**BANK NOTE AUTHENTICATION**

* **SAKSHAM YADAV ( 18CSU187 )**
* **SALONI YADAV ( 18CSU188 )**

**ABOUT THE DATASET**

*The data file Bank\_Note\_Authentication.xls is the source of information for the classification problem.*

*The number of instances (rows) in the data set is 1372, and the number of variables (columns) is 5.*

*In that way, this problem has the following variables:*

◆ *Variance, used as input*

◆ *Skewness, used as input*

◆ *Curtosis, used as input*

◆ *Entropy, used as input*

◆***Class, used as target. It can only have two values: 0 (false) or 1 (true).***

**ABOUT THE PROJECT**

*The instances are divided into training and testing subsets. This is a classification project, since the variable to be predicted is binary (fraudulent or legal).*

*The goal here is to model the probability that a banknote is fraudulent, as a function of its features.*

**STEPS FOLLOWED**

* **Importing the dataset**
* **Doing the data pre-processing**

*Handling the missing values if any*

*Converting the Target feature values to nominal values*

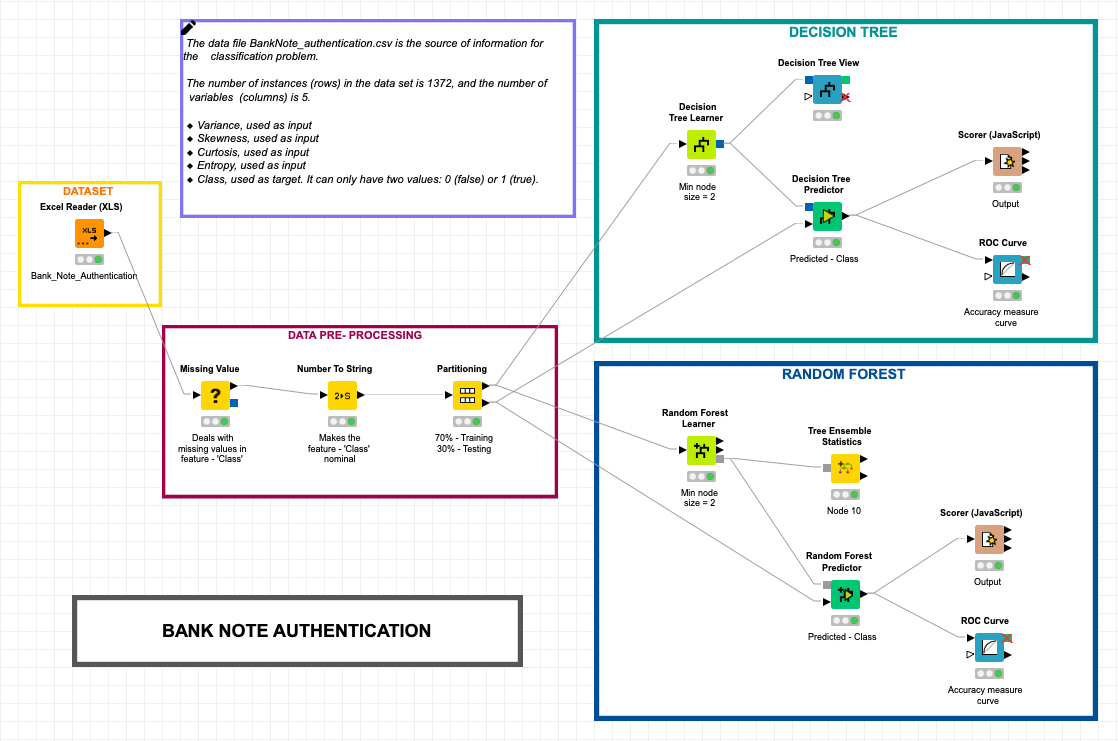
*Partitioning the dataset for training and testing processes*

* **Implementing the models**

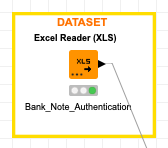
*Decision tree*

*Random forest*

**KNIME WORKFLOW**

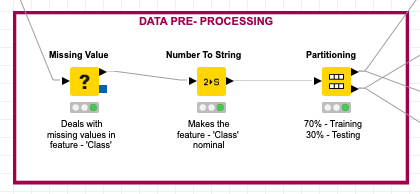
**

**NODES USED**

* **DataSet**
* ***Excel Reader***

Reads the data from 1 file

*My file is Bank\_Note\_Authentication.xls*

* **Data Pre-Processing**
* ***Missing Value***

Handles the missing value present in any feature.

*Here, I have replaced the Missing Values of a feature with the mean of that feature.*

* ***Number To String***

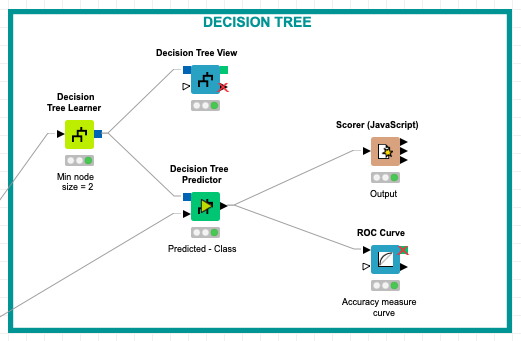
Converts the numerical value to a categorical value.

*Here, I have converts the ‘Class’ feature values from Double to String*

* ***Partitioning***

Partitions the dataset into 2 sets. One for training process & other for testing process.

*Here, I have split my dataset into ratio of 7:3, i.e. 70% for training the model and 30% for testing the model.*

* **Decision Tree Model**
* ***Decision Tree Learner***

Trains the model on the training dataset.

*Here, 70% of my training dataset is used for this process.*

* ***Decision Tree View***

Plots the view of the decision tree.

* ***Decision Tree Predictor***

Applies the model to the test dataset for prediction. One input node is attached to Decision tree learner node and other one to testing dataset.

*Here, my predicted feature is ‘Class’.*

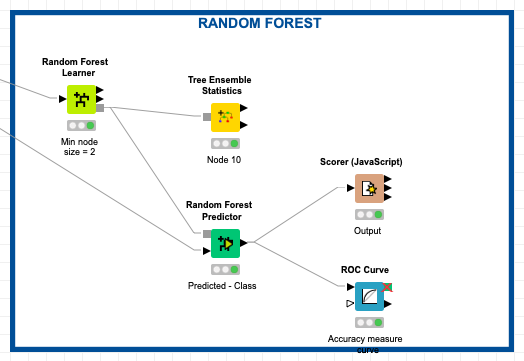
* ***Scorer ( JavaScript )***

Evaluate the accuracy of the model.

*My model gives the accuracy of 98.79% with error of 1.21%*.

* ***ROC Curve***

Shows the classification model performance as described by FPR & TPR.

* **Random Forest Model**
* ***Random Forest Learner***

Trains the model on the training dataset.

*Here, 70% of my training dataset is used for this process.*

* ***Tree Ensemble Statistics***

Provides basic statistics on the tree ensemble and its trees such as depth of the trees and the number of nodes in the trees.

* ***Random Forest Predictor***

Predicts the output. One input node is attached to Decision tree learner node and other one to testing dataset.

*Here, my predicted feature is ‘Class’.*

* ***Scorer ( JavaScript )***

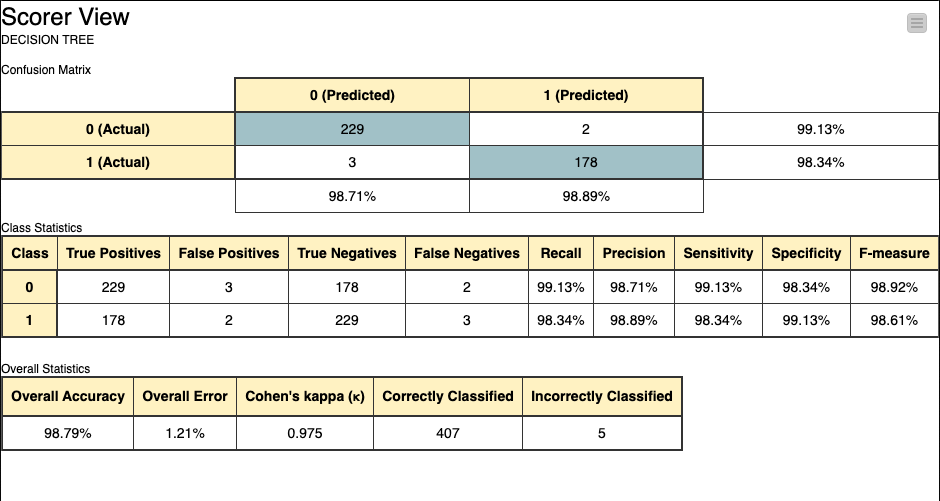
Evaluate the accuracy of the model.

*My model gives the accuracy of 99.27% with error of 0.73%.*

* ***ROC Curve***

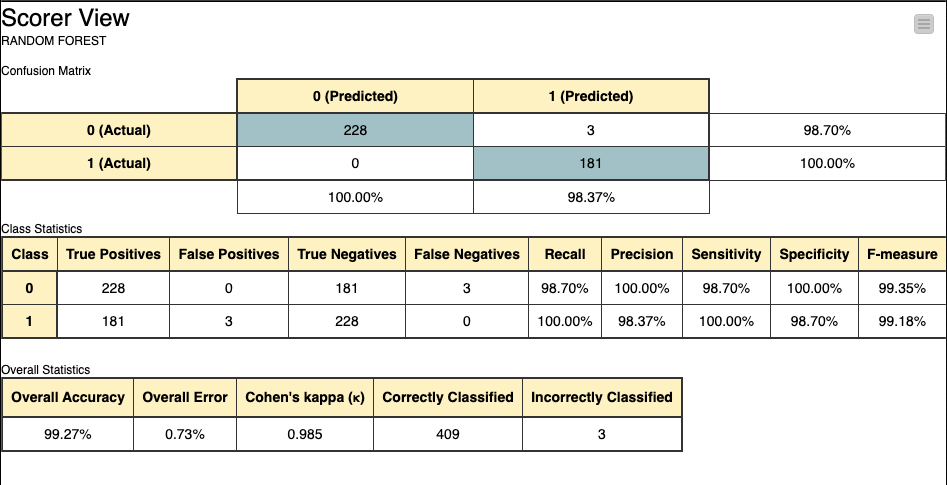
Shows the classification model performance as described by FPR & TPR.

**OUTPUT OF THE 2 MODELS**

* ***DECISION TREE***

***Accuracy - 98.79%***

***Error - 1.21%***

* ***RANDOM FOREST***

***Accuracy - 99.27%***

***Error - 0.73%***

**COMPARING THE 2 MODELS**

***ON COMPARING BOTH THE MODELS BUILT, WE SEE RANDOM FOREST DOMINATES OVER DECISION TREE.***