Nasri Chiheb

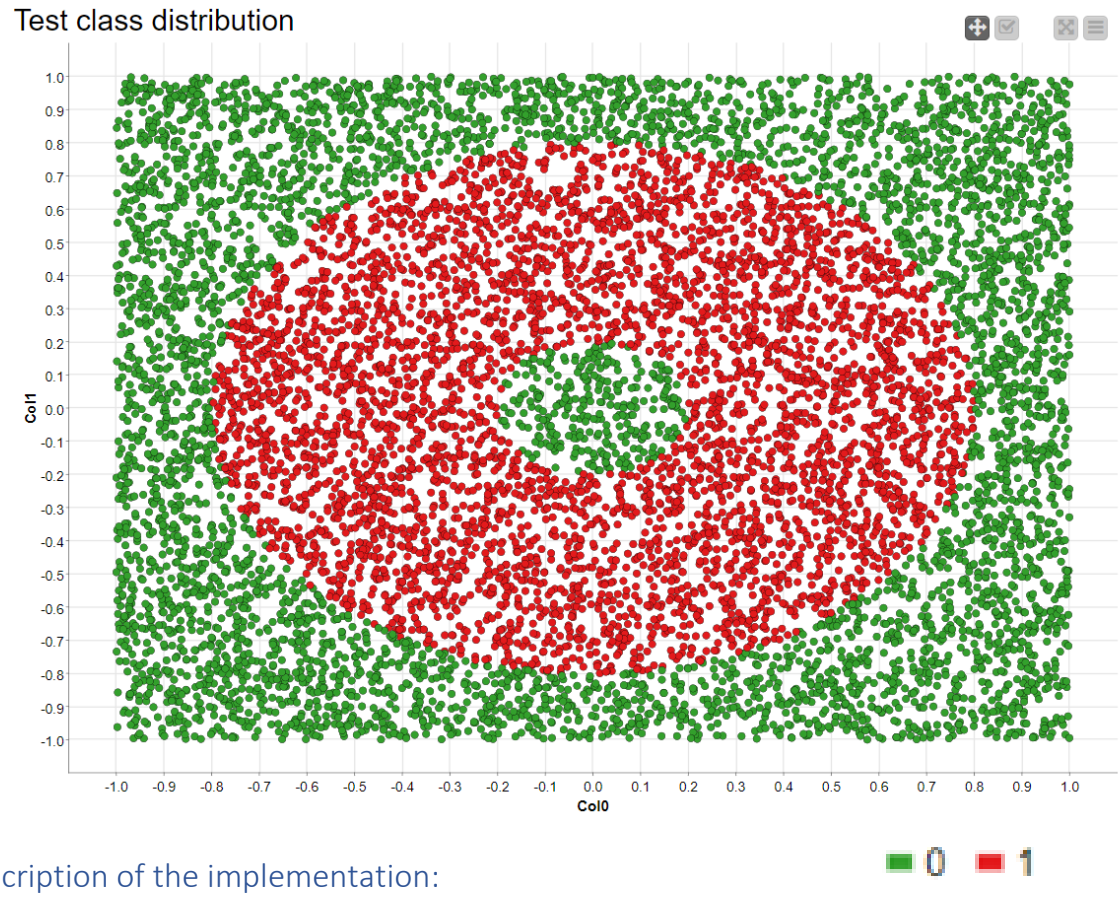
UNIVERSITAT ROVIRA I VIRGILI

Neural and Evolutionary Computation

A2: Comparison of SVM with BP and Linear Regression

Introduction:

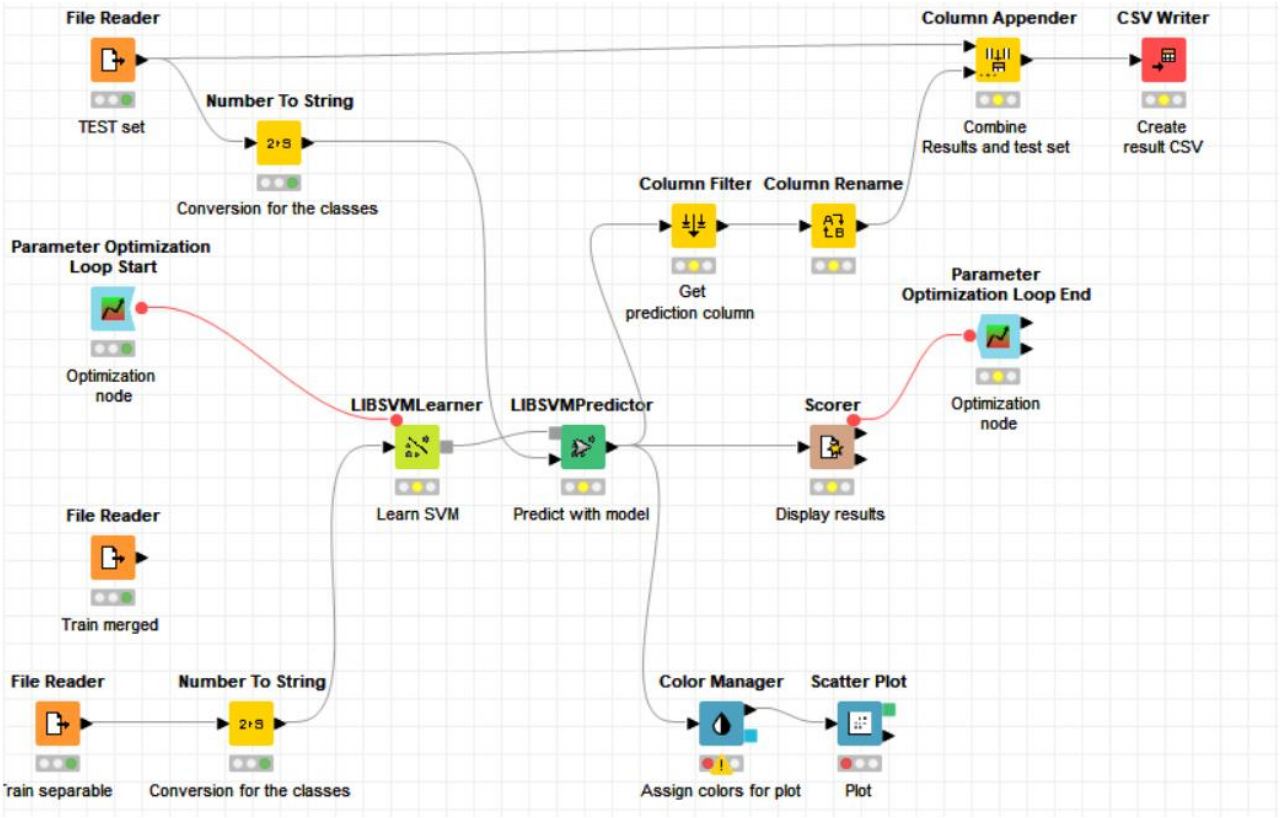
The goal of this assignment is to compare three classification methods SVM , BP and Linear Regression. These methods will be compared using the same datasets. Prediction will be binary classification of 10 000 row test set. All methods will be trained with two different dataset the merged and the separable ring. In this report we will discuss about the implementation by giving a short description of the used software and implementation decisions for the methods. Then, we will discuss about the results and finally the how to run part. Please find attached to this report the csv results and the code.

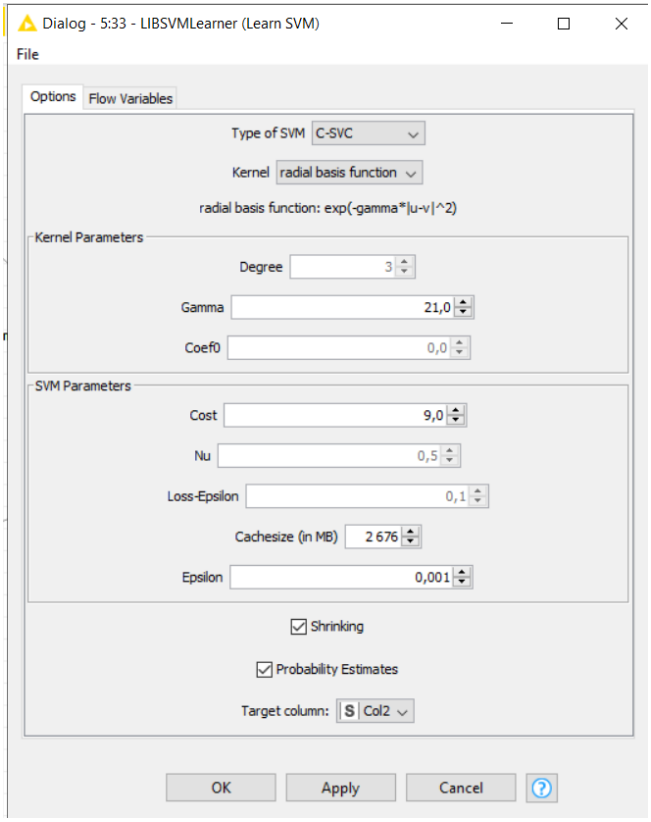


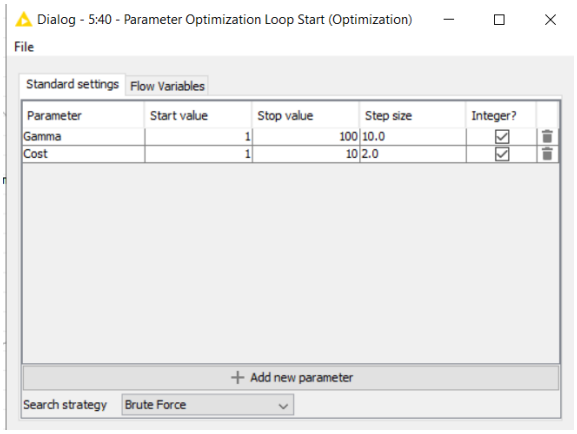
Description of the implementation:

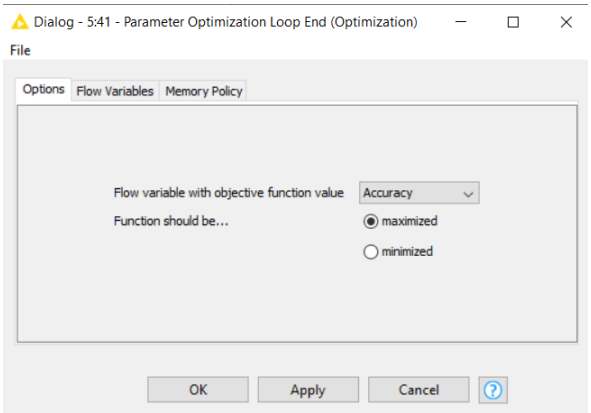
The implementation will be maid with the free software Knime. It is an open source data analytics platform. I chose this software because I am used to it and also it provides multiple choices for machine learning. This software is also ideal for this assignment as the implementations is simplified. For all the implementations the hole dataset was used.

SVM : SVM is implemented with LibSVM node. I chose to use radial basis function for the kernel with gamma = 21 and Cost = 9. Gamma and C values where determined by optimization loops. I chose RBF kernel because it is the most appropriate for this dataset.

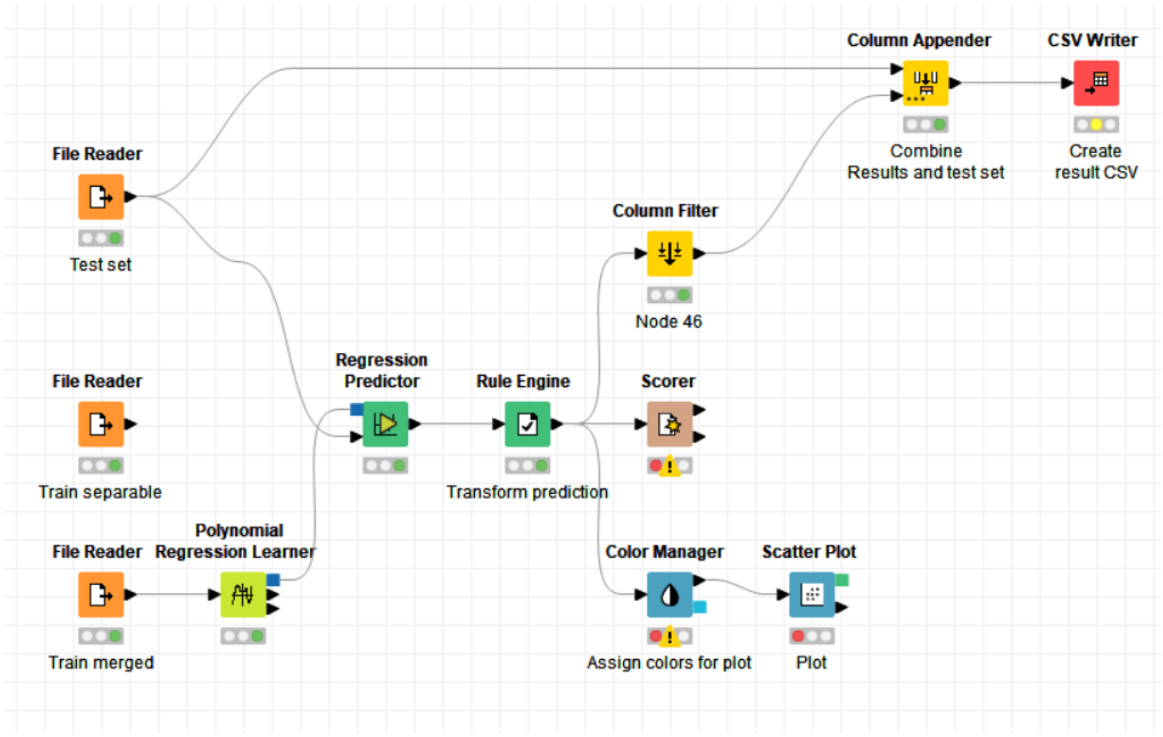




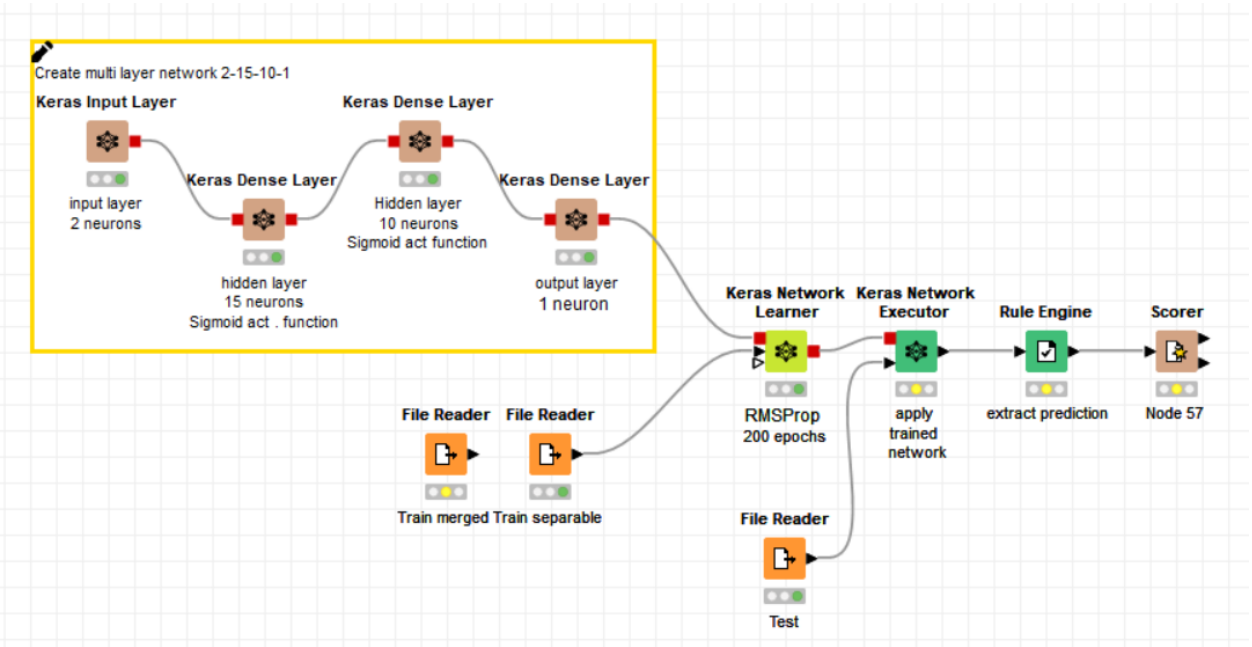


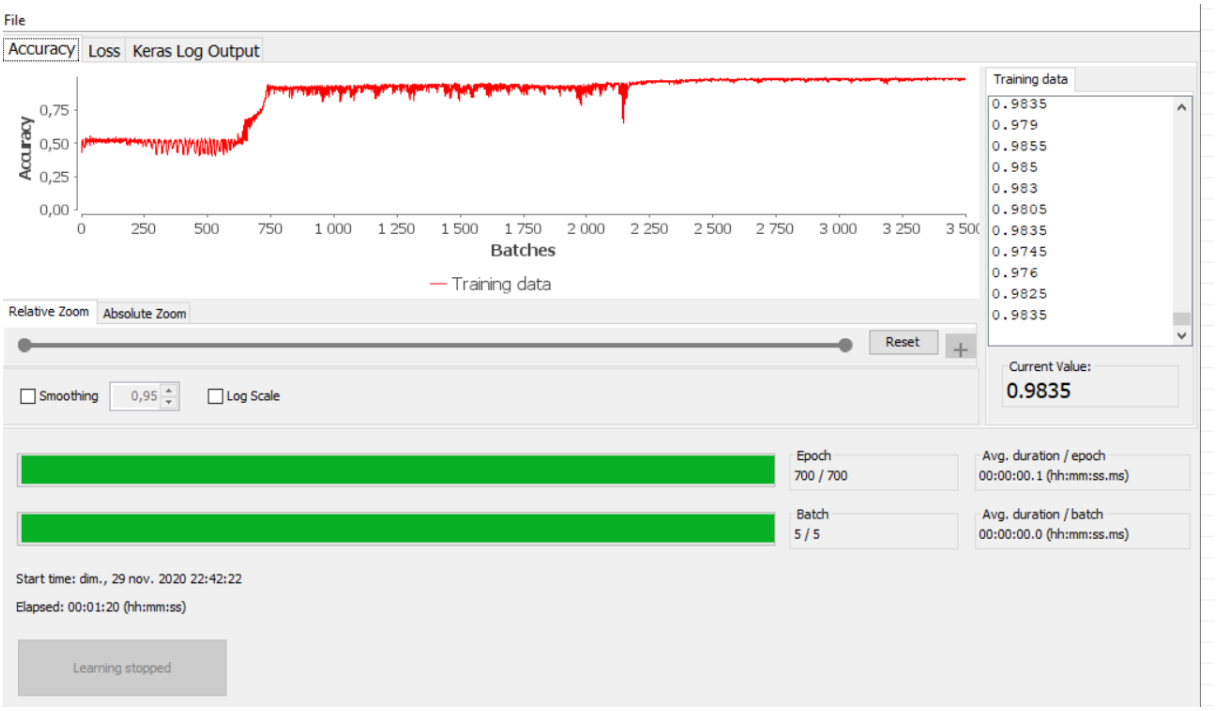


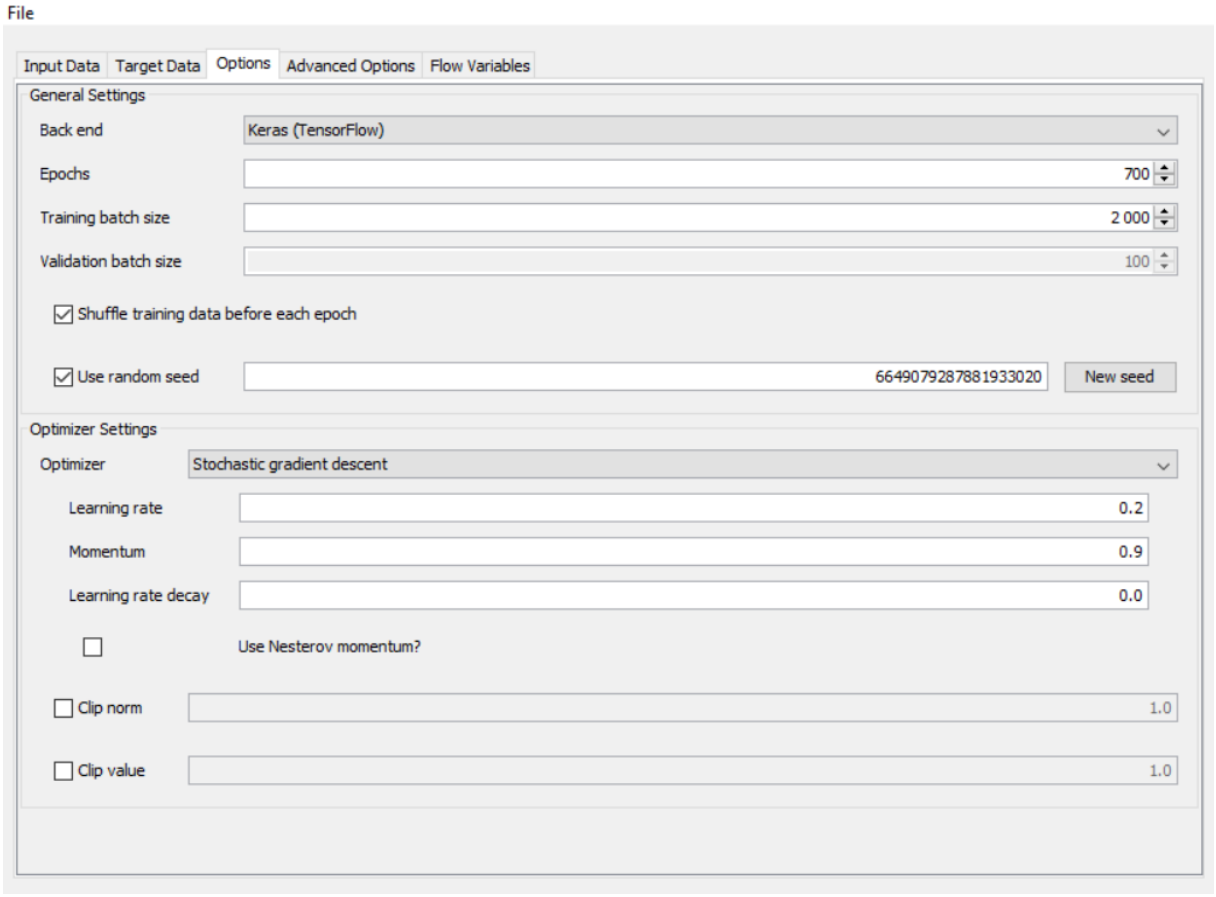
MLR : MLR model was implemented with the Polynomial Regression Learner. Predictions were maid by Regression Predictor node. Predicted results at the end were converted if it's above 0.5 it belongs to class 1 otherwise it is class 0.



BP : Backpropagation was implemented with TensorFlow/Keras node. First, we create a 2-15-10-1 multilayer network. We define activation function as Sigmoid. Train model and we use it to predict with the test set. Network was optimized with stochastic gradient descent, Learning rate of 0.2 , Momentum of 0.1 ,700 epochs and 2 000 as training batch size. These parameters were defined with Learning monitor. Parameters could be more optimized by adding loops but it's a lot time consuming.

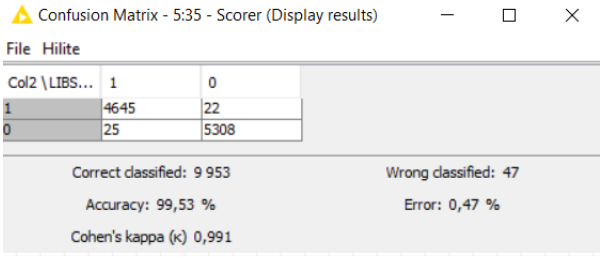


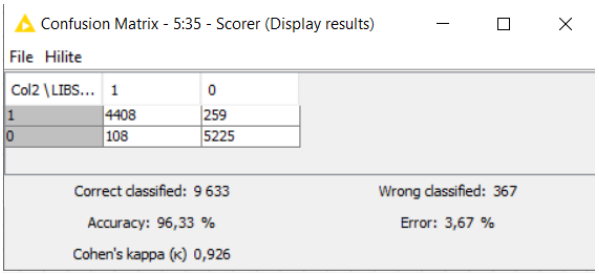

File
Accuracy
Loss
Keras Log Output
Accuracy
0,75
0,50
0,25
0,00
0
250
500
750
1 000
1 250
1 500
1 750
2 000
2 250
2 500
2 750
3 000
3 250
Batches
Training data
Training data
0.9835
0.979
0.9855
0.985
0.983
0.9805
0.9835
0.9745
0.976
0.9825
0.9835
Relative Zoom
Absolute Zoom
Reset
Smoothing
0,95
Log Scale
Current Value:
0.9835
Epoch
700 / 700
Avg. duration / epoch
00:00:00.1 (hh:mm:ss.ms)
Batch
5 / 5
Avg. duration / batch
00:00:00.0 (hh:mm:ss.ms)
Start time: dim., 29 nov. 2020 22:42:22
Elapsed: 00:01:20 (hh:mm:ss)
Learning stopped

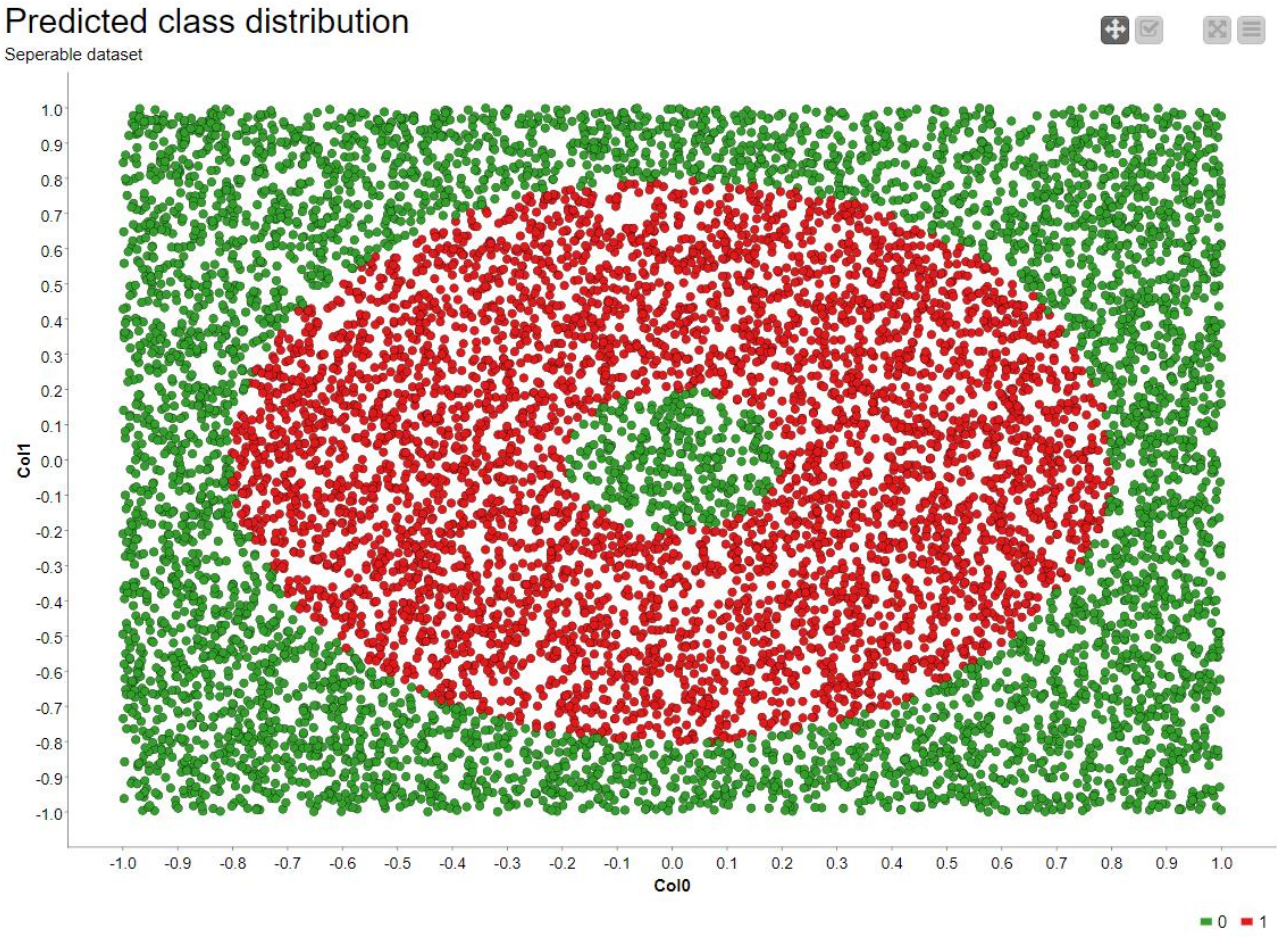

File
Input Data
Target Data
Options
Advanced Options
Flow Variables
General Settings
Back end
Keras (TensorFlow)
Epochs
700
Training batch size
2 000
Validation batch size
100
Shuffle training data before each epoch
Use random seed
6649079287881933020
New seed
Optimizer Settings
Optimizer
Stochastic gradient descent
Learning rate
0.2
Momentum
0.9
Learning rate decay
0.0
Use Nesterov momentum?
Clip norm
1.0
Clip value
1.0

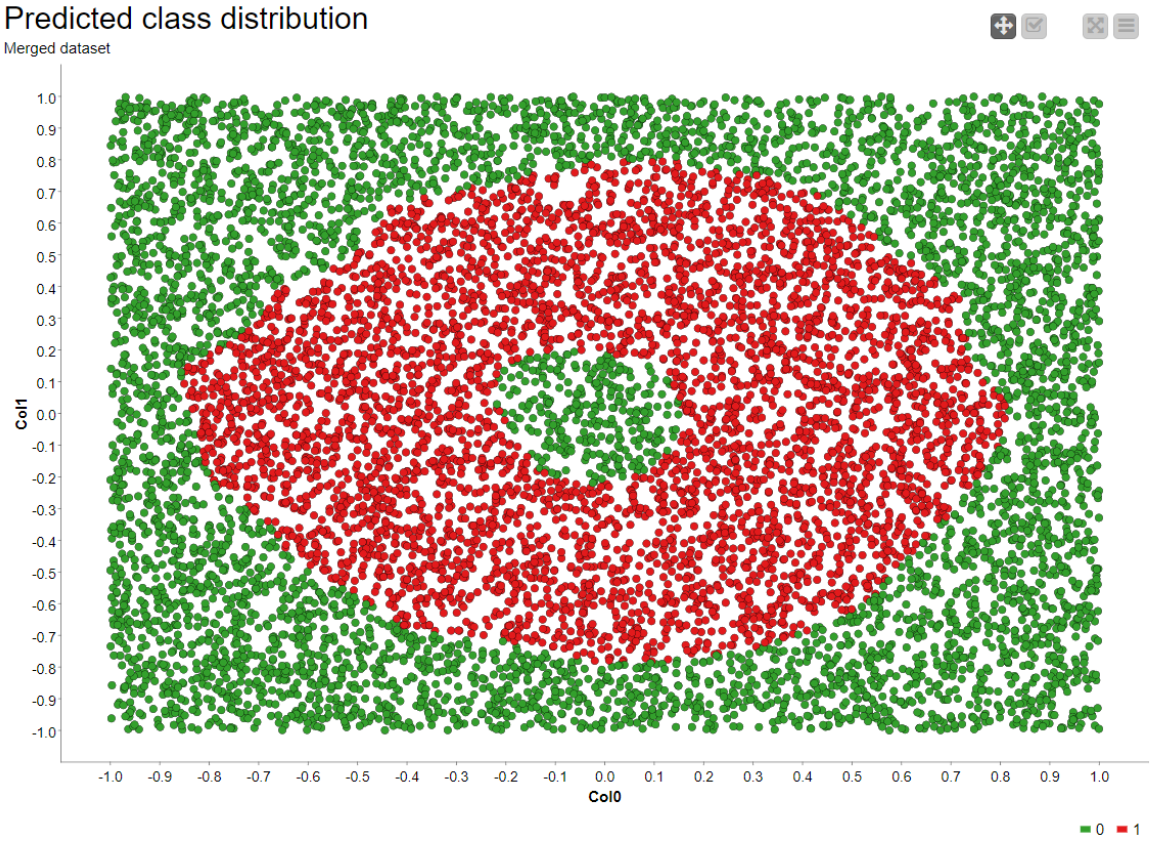
Results:

SVM : As shown in the confusion matrix error rate for SVM with <u>separable</u> dataset for train is 0,47% and for <u>merged</u> 3,67%.









MLR: As show in the confusion matrix For MLR error rate is of 3,72% with both datasets. As we can see the trained model is not able to detect classes is the middle.

Col2 \ Pred...	1	0
1	4637	30
0	342	4991

Correct classified: 9 628 Wrong classified: 372

Accuracy: 96,28 % Error: 3,72 %

Cohen's kappa (κ) 0,926

BP : As show in the confusion matrix error rate for separable train set is 1,79% and 7,24% for merged train set. As we can see BP is unable to detect middle zone with merged dataset as training set.

File Hilite

Col2 \ Pred...	1	0
1	4500	167
0	12	5321

Correct classified: 9 821 Wrong classified: 179

Accuracy: 98,21 % Error: 1,79 %

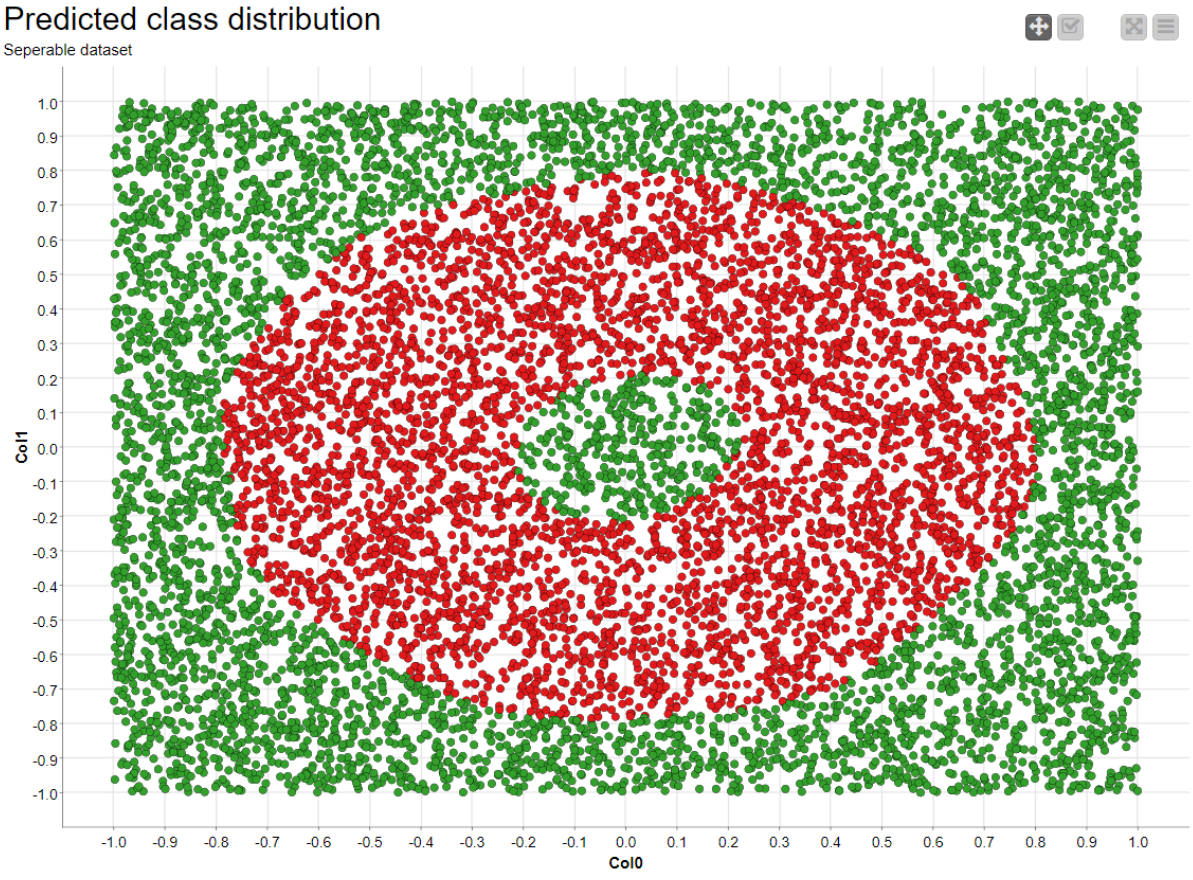
Cohen's kappa (κ) 0,964

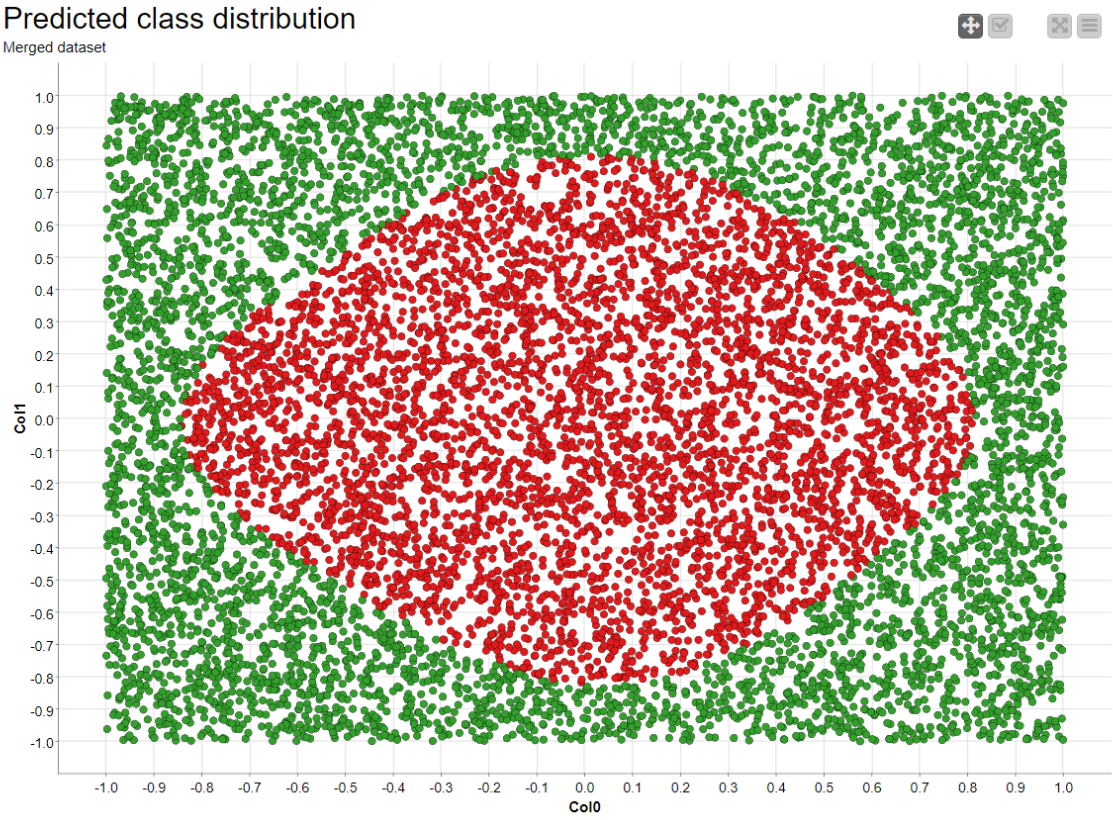
Col2 \ Pred...	1	0
1	4320	347
0	377	4956

Correct classified: 9 276 Wrong classified: 724

Accuracy: 92,76 % Error: 7,24 %

Cohen's kappa (κ) 0,855





To conclude, we can say that overall SVM is best choice for this dataset as we got the best error rate with both datasets, and computing time was significantly better than for BP. MLR has shown the worst result with merged data set but it is better with separated dataset and has the best execution time. BP has shown good result but is not efficient with merged dataset and is time consuming. All results above are maid with same parameters. The optimization was made with separated dataset. These results can be improved for BP and SVM with parameters looping but I did not do it by lack of time.

How to run:

Prerequisite :

- Knime
- Different workflows corresponding to each implementation method
- Packages used will be asked to download when you run the nodes

Run :

- Open Workflow (e.g BP.knwf)
- Run