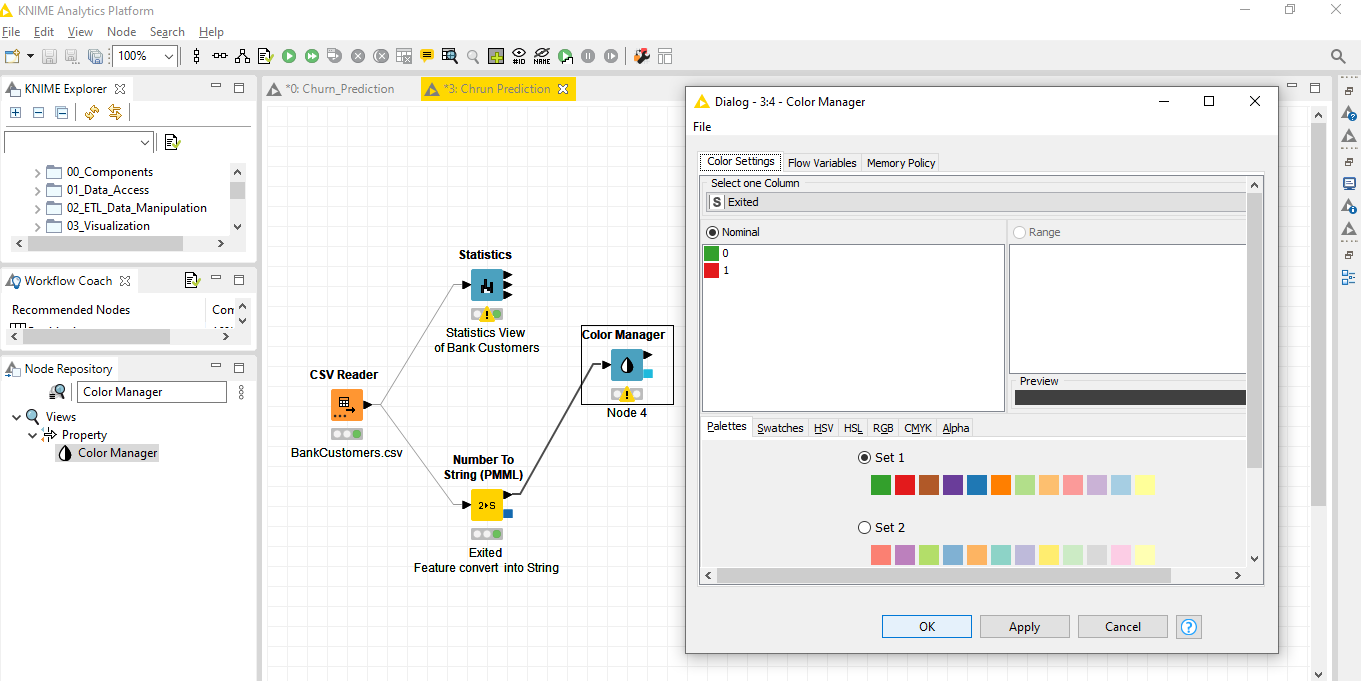
**Step 5– Color Manager Node, Color By Churn**

To set Color by Churn, by using the Color Manager Node.

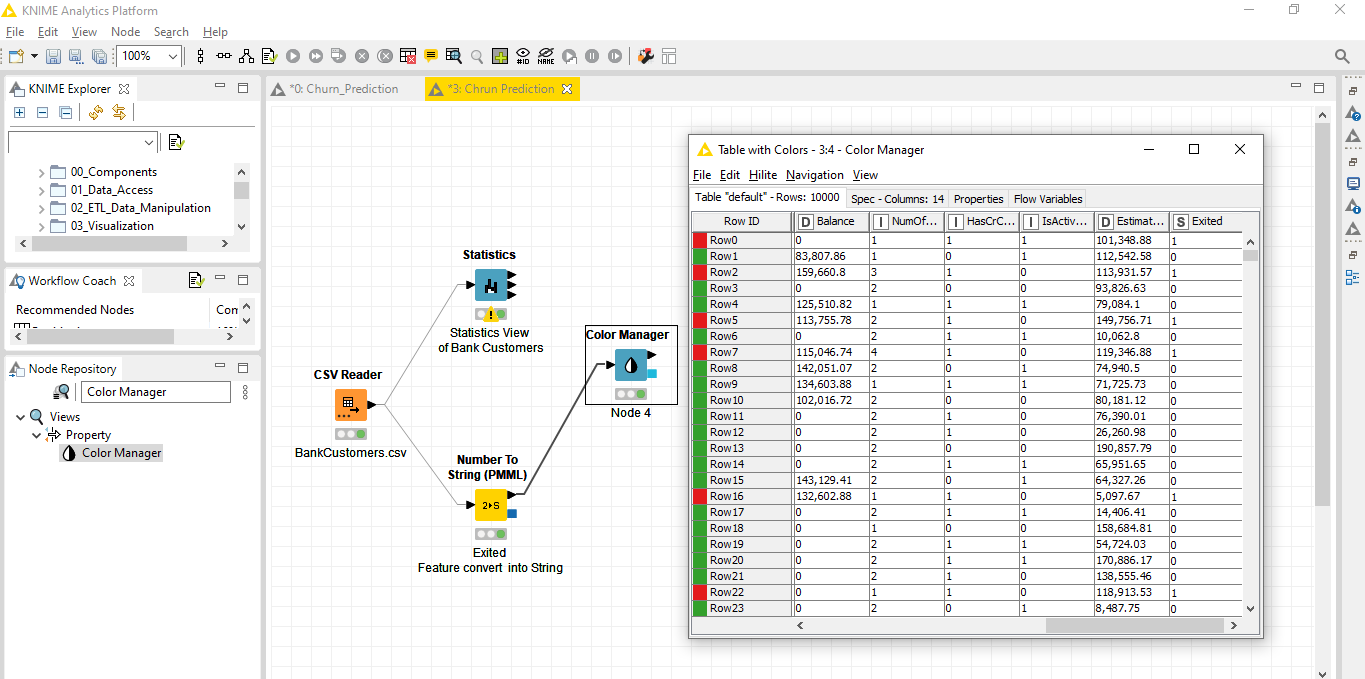
In the node repository panel on the left:

* write “Color Manager” in the search field
* drag&drop the node to your workflow editor

To open the configuration dialog, right-click the node and choose Configure. Here, we set the Color of Nominal Green for 0 and Red for 1 and then click OK.



After executing the node, the table with Color is available at the output port of this node.



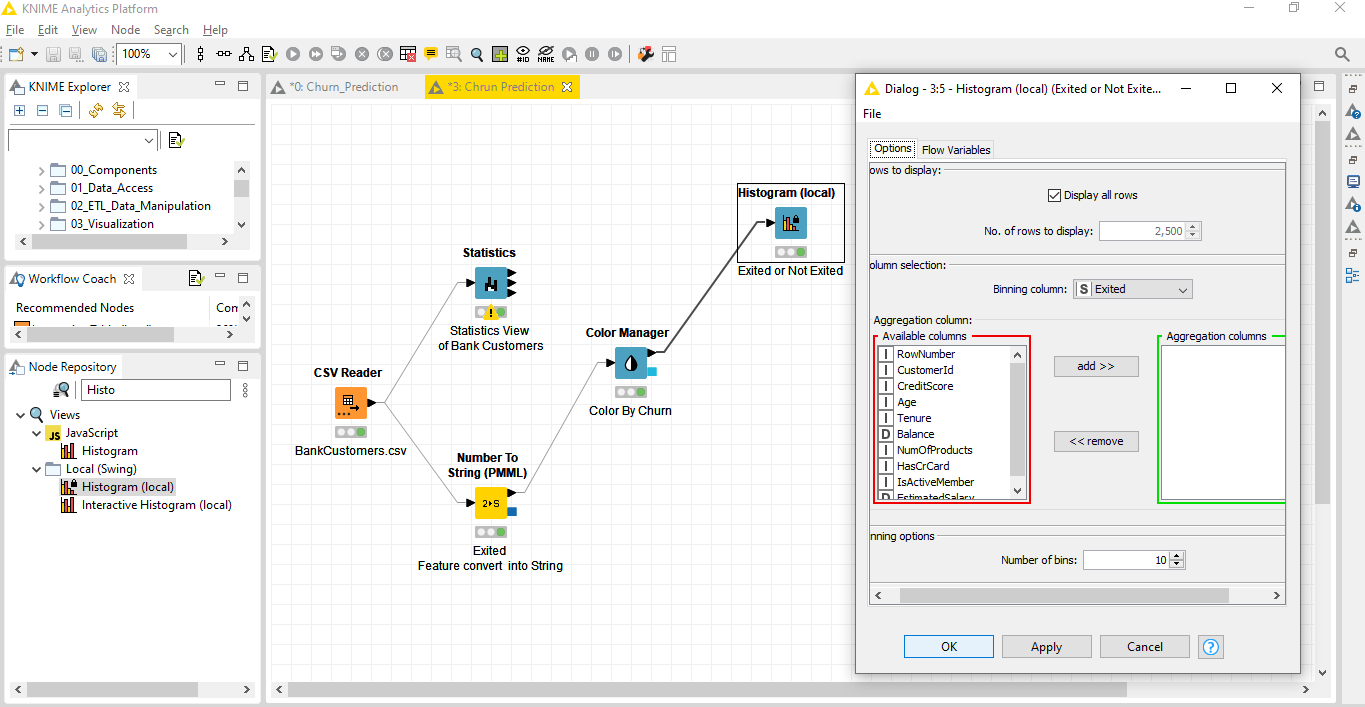
**Step 6–Histogram Node, Count Exited or Not Exited Bank Customers**

To Visualize the Histogram and count the value of Exited Bank Customer=1 and Not Exited Bank Customer= 0, by using the Histogram Node.

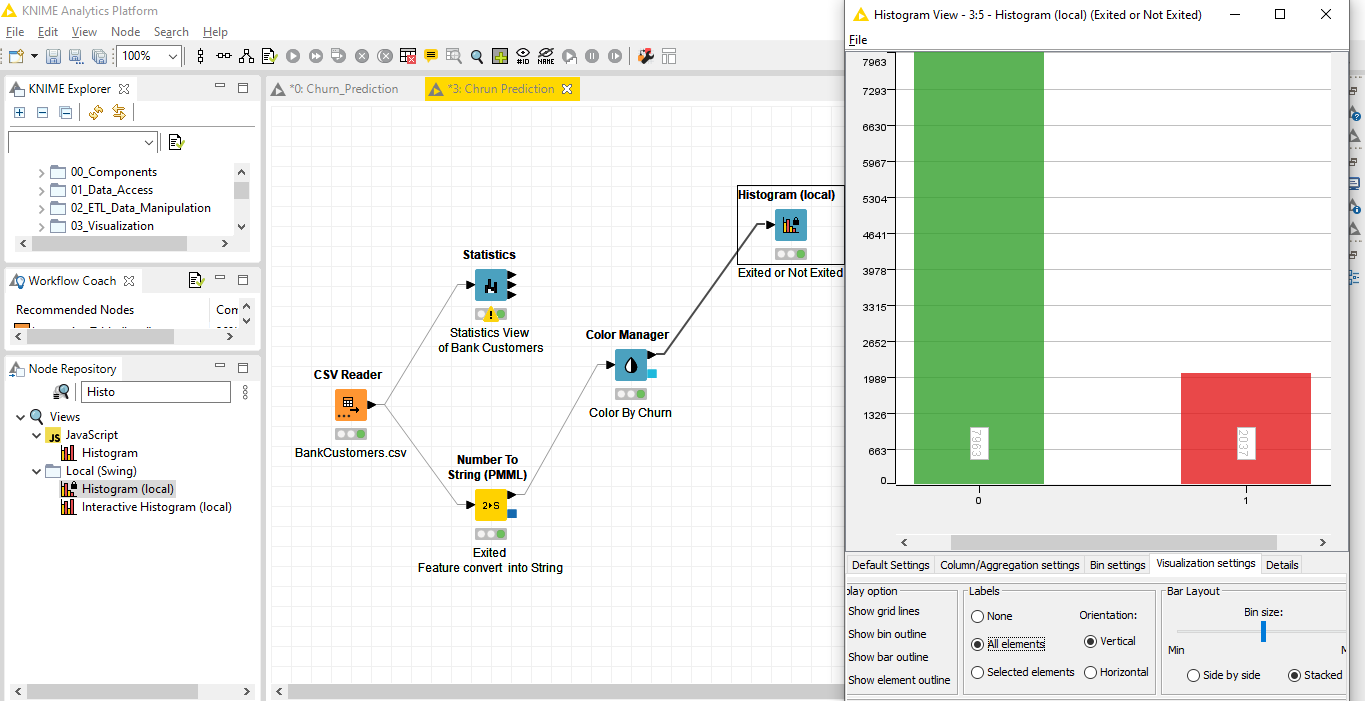
In the node repository panel on the left:

* write “Histogram” in the search field
* drag&drop the node to your workflow editor

To open the configuration dialog, right-click the node and choose Configure. Here we select Binning Column is Exited in Column Selection field, and then click OK.

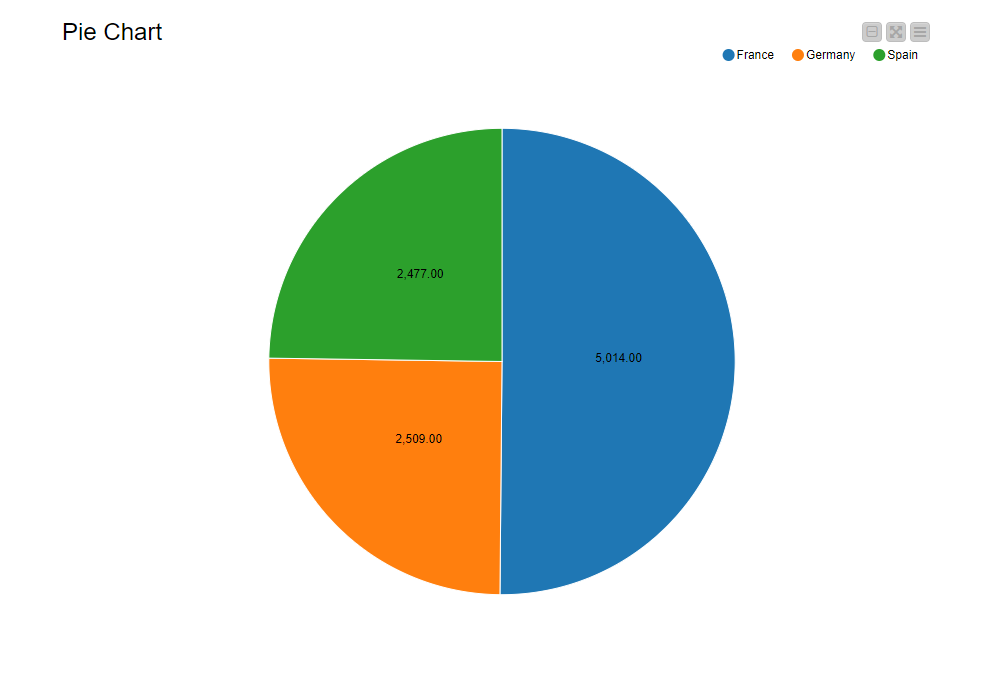


After executing this node, the histogram View is available at the output port of this node. Here we get a count of Bank Exited(1) customer count is 2037 and Bank Not Exited(0) Customer count is 7963.



**Step 7– Pie/Donut Chart Node, Customer’s Geography**

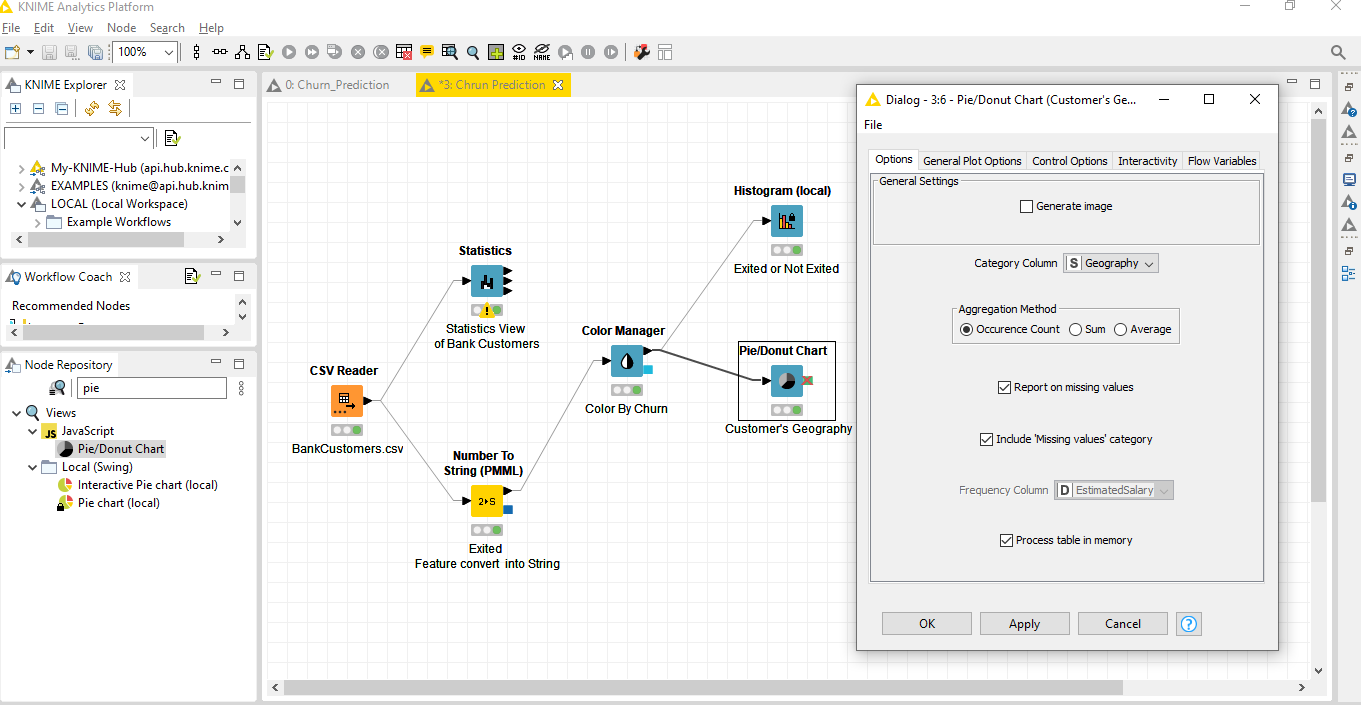
To Visualize the Pie Chart and count the Customer’s Geography by using the Pie/Donut Chart Node.



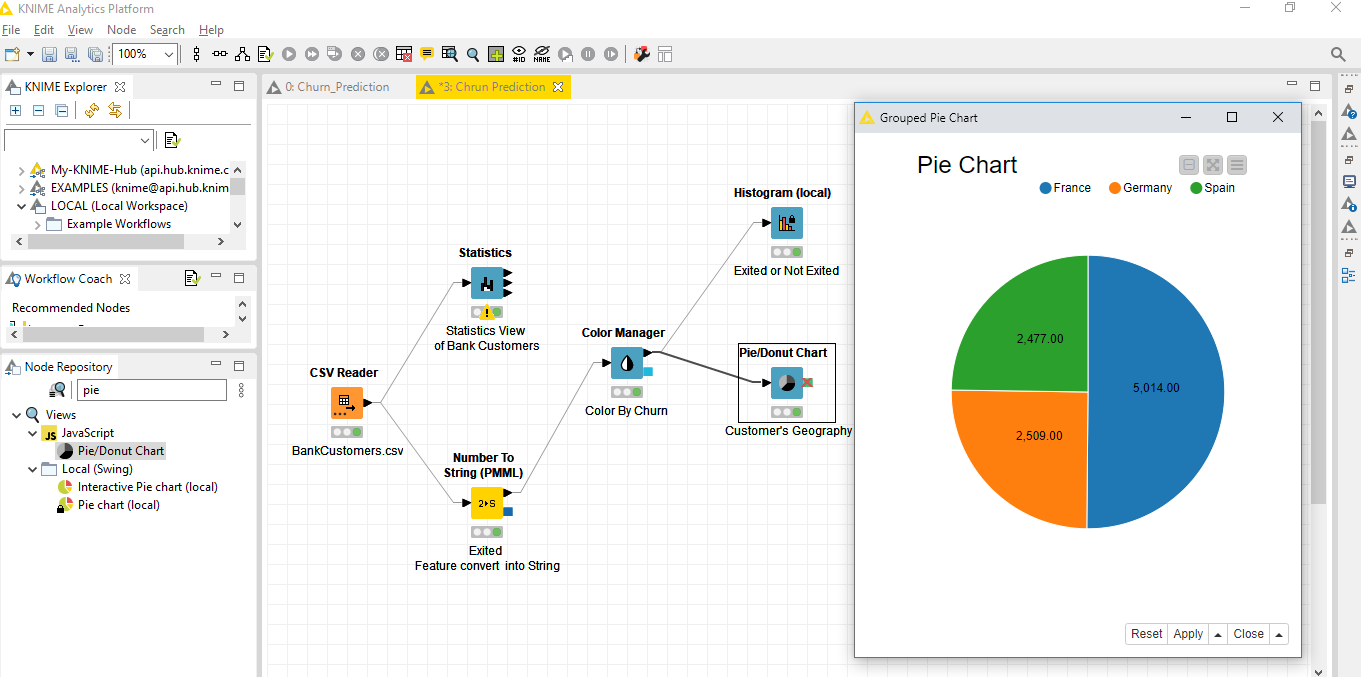
In the node repository panel on the left:

* write “Pie/Donut Chart” in the search field
* drag&drop the node to your workflow editor

To open the configuration dialog, right-click the node and choose Configure. Here we select Category is Geography in Column Selection field, and select Aggregation is Occurrence count and then click OK.



After executing this node here in this Visualization there are three countries like: Spain, Germany, France. Over all Bank customers count of Spain is 2477, Germany is 2509 and France count is 5014.



**Step 8–** **Partitioning Node, Split Data into Train & Test**

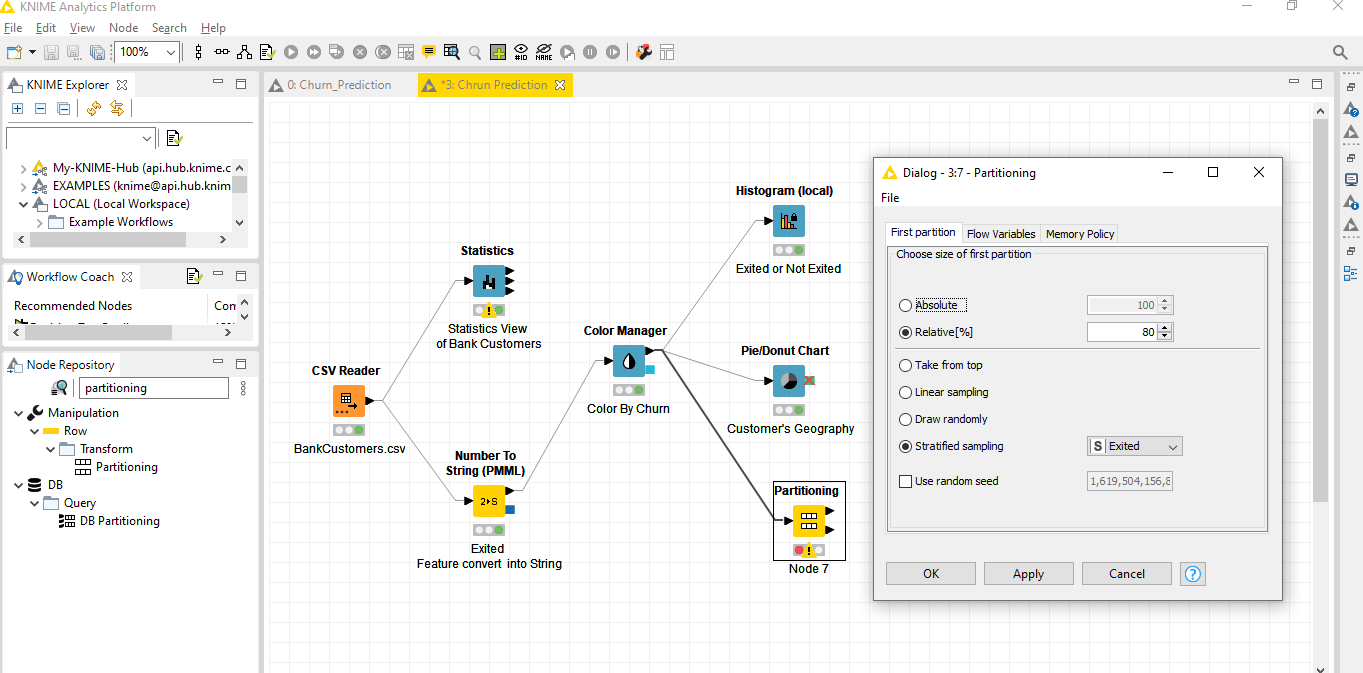
For building a Machine Learning Algorithm we always need train and evaluate (Testing) data. For this reason, the Partitioning node is required to partition most of the data (80%) for training and the small remaining amount (20%) for evaluation (Testing).

To Split the data into Train & Test by using the Partitioning Node.

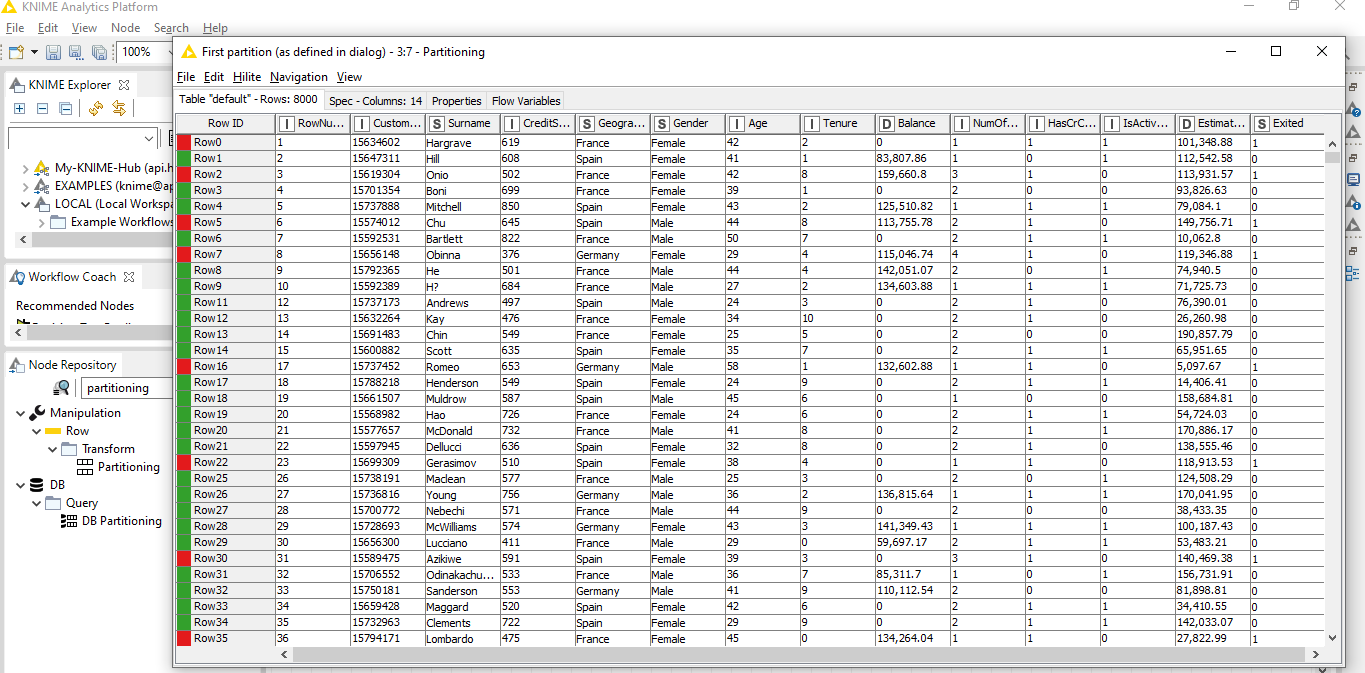
In the node repository panel on the left:

* write “Partitioning” in the search field
* drag&drop the node to your workflow editor

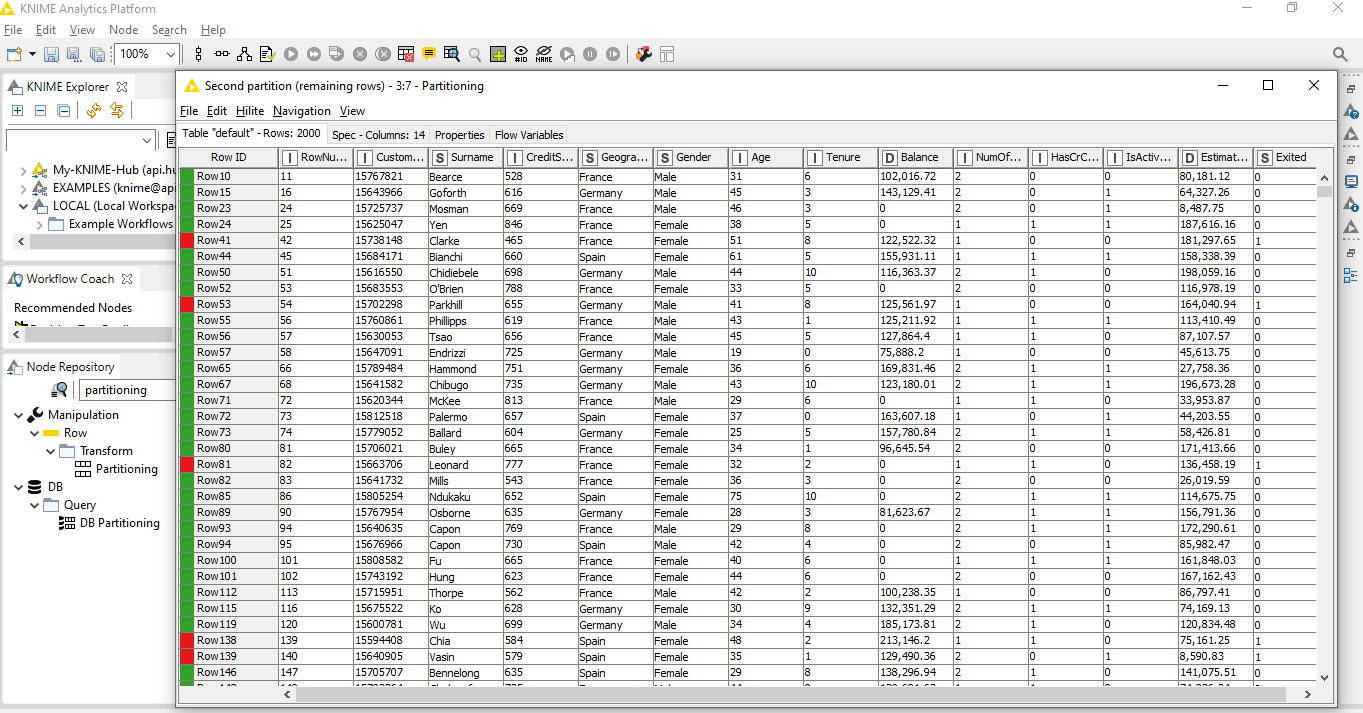
To open the configuration dialog, right-click the node and choose Configure. Here we select First Partition is Relative [%] is 80 % data for training purpose and 20% for model testing. Select stratified sampling is Exited and then click OK.



After executing this node here in this data we get two partitioning **First** Partition and **Second** Partition. In First partition we get 8000 rows and 14 columns like 80% data for training.



In Second partition we get 2000 rows and 14 columns for Testing



**Step 9–** **Decision Tree Learner Node,**

**Train Decision Tree Model**

As usual, we are spoiled for choice when it comes to choosing a machine learning algorithm for training. For use cases, we often deploy a **decision tree** because of its nice tree visualization and highlighting property. However, be aware that you can use any other available machine learning algorithm as long as it produces nominal class-like predictions. For example, **Random Forests**, **Ensemble Trees**, are currently the most frequently adopted machine learning algorithms.

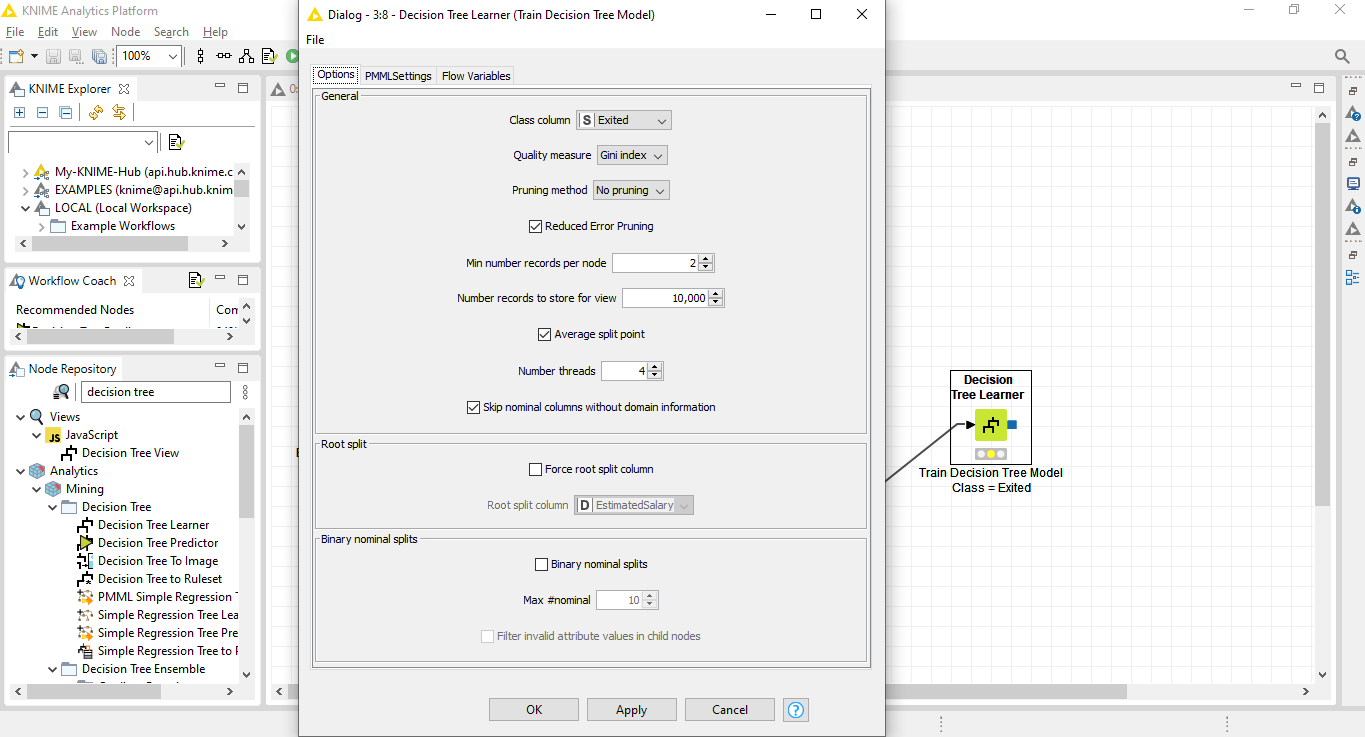
Whatever machine learning algorithm you choose, you always need to train it and evaluate it. For this reason, the Partitioning node is required to partition most of the data (80%) for training and the small remaining amount (20%) for evaluation.

For train a Machine Learning Decision Tree Algorithm we required **Decision Tree Learner node.**

In the node repository panel on the left:

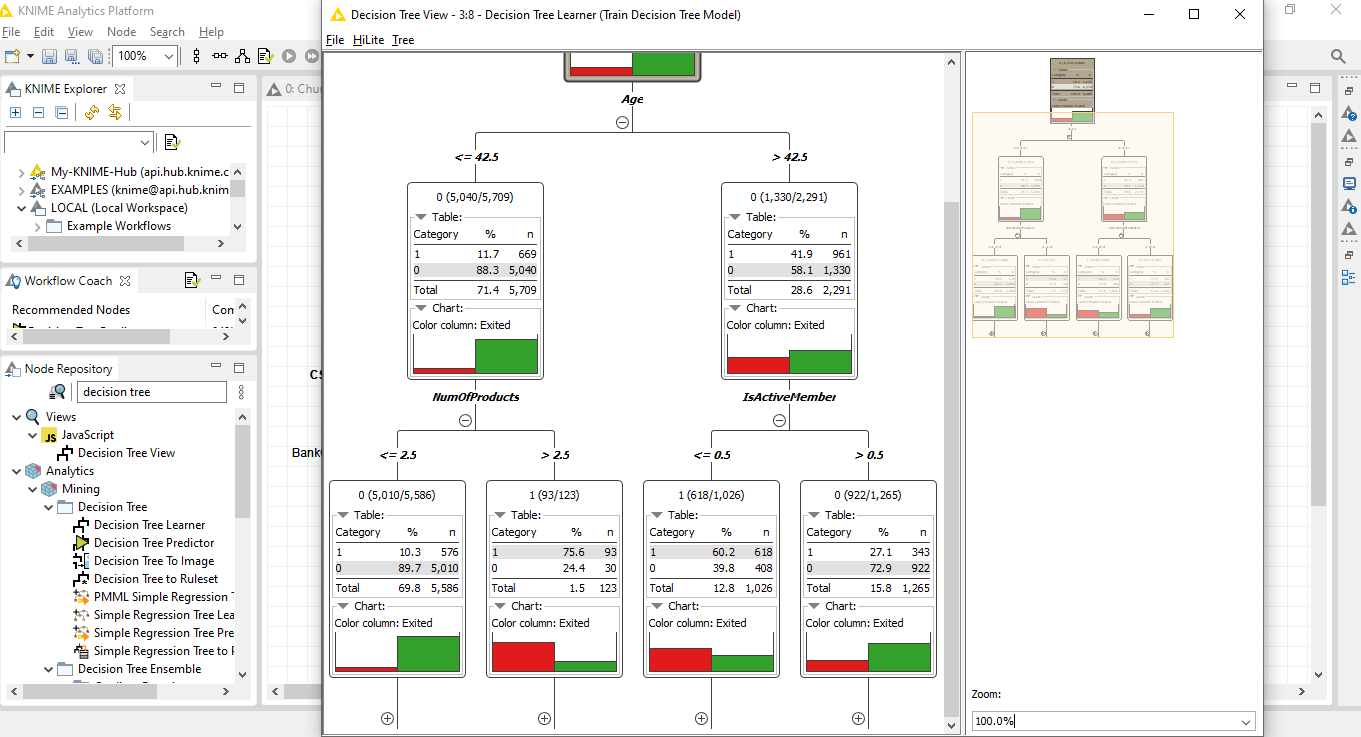
* write “Decision Tree Learner” in the search field
* drag&drop the node to your workflow editor

To open the configuration dialog, right-click the node and choose Configure



To train a decision tree (Decision Tree Learner node), you need to specify the column with the class values to learn (Exited), Gini Index is (quality measure), a pruning method (No pruning), the depth of the tree through the number of records per node (higher number – shorter tree), and the split strategies for nominal and numerical values

After executing this node. At the end of the training phase, the “View” option in the node context menu shows the decision path through the tree to reach leaves with churning and not churning customers.



**Step 10–** **Decision Tree Predictor**

**Model Evaluation:**

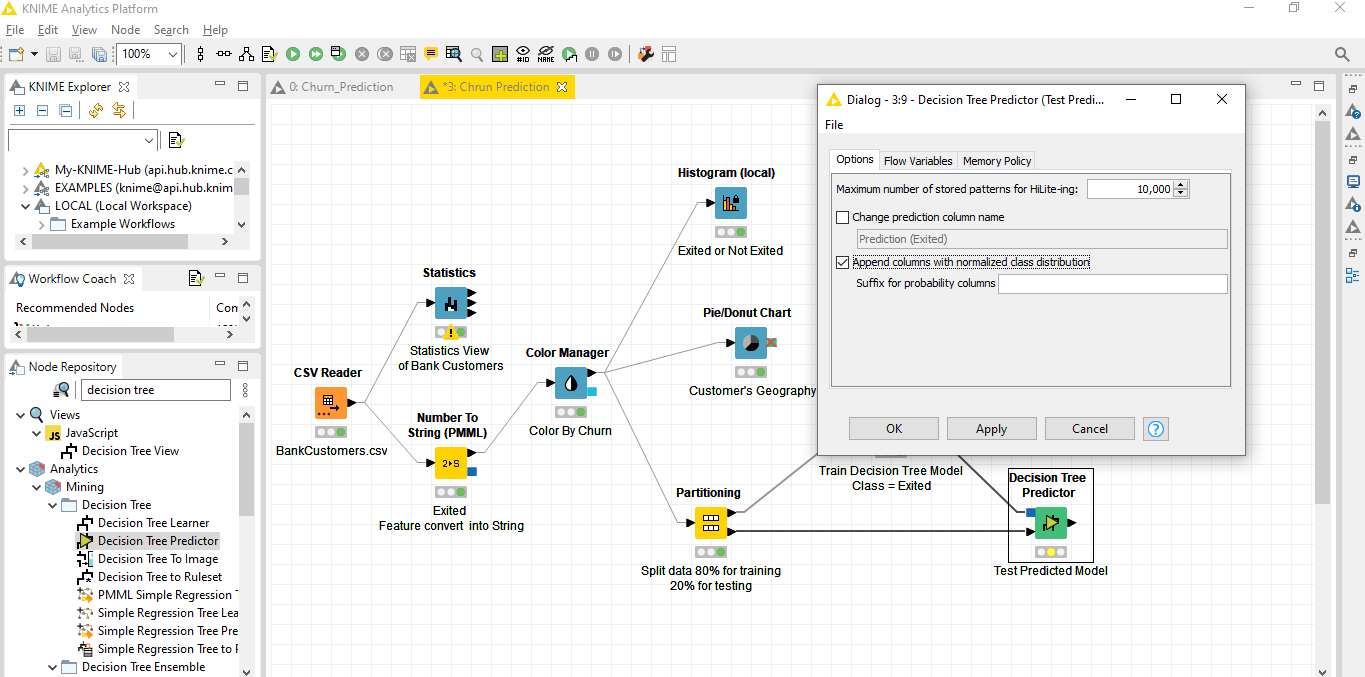
We need to evaluate it before running it for real on real data. For the evaluation, we use that 20% of data we have kept aside and not used in the training phase, to feed a Decision Tree Predictor node. This node applies the model to all data rows one by one and produces the likelihood that customer has of Exited or Not, given contract and operational data (P(Exited=0/1)). Depending on the value of such probability, a predicted class will be assigned to the data row (Prediction (Exited) =0/1).

For Evaluating Trained Machine Learning Model by using the Decision Tree Predictor Node.

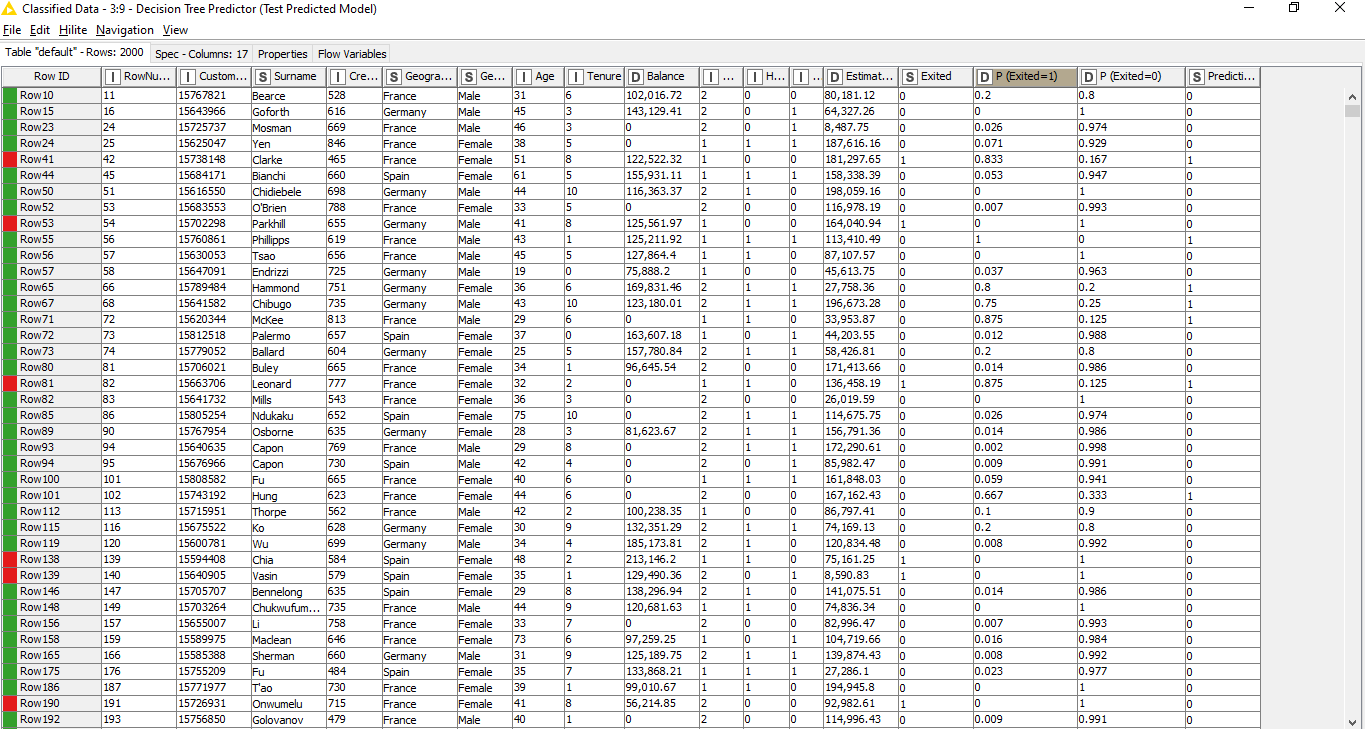
In the node repository panel on the left:

* write “Decision Tree Predictor” in the search field
* drag&drop the node to your workflow editor

To open the configuration dialog, right-click the node and choose Configure. Here we check the append columns with normalized class distribution and then click ok.



After executing this node, we get three new columns like: Predict (Exited =1), Predict (Exited =0) and prediction Exited.



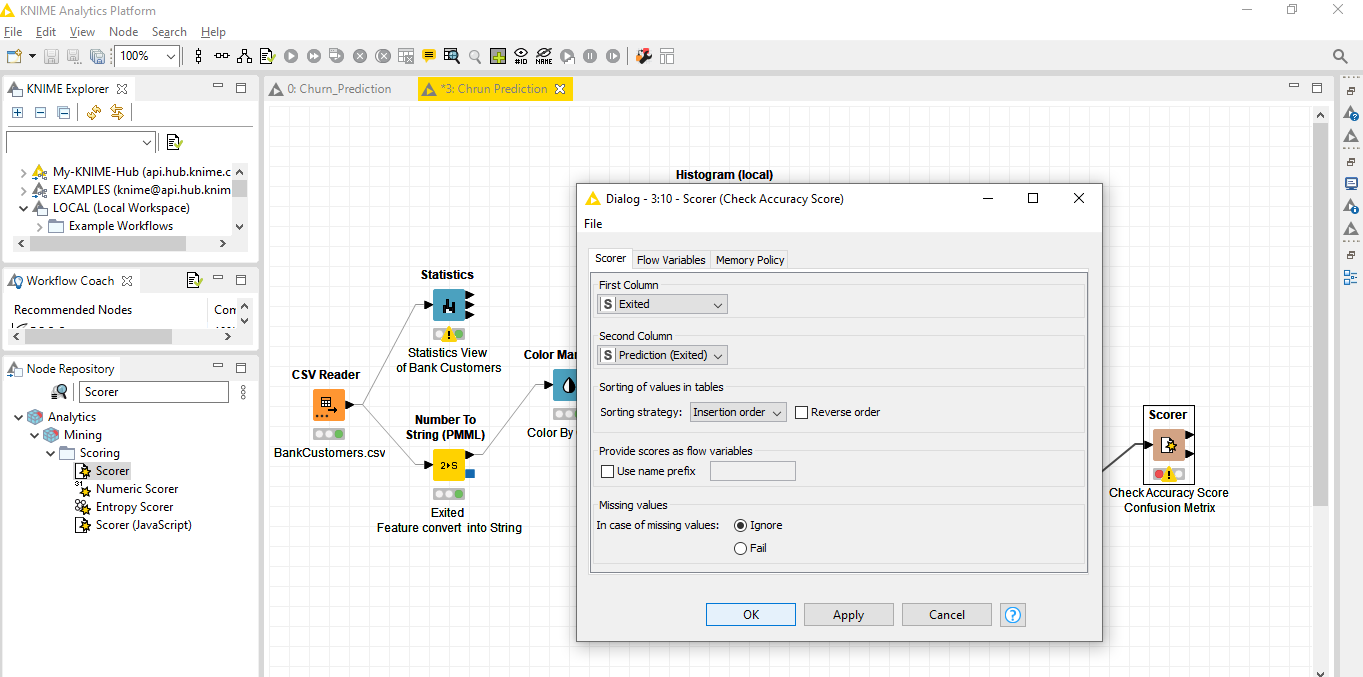
**Step 11–** **Scorer Node, Check the Accuracy Score & Confusion Metrix**

For Check the Model Accuracy Score and Confusion Metrix by using the Scorer Node.

In the node repository panel on the left:

* write “Scorer” in the search field
* drag&drop the node to your workflow editor

To open the configuration dialog, right-click the node and choose Configure. Here we select the first column as Exited and Second column is Prediction Exited and then click ok.



After executing this node, we get 82% Model Accuracy. Now here we have True Positive value is 204, False positive Value is 203, Ture Negative value is 1436 and False Negative Value is 157.

**Step 12–PMML Writer Node**

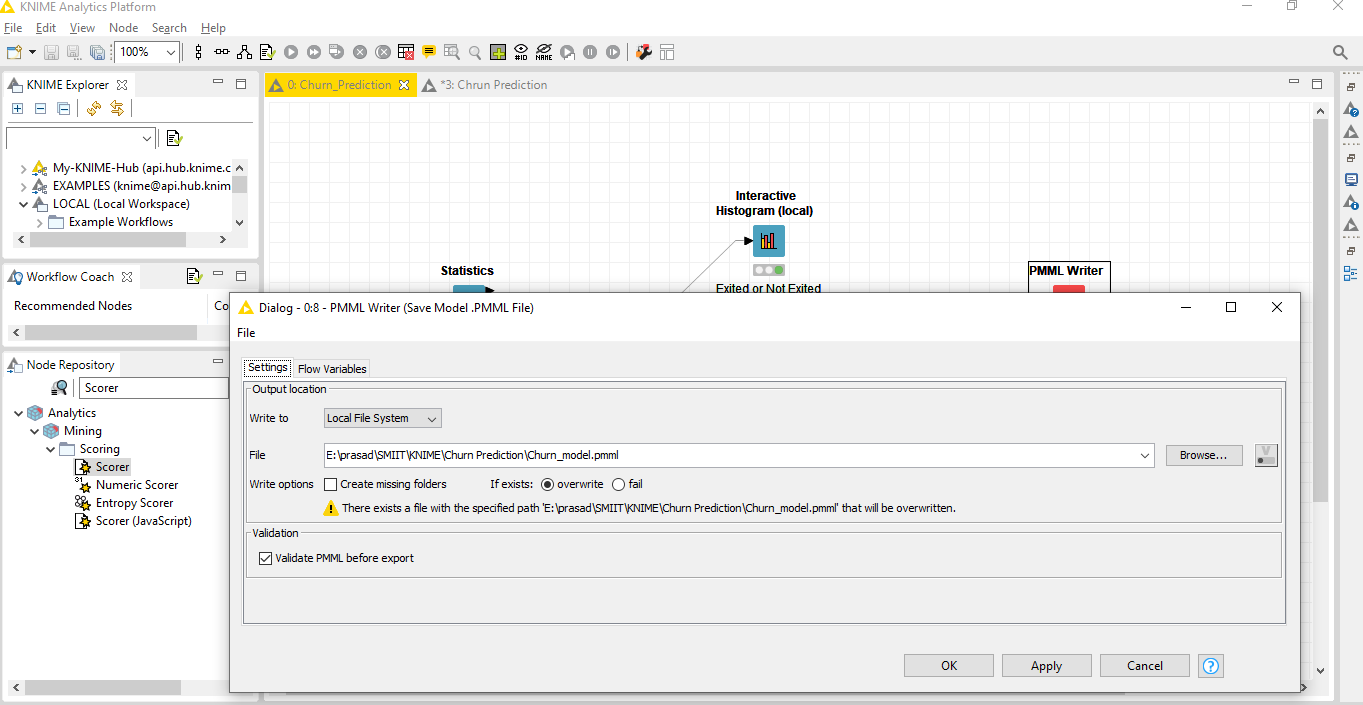
**Save the model in .PMML file**

For Save the Predicted Model using the PMML Writer Node.

In the node repository panel on the left:

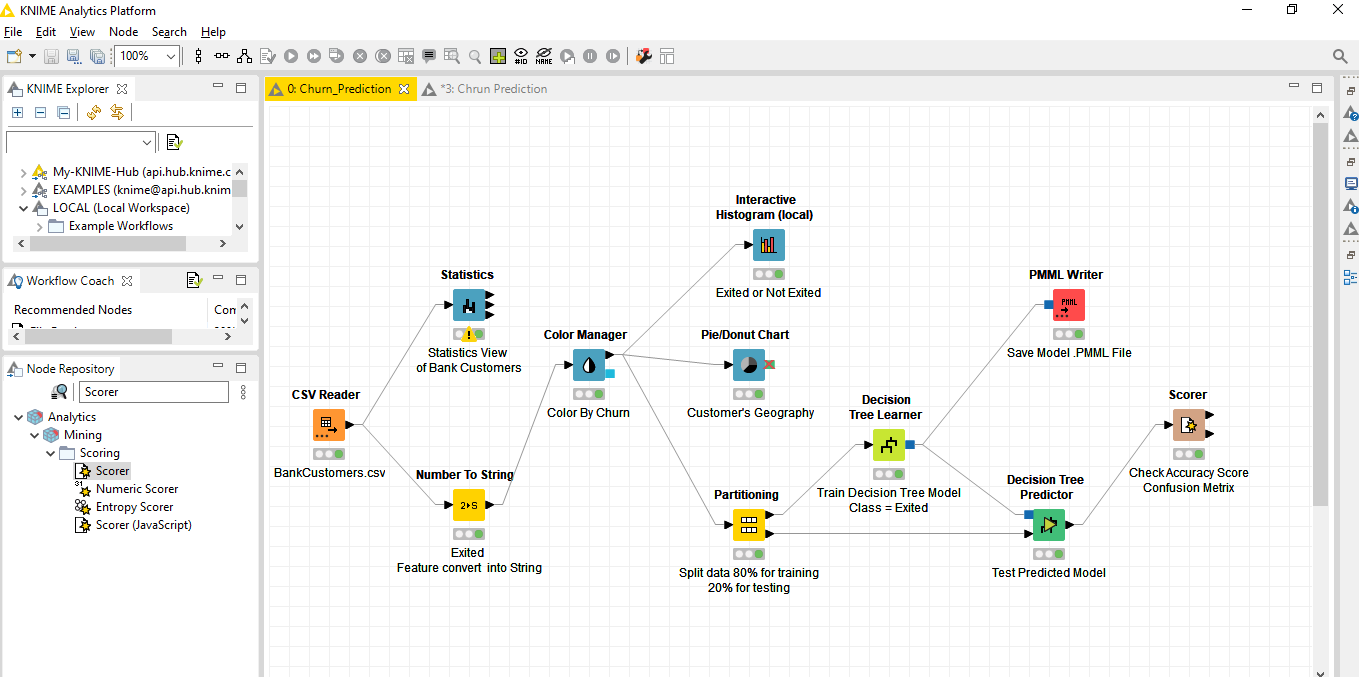
* write “PMML Writer” in the search field
* drag&drop the node to your workflow editor

To open the configuration dialog, right-click the node and choose Configure. Here we select set the path where we want to save the model and then click ok.



After executing this node our predicted model Save in .PMML file

**Final View of all connected Nodes**



**Deployment**

When we are satisfied with our model performance, we can move it into production for deployment on real data. Here we need only read the stream of real-life data coming in through a file or database or whatever other data source and the generated model. We then apply a Decision Tree Predictor, a PMML Predictor to run the model on the real-life input data. 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, if so specified in the predictor configuration settings.

Please note that a PMML Predictor will make you independent of the selected machine learning model!

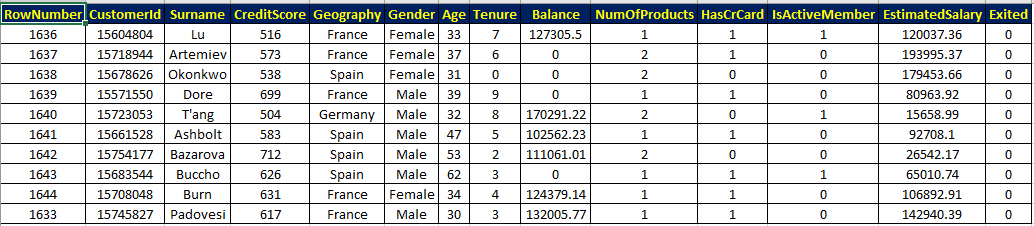
If your model needs some data preprocessing, this can also be added to the PMML model using a PMML-compatible data manipulation node in the training workflow.

Also notice that, if the data preparation part has been incorporated into the PMML model, the only really necessary node in this deployment productive workflow, besides the reader nodes, is the predictor node.

This workflow is an example of how to deploy PMML Model (built in workflow Churn Prediction) for churn prediction.

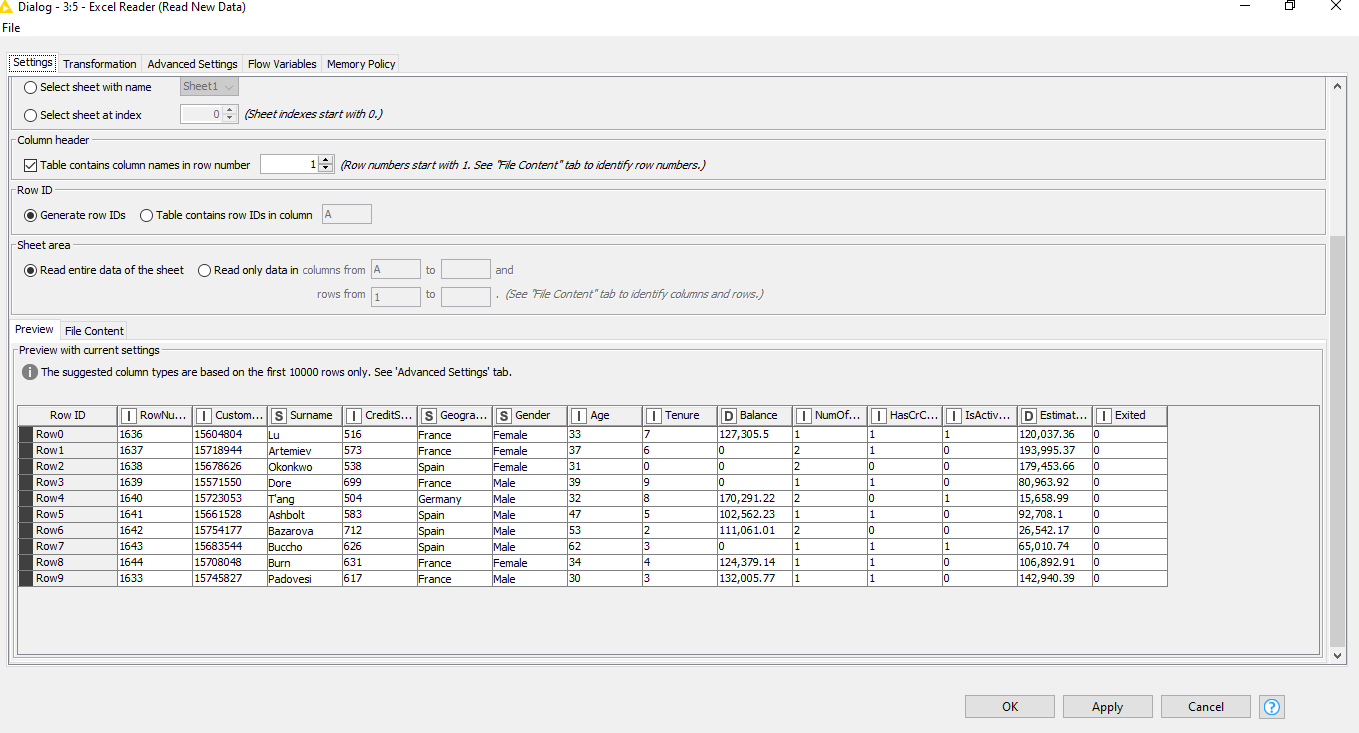
Using PMML you only need 4 nodes for the whole workflow to export data for the report. PMML is transparent to the Decision Tree model the PMML Predictor node understands all.

**Read New Data for Model Deployment:**



**Step 13 – Drag and drop New Data file into workbench editor**

Drag and drop the new data TestBankCustomer.xlsx into the workbench editor. Excel Reader node will appear on the workflow editor and its configuration dialog will pop-up.

After select configure then will open below dialog box. Here we shown head of the data frame rows and columns. Click on OK.

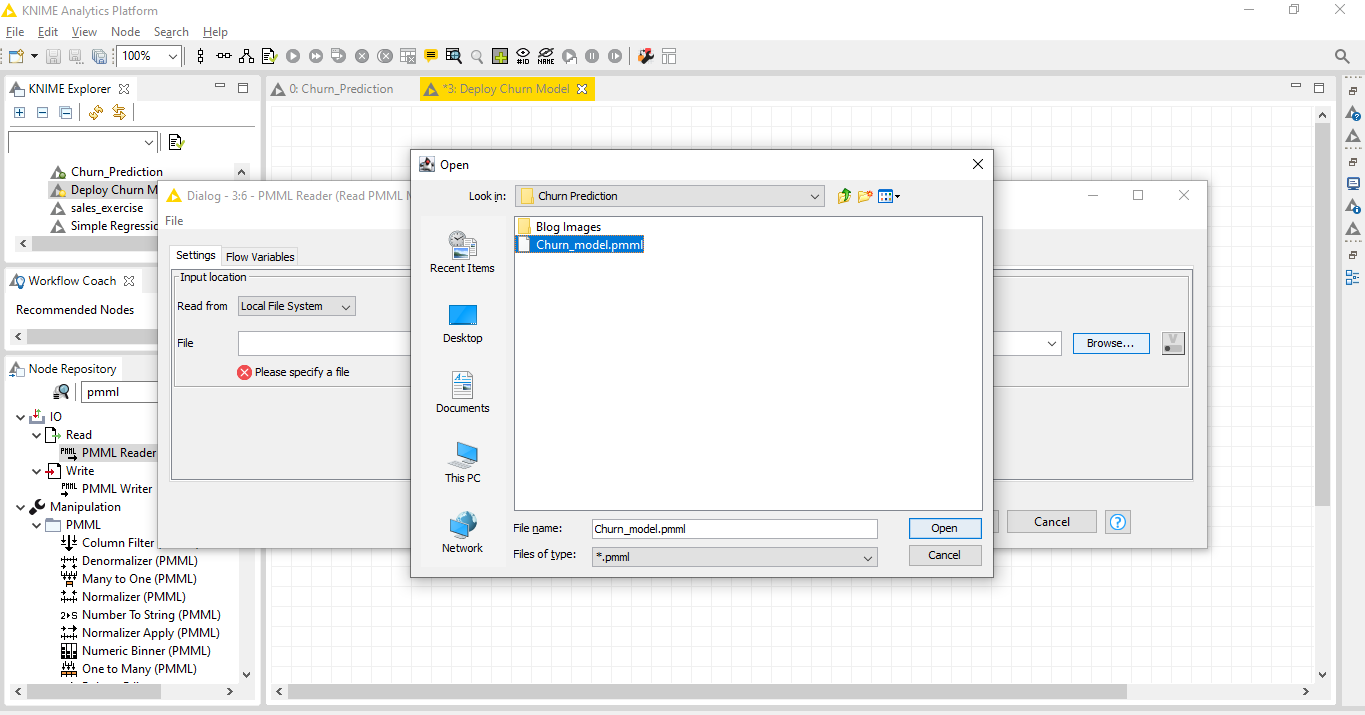
**Step 14–PMML Reader Node, Load the model**

For Load the Predicted Model using the PMML Reader Node.

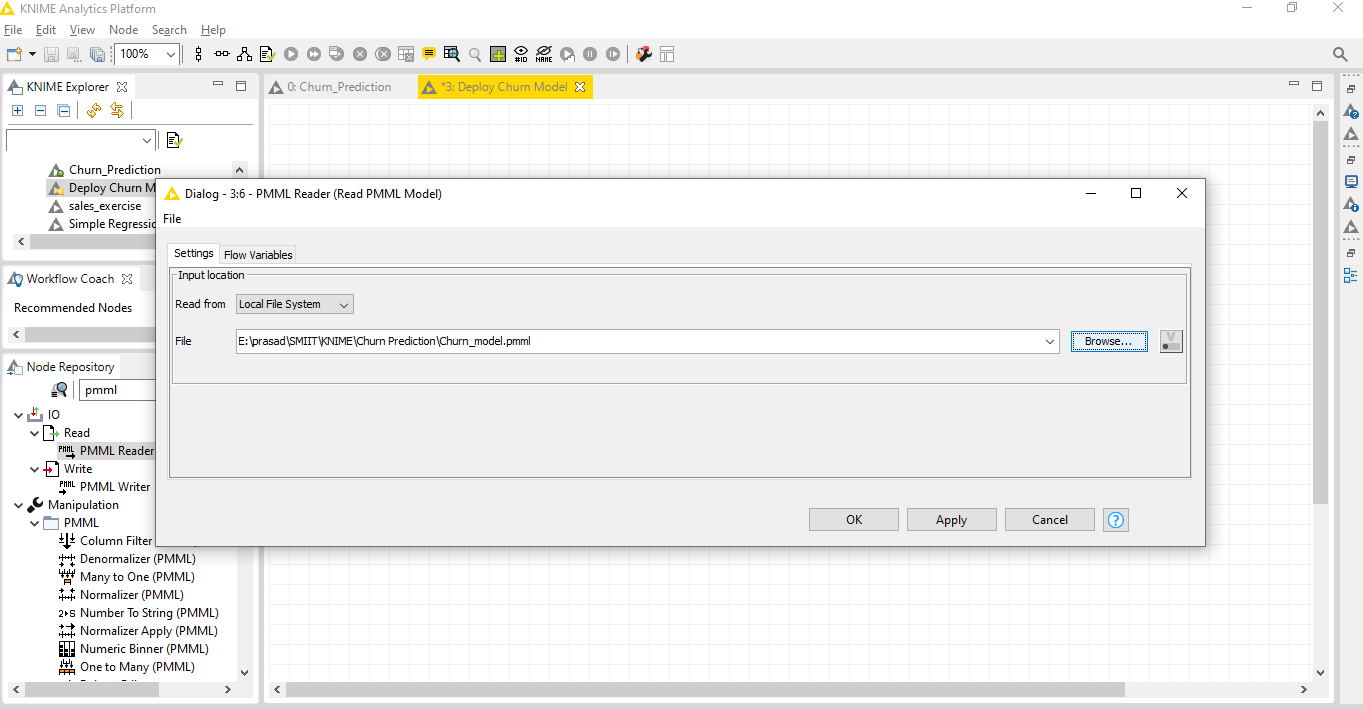
In the node repository panel on the left:

* write “PMML Reader” in the search field
* drag&drop the node to your workflow editor

To open the configuration dialog, right-click the node and choose Configure. Here we Load our predicted model from existing path and then click ok.



After executing this node our predicted model will be loaded on the workflow.



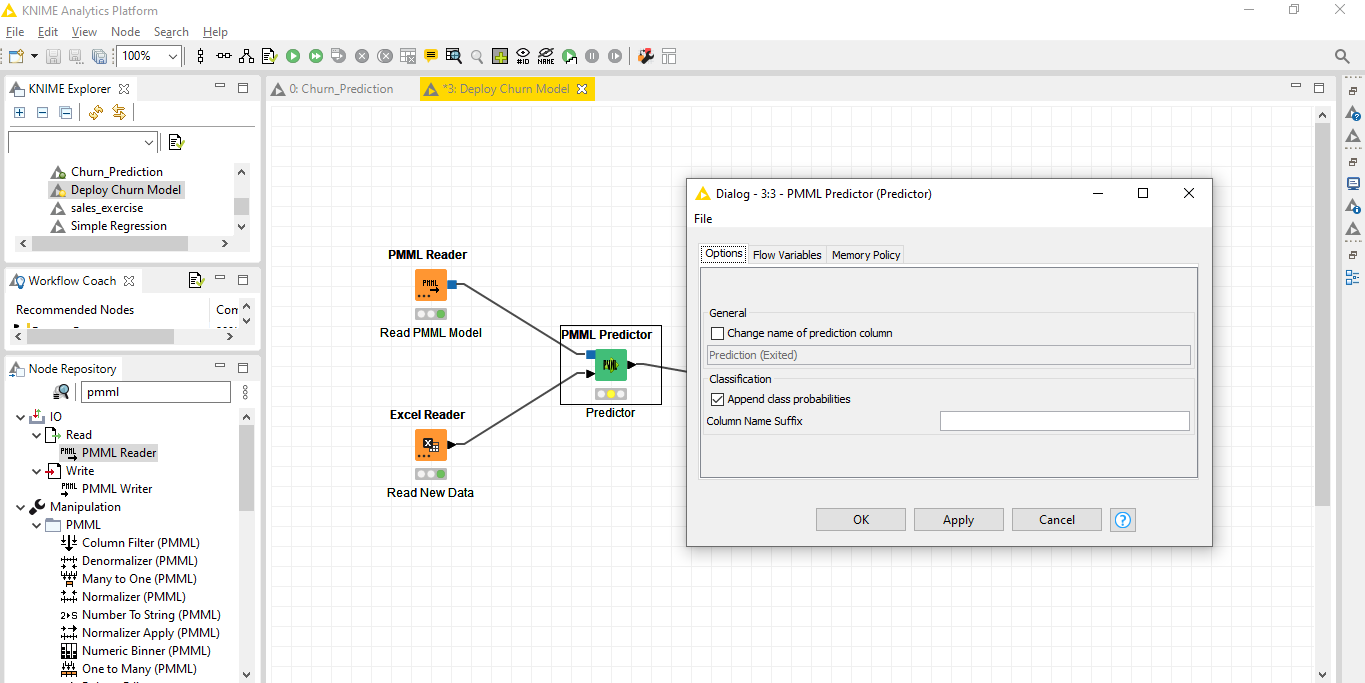
**Step 15–PMML Predictor Node**

Predict customer will Exit bank or not that’s way we use PMML predictor Node.

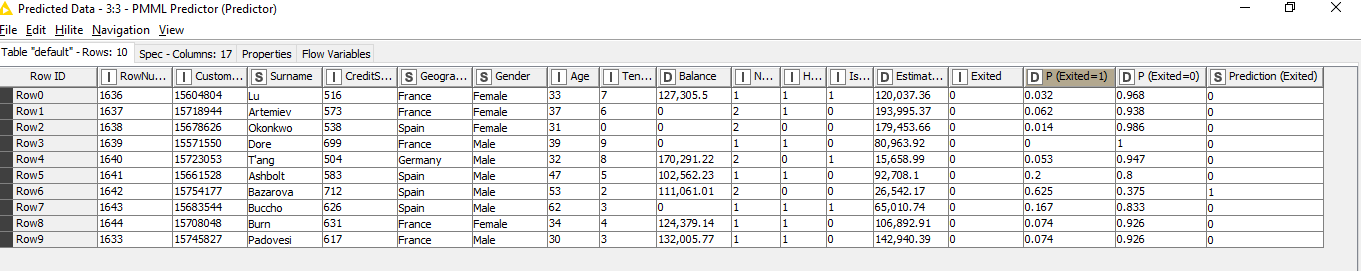
In the node repository panel on the left:

* write “PMML Predictor” in the search field
* drag&drop the node to your workflow editor

To open the configuration dialog, right-click the node and choose Configure. Here we check append class probabilities then click ok.



After executing this node, we get three new columns like: Predict (Exited =1), Predict (Exited =0) and prediction Exited.



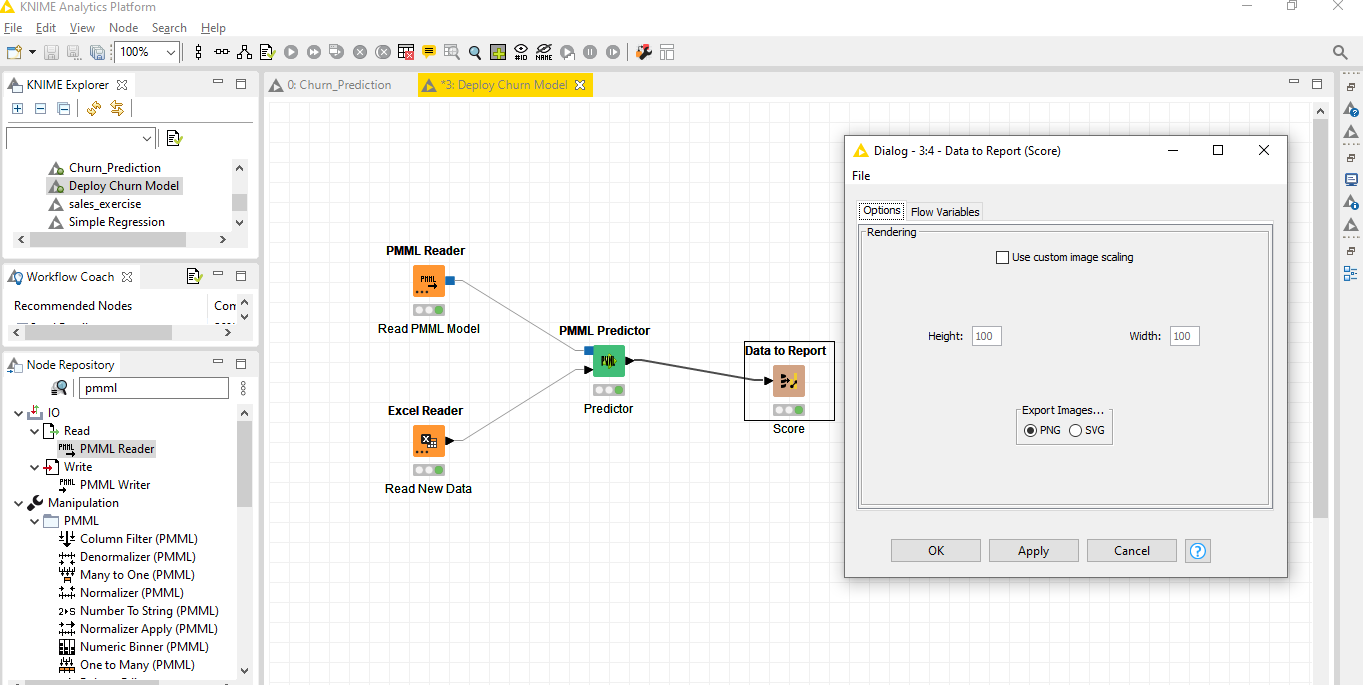
**Step 16–Data To Report Node**

View the predicted report of customer will Exit bank or not that’s way we use this Node.

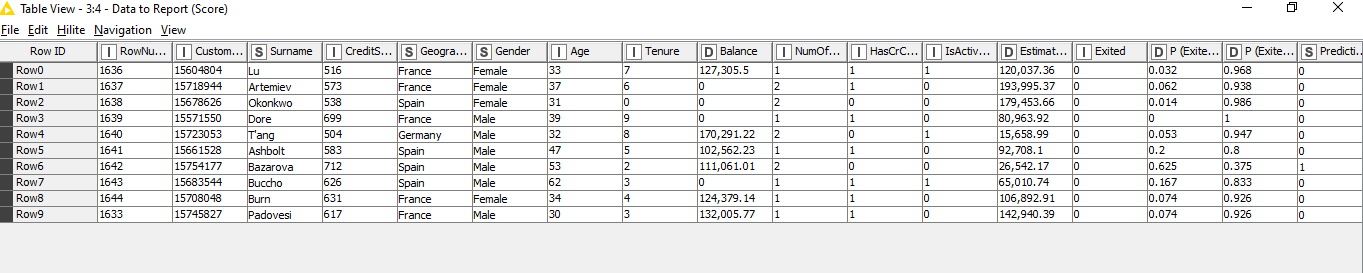
In the node repository panel on the left:

* write “data to report” in the search field
* drag&drop the node to your workflow editor

To open the configuration dialog, right-click the node and choose Configure. Here we check append class probabilities then click ok.



After executing this node, we get three new columns like: Predict (Exited =1), Predict (Exited =0) and prediction Exited.



These Blog and KNIME Workflow is available on our GitHub.

**Blog Site:** <https://www.smiit.xyz/churn-prediction-using-knime-tool/>

**GitHub Link:** <https://github.com/SMIIT-Projects/Churn-Prediction-Using-KNIME-Tool>

## Tools and Technologies:

## KNIME 4.3.2 (64bit)

Thank You!