

Neural and Evolutionary Computation (NEC)

A2: Classification with SVM, BP and MLR

Professors: SERGIO GÓMEZ JIMÉNEZ, JORDI DUCH GAVALDÀ

Students: SEBASTIAN BUZDUGAN, ERIKA LOZADA MARTINEZ

Main Concepts:

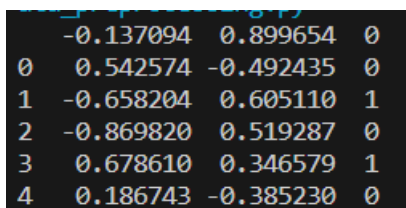
The task involves data classification using Support Vector Machines (SVM), Back-Propagation (BP), and Multiple Linear Regression (MLR) on different datasets. Key aspects include parameter selection via cross-validation, data preprocessing, normalization, and evaluation through error rates, confusion matrices, ROC curves, and AUC. The assignment emphasizes finding optimal parameters for each algorithm, handling both binary and multifeatured datasets, and includes comprehensive documentation of methods and analyses.

Part 1: Selecting and analyzing the datasets

Ring dataset

In this dataset the two first columns are the input and the last one is the classifier:

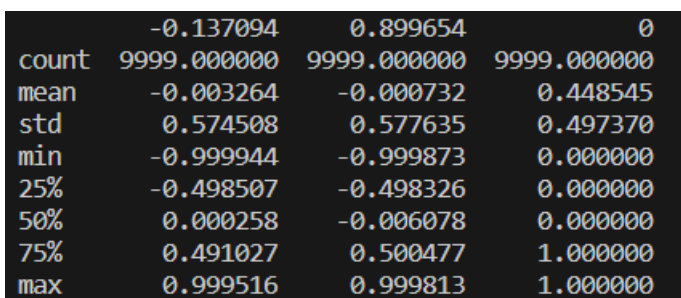
The head of the dataset is the following:



	-0.137094	0.899654	0
0	0.542574	-0.492435	0
1	-0.658204	0.605110	1
2	-0.869820	0.519287	0
3	0.678610	0.346579	1
4	0.186743	-0.385230	0

Figure 1: Ring dataset head

The descriptive statistics of the dataset are the following:



	-0.137094	0.899654	0
count	9999.000000	9999.000000	9999.000000
mean	-0.003264	-0.000732	0.448545
std	0.574508	0.577635	0.497370
min	-0.999944	-0.999873	0.000000
25%	-0.498507	-0.498326	0.000000
50%	0.000258	-0.006078	0.000000
75%	0.491027	0.500477	1.000000
max	0.999516	0.999813	1.000000

Figure 2: Ring dataset description

Data seems to be normalized. Here below there is the plot of the ring dataset:

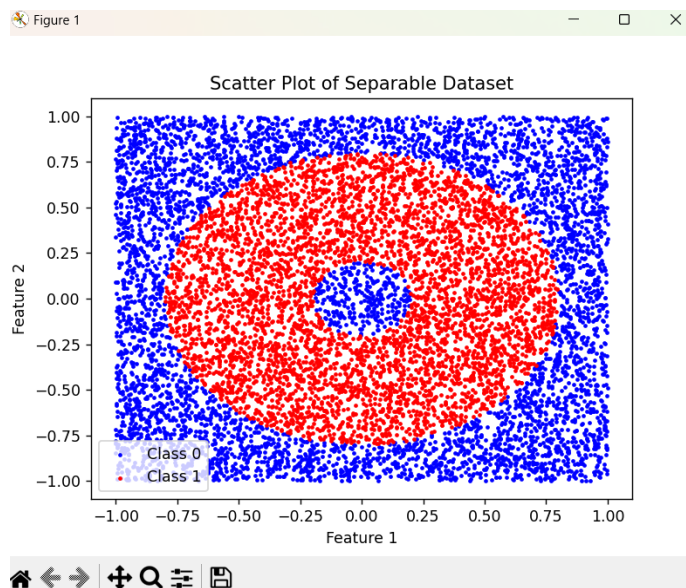


Figure 3: Separable dataset

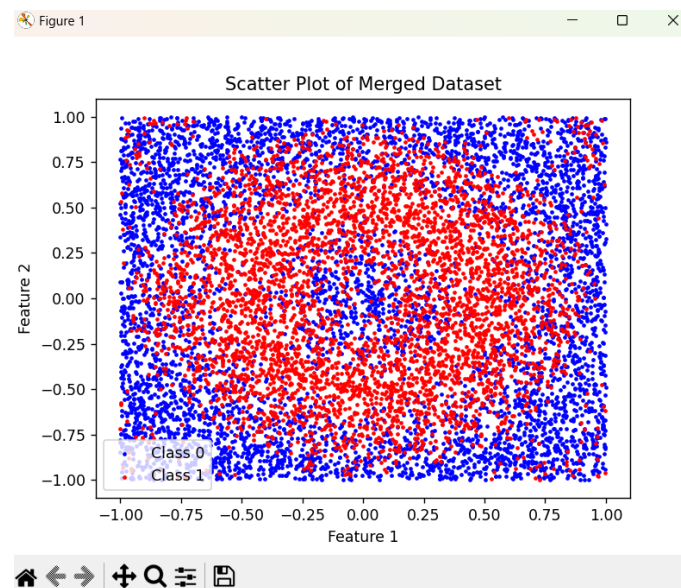


Figure 4: Merged dataset

Bank dataset

This dataset has 20 features, the first 19 columns are input and the last one is classifier for prediction. From this dataset the chosen file was the bank-additional.csv.

The head of the dataset and the descriptive statistics are the following:

	age	job	marital	education	default	housing	loan	...	poutcome	emp.var.rate	cons.price.idx	cons.conf.idx	euribor3m	nr.employed	y
0	30	blue-collar	married	basic.9y	no	yes	no	...	nonexistent	-1.8	92.893	-46.2	1.313	5099.1	no
1	39	services	single	high.school	no	no	no	...	nonexistent	1.1	93.994	-36.4	4.855	5191.0	no
2	25	services	married	high.school	no	yes	no	...	nonexistent	1.4	94.465	-41.8	4.962	5228.1	no
3	38	services	married	basic.9y	no	unknown	unknown	...	nonexistent	1.4	94.465	-41.8	4.959	5228.1	no
4	47	admin.	married	university.degree	no	yes	no	...	nonexistent	-0.1	93.200	-42.0	4.191	5195.8	no
[5 rows x 21 columns]															
	age	duration	campaign	pdays	previous	emp.var.rate	cons.price.idx	cons.conf.idx	euribor3m	nr.employed					
count	4119.000000	4119.000000	4119.000000	4119.000000	4119.000000	4119.000000	4119.000000	4119.000000	4119.000000	4119.000000	4119.000000	4119.000000	4119.000000	4119.000000	4119.000000
mean	40.113620	256.788055	2.537266	960.422190	0.190337	0.084972	93.579704	-40.499102	3.621356	5166.481695					
std	10.313362	254.703736	2.568159	191.922786	0.541788	1.563114	0.579349	4.594578	1.733591	73.667904					
min	18.000000	0.000000	1.000000	0.000000	0.000000	-3.400000	92.201000	-50.800000	0.635000	4963.600000					
25%	32.000000	103.000000	1.000000	999.000000	0.000000	-1.800000	93.075000	-42.700000	1.334000	5099.100000					
50%	38.000000	181.000000	2.000000	999.000000	0.000000	1.100000	93.749000	-41.800000	4.857000	5191.000000					
75%	47.000000	317.000000	3.000000	999.000000	0.000000	1.400000	93.994000	-36.400000	4.961000	5228.100000					
max	88.000000	3643.000000	35.000000	999.000000	6.000000	1.400000	94.767000	-26.900000	5.045000	5228.100000					

Figure 5: Bank dataset head and description

The variables of the dataset are:

- "age": numerical
- "job": categorical
- "marital": categorical
- "education": categorical
- "default": categorical
- "housing": categorical
- "loan": categorical
- "contact": categorical
- "month": categorical
- "day_of_week": categorical
- "duration": numerical
- "campaign": numerical
- "pdays": numerical
- "previous": numerical
- "poutcome": categorical
- "emp.var.rate": numerical

- "cons.price.idx":numerical
- "cons.conf.idx":numerical
- "euribor3m":numerical
- "nr.employed":numerical
- "y":categorical

In this dataset there were unknown values that were filled with the mode of the column. The yes/no column was converted to numerical values that represent the classes, yes values were converted to 1 and no to 0. The rest of categorical columns were converted to numbers.

Then, the processed dataset is the following:

	age	job	marital	education	default	housing	loan	contact	...	previous	outcome	emp.var.rate	cons.price.idx	cons.conf.idx	euribor3m	nr.employed	y
1	39	1	1	1	0	0	0	1	...	0	0	1.1	93.994	-36.4	4.855	5191.0	0
2	25	1	0	1	0	1	0	1	...	0	0	1.4	94.465	-41.8	4.962	5228.1	0
3	38	1	0	0	0	1	0	1	...	0	0	1.4	94.465	-41.8	4.959	5228.1	0
4	47	2	0	2	0	1	0	0	...	0	0	-0.1	93.200	-42.0	4.191	5195.8	0
5	32	1	1	2	0	0	0	0	...	2	1	-1.1	94.199	-37.5	0.884	4963.6	0

Figure 6: Bank dataset removed unknown

Since, the last column is using as classifier, the rest of categorical variables of the dataset are not considered.

	age	duration	campaign	pdays	previous	emp.var.rate	cons.price.idx	cons.conf.idx	euribor3m	nr.employed	y
1	39	346	4	999	0	1.1	93.994	-36.4	4.855	5191.0	0
2	25	227	1	999	0	1.4	94.465	-41.8	4.962	5228.1	0
3	38	17	3	999	0	1.4	94.465	-41.8	4.959	5228.1	0
4	47	58	1	999	0	-0.1	93.200	-42.0	4.191	5195.8	0
5	32	128	3	999	2	-1.1	94.199	-37.5	0.884	4963.6	0

Figure 7: Bank dataset head

Finally data was normalized as follows:

- age: There are positive values that vary widely, so it will be normalized with Min-Max scaling between 0 and 1.
- duration: There are positive values that vary widely, so it will be normalized with Min-Max scaling between 0 and 1.
- campaign: There are positive values that vary widely, so it will be normalized with Min-Max scaling between 0 and 1.
- pdays: There are positive values that vary widely, so it will be normalized with Min-Max scaling between 0 and 1.
- previous: There are positive values that vary widely, so it will be normalized with Min-Max scaling between 0 and 1.
- emp.var.rate: There are positive values that vary widely, so it will be normalized with Min-Max scaling between 0 and 1.
- cons.price.idx: There are positive values that vary widely, so it will be normalized with Min-Max scaling between 0 and 1.
- cons.conf.idx: There are positive values that vary widely, so it will be normalized with Min-Max scaling between 0 and 1.
- euribor3m: There are positive values that vary widely, so it will be normalized with Min-Max scaling between 0 and 1.

- **nr.employed:** There are positive values that vary widely, so it will be normalized with Min-Max scaling between 0 and 1.

The head of the resulting dataset is:

	age	duration	campaign	pdays	previous	emp.var.rate	cons.price.idx	cons.conf.idx	euribor3m	nr.employed	y
1	0.300000	0.094977	0.088235	1.0	0.000000	0.937500	0.698753	0.602510	0.956916	0.859735	0
2	0.100000	0.062311	0.000000	1.0	0.000000	1.000000	0.882307	0.376569	0.981179	1.000000	0
3	0.285714	0.004666	0.058824	1.0	0.000000	1.000000	0.882307	0.376569	0.980499	1.000000	0
4	0.414286	0.015921	0.000000	1.0	0.000000	0.687500	0.389322	0.368201	0.806349	0.877883	0
5	0.200000	0.035136	0.058824	1.0	0.333333	0.479167	0.778644	0.556485	0.056463	0.000000	0

Figure 8: Bank dataset normalized head

Here below there is the plot of the Banknote dataset:

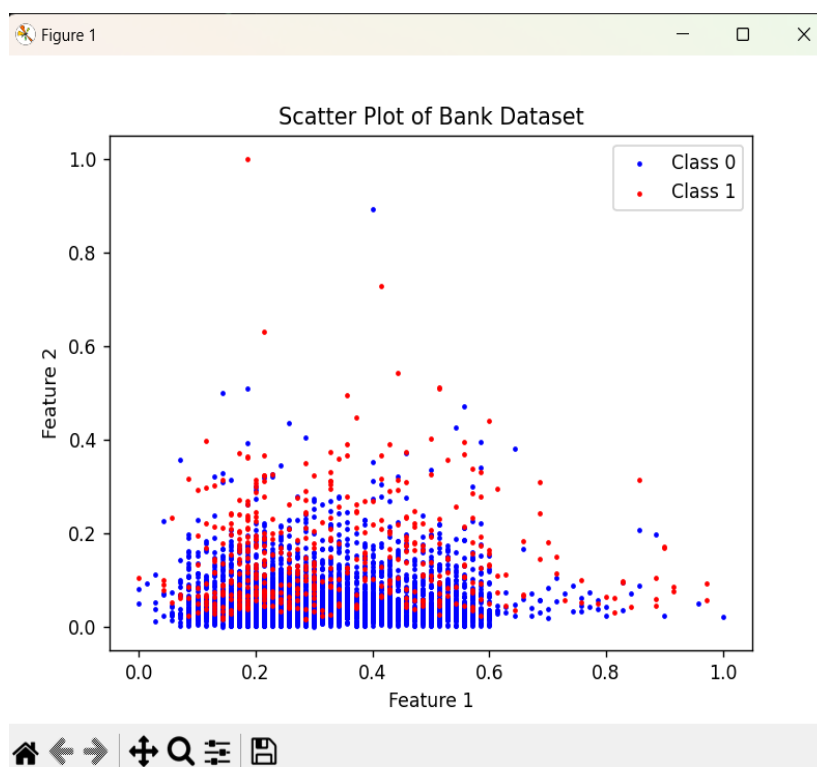


Figure 9: Bank dataset normalized

Personalized dataset: Banknote authentication

Dataset variables

The personalized dataset was obtained from the University of California, Irvine Machine Learning Repository [1]. This dataset contains data extracted from images that were taken for the evaluation of an authentication procedure for banknotes. The dataset was downloaded from <https://archive.ics.uci.edu/dataset/267/banknote+authentication>.

1. variance of Wavelet Transformed image (continuous)
2. skewness of Wavelet Transformed image (continuous)

3. curtosis of Wavelet Transformed image (continuous)
4. entropy of image (continuous)
5. class (integer)

Here below there is the head of the dataset:

	3.6216	8.6661	-2.8073	-0.44699	0
0	4.54590	8.1674	-2.4586	-1.46210	0
1	3.86600	-2.6383	1.9242	0.10645	0
2	3.45660	9.5228	-4.0112	-3.59440	0
3	0.32924	-4.4552	4.5718	-0.98880	0
4	4.36840	9.6718	-3.9606	-3.16250	0

Figure 10: Banknote head

In this dataset the first four columns are the input and the last one is the classification one. The following image contains the descriptive statistics of the dataset:

	3.6216	8.6661	-2.8073	-0.44699	0
count	1371.000000	1371.000000	1371.000000	1371.000000	1371.000000
mean	0.431410	1.917434	1.400694	-1.192200	0.444931
std	2.842494	5.868359	4.310105	2.101683	0.497139
min	-7.042100	-13.773100	-5.286100	-8.548200	0.000000
25%	-1.774700	-1.711300	-1.553350	-2.417000	0.000000
50%	0.495710	2.313400	0.616630	-0.586650	0.000000
75%	2.814650	6.813100	3.181600	0.394810	1.000000
max	6.824800	12.951600	17.927400	2.449500	1.000000

Figure 11: Banknote descriptive statistics

Then data normalization was applied in numerical variables as follows:

- Variance: There are positive values that vary widely, so it will be normalized with Min-Max scaling between 0 and 1.
- Skewness: There are positive values that vary widely, so it will be normalized with Min-Max scaling between 0 and 1.
- Curtosis: It follows normal distribution so Z-score normalization will be applied.
- Entropy: It follows normal distribution so Z-score normalization will be applied.
- Class: No normalization is needed for this variable since it is categorical.

Data was shuffled to avoid sorting,

The head of the dataset after normalization is the following:

	3.6216	8.6661	-2.8073	-0.44699	0
430	0.602211	0.699574	0.139348	0.633563	0
588	0.575342	0.838715	-0.704343	0.232126	0
296	0.599463	0.816028	-0.370572	0.390938	0
184	0.417426	0.922495	0.123038	0.625802	0
244	0.811681	0.410373	0.141738	0.634700	0

Figure 12: Banknote normalized head

Here below there is the plot of the Banknote dataset:

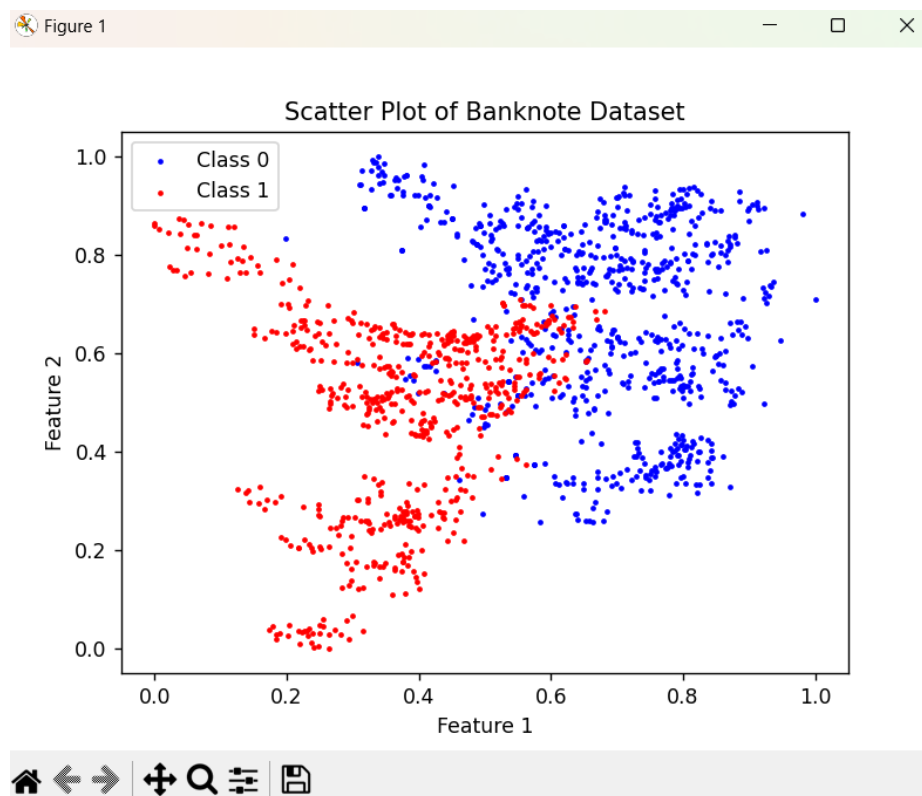


Figure 13: Banknote normalized

Part 2: Classification problem

1. Support Vector Machines (SVM)

Support Vector Machines (SVM) are a supervised learning algorithm primarily used for classification tasks. They work by finding a hyperplane in a multi-dimensional space to distinctly classify data points. SVM is particularly effective due to its ability to use kernel functions, making it capable of handling both linear and non-linear data [2].

The SVM model has the following parameters:

- C: Is the regularization.
- kernel: Specifies the kernel type that can be 'linear', 'poly', 'rbf', 'sigmoid' and 'precomputed'.
- gamma: Kernel coefficient influences the shape of the decision boundary.
- degree: Degree of the polynomial kernel function.
- coef0: Independent term in the kernel function used in 'poly' and 'sigmoid'.
- shrinking: Whether to use the shrinking heuristic to speed up the training process.
- probability: Whether to enable probability estimates.
- tol: Tolerance for stopping criterion.
- class_weight: Weights associated with classes.
- max_iter: Hard limit on iterations within solver.

Ring dataset

The supervised training of the dataset was performed for the following parameters:

- Kernel:'linear', 'poly', 'rbf', 'sigmoid', 'precomputed'
- Constant:100,200,300,400,500,600,700,800,900

From the execution it was possible to identify that the parameters in which the accuracy is the maximum and the classification error is the minimum is:

- Kernel: rbf
- Constant:701

With the best parameters the prediction was performed with the 80% of the dataset for training and the other 20% for test:

- Separable dataset

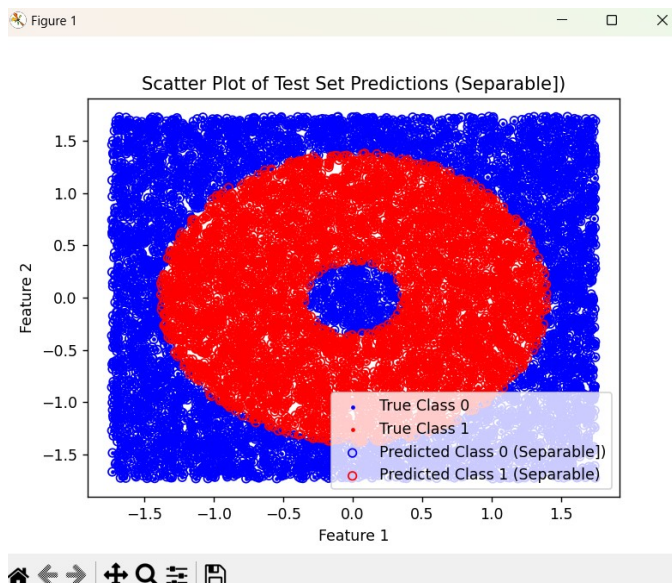


Figure 14: Ring separable SVM prediction

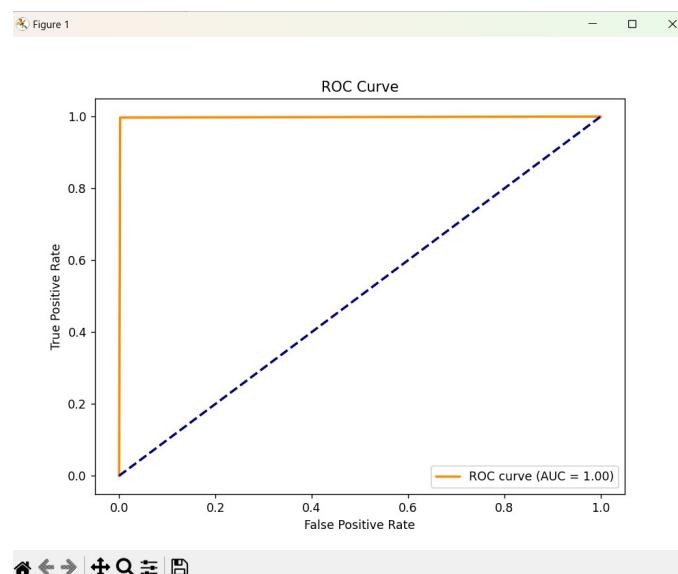


Figure 15: Ring separable SVM ROC

```
Cross-Validation Scores: [0.9995 0.9975 0.9965 0.9955 0.996 ]
Mean Cross-Validation Score: 0.9969999999999999
```

```
PARAM> kernel=rbf C=701
Separable dataset
Accuracy on the test set for dataset: 1.00
Classification error on the test set for dataset: 0.25%

Confusion Matrix for Separable Dataset:
[[5321  12]
 [ 13 4654]]

Classification Report for Separable Dataset:
              precision    recall  f1-score   support

     0         1.00      1.00      1.00     5333
     1         1.00      1.00      1.00     4667

   accuracy              1.00      10000
  macro avg              1.00      10000
 weighted avg              1.00      10000
```

Figure 16: Ring separable SVM result

The confusion matrix is detailed as follows:

- True Positive (TP): Instances predicted correctly positive by the model.
- False Negative (FN): Instances predicted incorrectly negative by the model.
- False Positive (FP): Instances predicted incorrectly positive by the model.
- True Negative (TN): Instances predicted correctly negative by the model.

From this definitions, the confusion matrix states that the model predicts more correct values than the incorrect ones, because the values in the upper left corner and in the bottom right corner are higher.

Additionally, the ROC curve shows that there curve is at top-left corner so the model is performing well.

- Merged dataset:

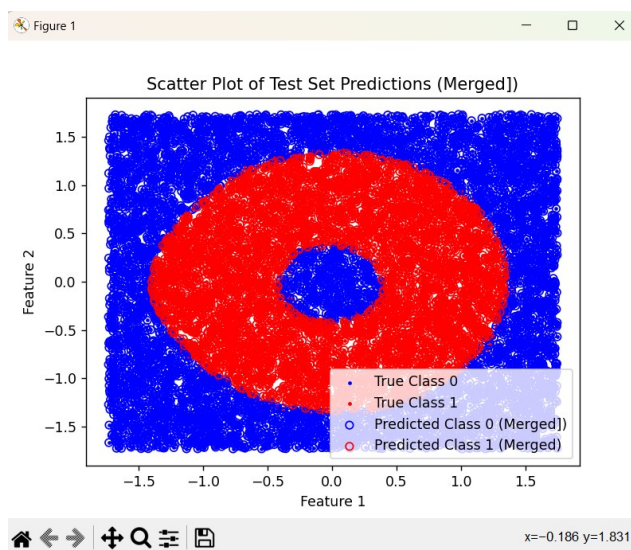


Figure 18: Merged separable SVM prediction

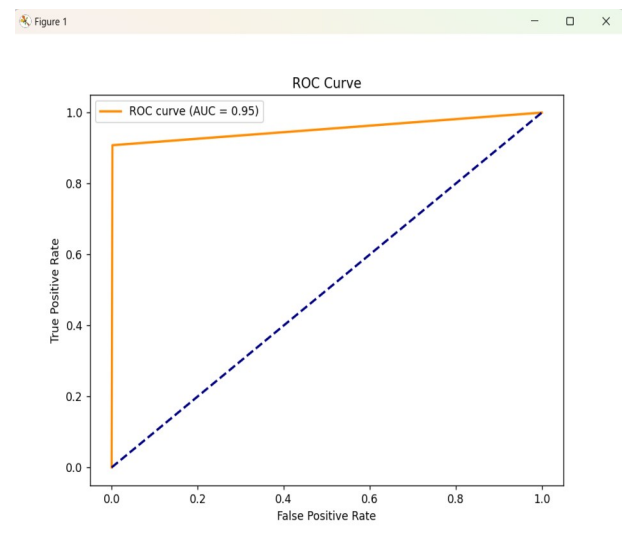


Figure 17: Merged separable SVM ROC

```
Cross-Validation Scores: [0.7785 0.782 0.778 0.7845 0.7715]
Mean Cross-Validation Score: 0.7788999999999999
```

```
Accuracy on the test set for dataset: 0.96
Classification error on the test set for dataset: 4.39%
```

```
Confusion Matrix for Merged Dataset:
[[5323  10]
 [ 429 4238]]
```

```
Classification Report for Merged Dataset:
              precision    recall  f1-score   support

     0       0.93        1.00        0.96        5333
     1       1.00        0.91        0.95        4667

   accuracy          0.96          0.96          0.96       10000
  macro avg          0.96          0.95          0.96       10000
 weighted avg          0.96          0.96          0.96       10000
```

Figure 19: Merged separable SVM result

The confusion matrix states that the model predicts more correct values than the incorrect ones, because the values in the upper left corner and in the bottom right corner are higher.

Additionally, the ROC curve shows that there curve is at top-left corner so the model is performing well.

Bank dataset

The supervised training of the dataset was performed for the following parameters:

- Kernel:'linear', 'poly', 'rbf', 'sigmoid', 'precomputed'
- Constant:1,2,3,4,5,6,7,8,9,10
- Degree:1,2,3,4,5

```
PARAM> kernel=linear C=7
Accuracy on the test set for dataset: 0.91
Classification error on the test set for dataset: 9.22%
PARAM> kernel=linear C=7
Accuracy on the test set for dataset: 0.91
Classification error on the test set for dataset: 9.22%
PARAM> kernel=linear C=7
Accuracy on the test set for dataset: 0.91
Classification error on the test set for dataset: 9.22%
PARAM> kernel=linear C=7
Accuracy on the test set for dataset: 0.91
Classification error on the test set for dataset: 9.22%
PARAM> kernel=linear C=7
Accuracy on the test set for dataset: 0.91
Classification error on the test set for dataset: 9.22%
PARAM> kernel=linear C=7
Accuracy on the test set for dataset: 0.91
Classification error on the test set for dataset: 9.22%
```

Figure 20: Bank execution automation

```
Classification error on the test set for dataset: 8.86%
PARAM> kernel=poly C=4
Accuracy on the test set for dataset: 0.92
Classification error on the test set for dataset: 8.01%
PARAM> kernel=poly C=4
Accuracy on the test set for dataset: 0.92
Classification error on the test set for dataset: 8.50%
PARAM> kernel=poly C=4
Accuracy on the test set for dataset: 0.91
Classification error on the test set for dataset: 9.22%
PARAM> kernel=poly C=4
Accuracy on the test set for dataset: 0.91
Classification error on the test set for dataset: 9.34%
PARAM> kernel=poly C=4
Accuracy on the test set for dataset: 0.91
Classification error on the test set for dataset: 9.34%
PARAM> kernel=poly C=4
Accuracy on the test set for dataset: 0.91
Classification error on the test set for dataset: 9.47%
```

```
PARAM> kernel=rbf C=1
Accuracy on the test set for dataset: 0.92
Classification error on the test set for dataset: 8.37%
PARAM> kernel=rbf C=1
Accuracy on the test set for dataset: 0.92
Classification error on the test set for dataset: 8.37%
PARAM> kernel=rbf C=2
Accuracy on the test set for dataset: 0.92
Classification error on the test set for dataset: 8.50%
PARAM> kernel=rbf C=2
Accuracy on the test set for dataset: 0.92
Classification error on the test set for dataset: 8.50%
PARAM> kernel=rbf C=2
Accuracy on the test set for dataset: 0.92
Classification error on the test set for dataset: 8.50%
PARAM> kernel=rbf C=2
Accuracy on the test set for dataset: 0.92
Classification error on the test set for dataset: 8.50%
```

From the execution it was possible to identify that the parameters in which the accuracy is the maximum and the classification error is the minimum is:

- Kernel: poly
- Constant:1

- Degree:4

With the best parameters the prediction was performed with the 80% of the dataset for training and the other 20% for test:

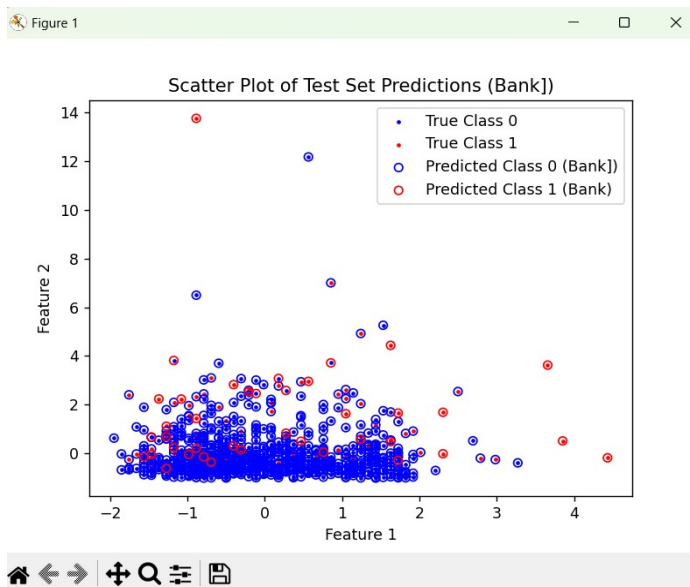


Figure 22: Bank prediction

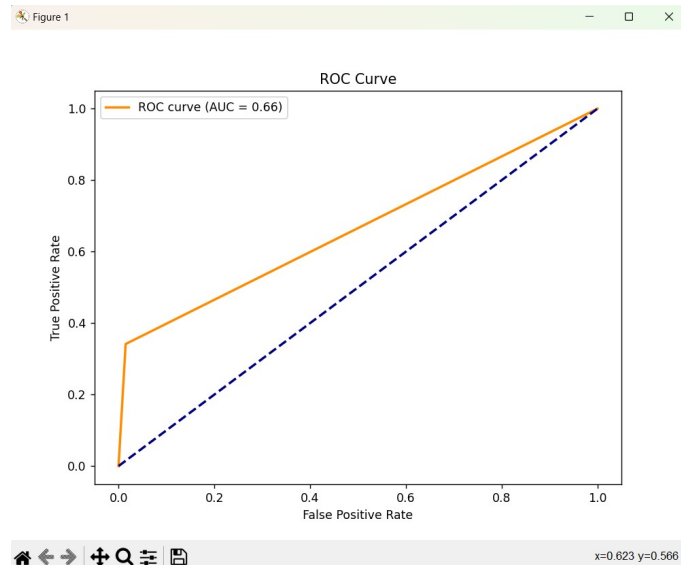


Figure 21: Bank ROC

The accuracy, the classification error, the confusion matrix and the classification report from the execution of the supervised training with the best parameters is the following:

```
Cross-Validation Scores: [0.90895296 0.90440061 0.91793313 0.89817629 0.90881459]
Mean Cross-Validation Score: 0.9076555156334318
```

```
Accuracy on the test set for dataset: 0.92
Classification error on the test set for dataset: 7.89%

Confusion Matrix for Bank Dataset:
[[731  11]
 [ 54  28]]

Classification Report for Bank Dataset:
              precision    recall  f1-score   support

     0       0.93         0.99         0.96         742
     1       0.72         0.34         0.46          82

   accuracy          0.92
  macro avg       0.82         0.66         0.71         824
 weighted avg       0.91         0.92         0.91         824
```

Figure 23: Bank result

Banknote dataset

The supervised training of the dataset was performed for the following parameters:

- Kernel:'linear', 'poly', 'rbf', 'sigmoid', 'precomputed'
- Constant:1,2,3,4,5,6,7,8,9,10
- Degree:1,2,3

```
PARAM> kernel=linear C=1 D1
Accuracy on the test set for dataset: 0.98
Classification error on the test set for dataset: 2.19%
PARAM> kernel=linear C=2 D1
Accuracy on the test set for dataset: 0.98
Classification error on the test set for dataset: 1.82%
PARAM> kernel=linear C=3 D1
Accuracy on the test set for dataset: 0.98
Classification error on the test set for dataset: 1.82%
PARAM> kernel=linear C=4 D1
Accuracy on the test set for dataset: 0.99
Classification error on the test set for dataset: 0.73%
PARAM> kernel=linear C=5 D1
Accuracy on the test set for dataset: 0.99
Classification error on the test set for dataset: 0.73%
PARAM> kernel=poly C=1 D1
Accuracy on the test set for dataset: 0.98
Classification error on the test set for dataset: 2.19%
PARAM> kernel=poly C=2 D1
Accuracy on the test set for dataset: 0.98
Classification error on the test set for dataset: 2.19%
```

```
PARAM> kernel=poly C=2 D1
Accuracy on the test set for dataset: 0.98
Classification error on the test set for dataset: 2.19%
PARAM> kernel=poly C=3 D1
Accuracy on the test set for dataset: 0.98
Classification error on the test set for dataset: 2.19%
PARAM> kernel=poly C=4 D1
Accuracy on the test set for dataset: 0.98
Classification error on the test set for dataset: 2.19%
PARAM> kernel=poly C=5 D1
Accuracy on the test set for dataset: 0.98
Classification error on the test set for dataset: 2.19%
PARAM> kernel=rbf C=1 D1
Accuracy on the test set for dataset: 0.99
Classification error on the test set for dataset: 1.09%
PARAM> kernel=rbf C=2 D1
Accuracy on the test set for dataset: 1.00
Classification error on the test set for dataset: 0.00%
PARAM> kernel=rbf C=3 D1
Accuracy on the test set for dataset: 1.00
Classification error on the test set for dataset: 0.00%
```

```
PARAM> kernel=rbf C=4 D1
Accuracy on the test set for dataset: 1.00
Classification error on the test set for dataset: 0.00%
PARAM> kernel=rbf C=5 D1
Accuracy on the test set for dataset: 1.00
Classification error on the test set for dataset: 0.00%
PARAM> kernel=sigmoid C=1 D1
Accuracy on the test set for dataset: 0.76
Classification error on the test set for dataset: 24.45%
PARAM> kernel=sigmoid C=2 D1
Accuracy on the test set for dataset: 0.76
Classification error on the test set for dataset: 24.09%
PARAM> kernel=sigmoid C=3 D1
Accuracy on the test set for dataset: 0.76
Classification error on the test set for dataset: 24.09%
PARAM> kernel=sigmoid C=4 D1
Accuracy on the test set for dataset: 0.76
Classification error on the test set for dataset: 24.09%
PARAM> kernel=sigmoid C=5 D1
Accuracy on the test set for dataset: 0.76
Classification error on the test set for dataset: 24.09%
```

From the execution it was possible to identify that the parameters in which the accuracy is the maximum and the classification error is the minimum is:

- Kernel: rbf
- Constant:2

With the best parameters the prediction was performed with the 80% of the dataset for training and the other 20% for test:

Figure 24: Banknote execution automation

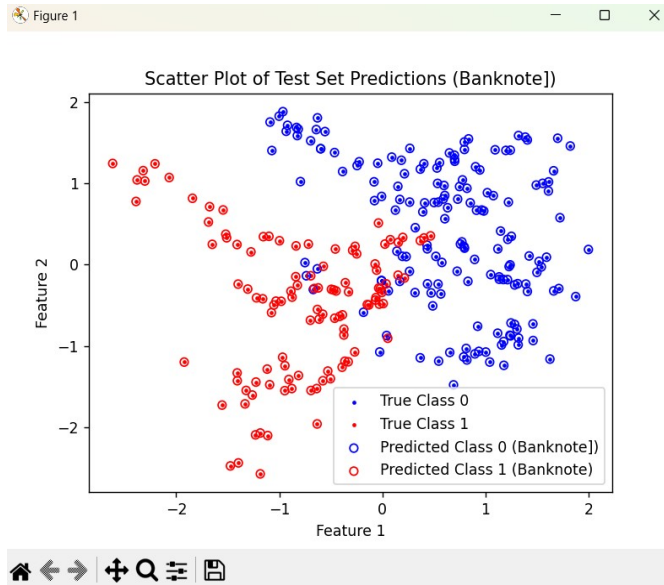


Figure 25: Banknote dataset prediction

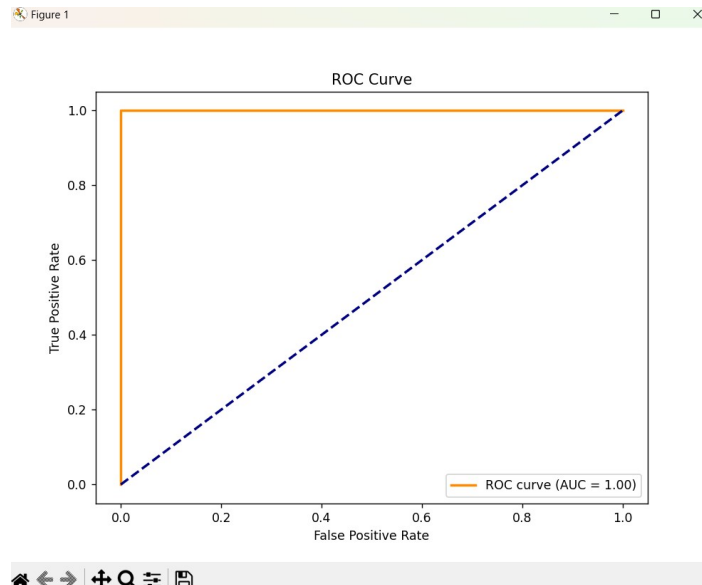


Figure 26: Banknote ROC

The accuracy, the classification error, the confusion matrix and the classification report from the execution of the supervised training with the best parameters is the following:

```
Cross-Validation Scores: [0.99543379 1. 0.99086758 1. 1. ]
Mean Cross-Validation Score: 0.9972602739726029
Accuracy on the test set for dataset: 1.00
Classification error on the test set for dataset: 0.00%

Confusion Matrix for Banknote Dataset:
[[164  0]
 [  0 110]]

Classification Report for Banknote Dataset:

```

	precision	recall	f1-score	support
0	1.00	1.00	1.00	164
1	1.00	1.00	1.00	110
accuracy			1.00	274
macro avg	1.00	1.00	1.00	274
weighted avg	1.00	1.00	1.00	274

Figure 27: Banknote dataset results

The confusion matrix states that the model predicts more correct values than the incorrect ones, because the values in the upper left corner and in the bottom right corner are higher.

Additionally, the ROC curve shows that there curve is at top-left corner so the model is performing well.

It was also used **Weka**, a popular data mining software, and the **LibSVM** package for SVM implementation, to better understand data [3]. LibSVM in Weka requires categorical class labels, so the dataset was labeled to categorical ('A' and 'B'). After this adjustment, the classification task was conducted in Weka's "Classify" tab, starting with the training set and then incorporating a test set for a comprehensive analysis.

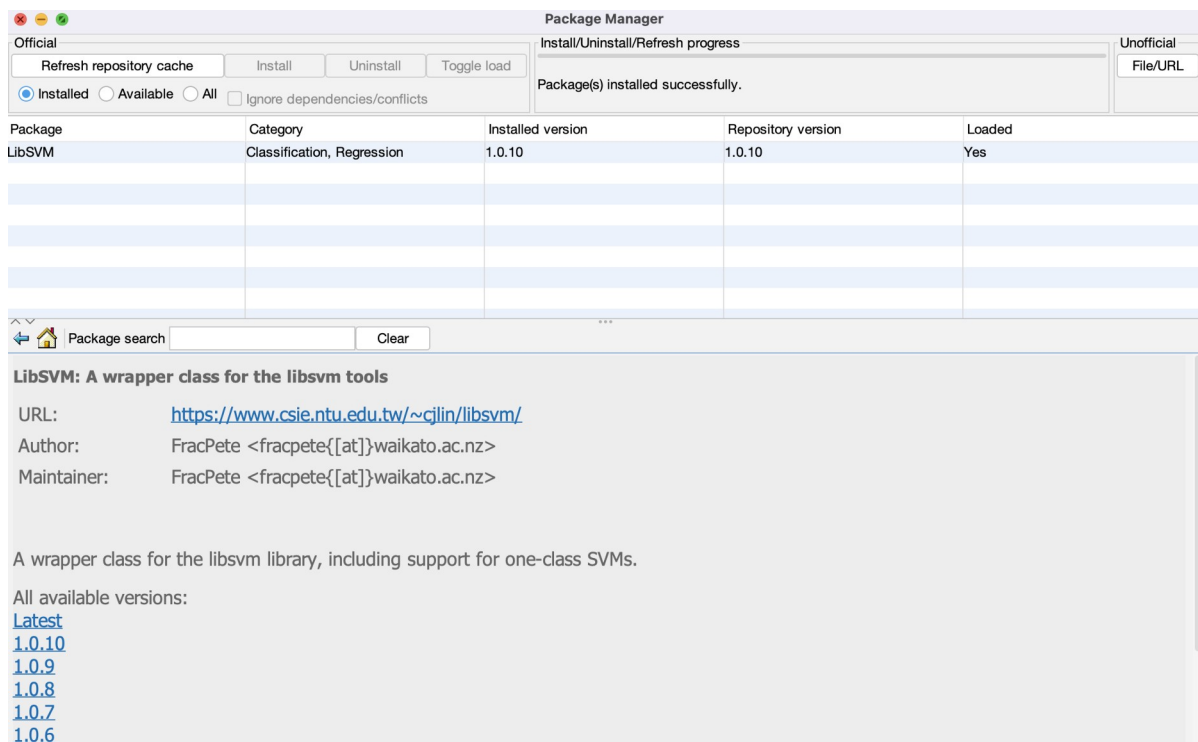


Figure 28: Weka SVM Ring

The first database used is “ring-merged-weka-letter” – and convert 0/1 into A/B because this package does not allow those values. With this database it was used the training set LibSVM and get the following output:

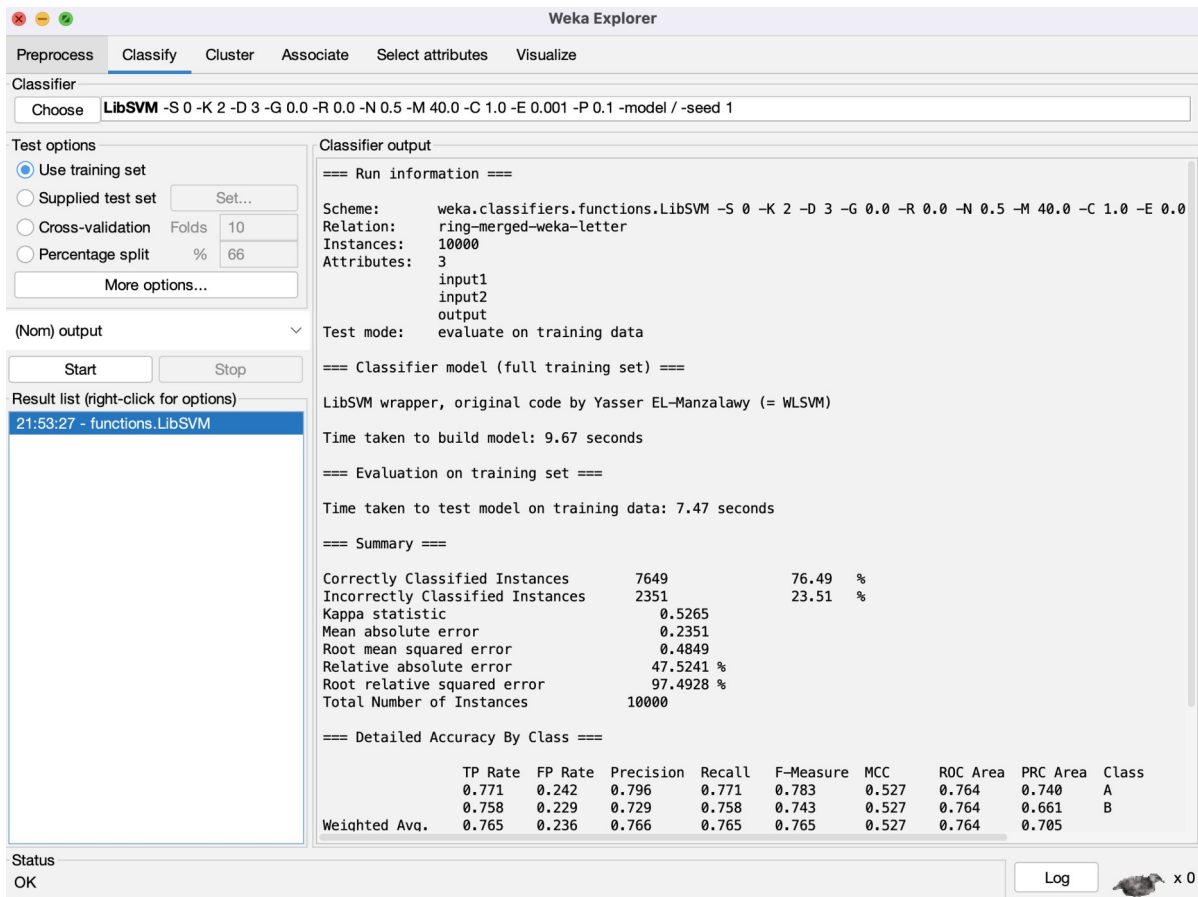


Figure 29: Weka SVM Ring

Then we need to use the custom test set in order to get the confusion Matrix. To compute the classification error using the formula :

$$E = 100 \cdot \frac{n_{01} + n_{10}}{n_{00} + n_{01} + n_{10} + n_{11}}$$

$$E = 100 \cdot \frac{300 + 118}{5033 + 300 + 118 + 4549}$$

$$E = 100 \cdot \frac{418}{10000} = 100 \cdot 0.0418 = 4.18\%$$

Figure 30: Classification error

$$E = 100 \cdot \frac{n_{01} + n_{10}}{n_{00} + n_{01} + n_{10} + n_{11}}$$

$$E = 100 \cdot \frac{300 + 391}{5033 + 300 + 391 + 4276}$$

$$E = 100 \cdot \frac{691}{10000} = 100 \cdot 0.0691 = 6.91\%$$

Figure 31: Classification error

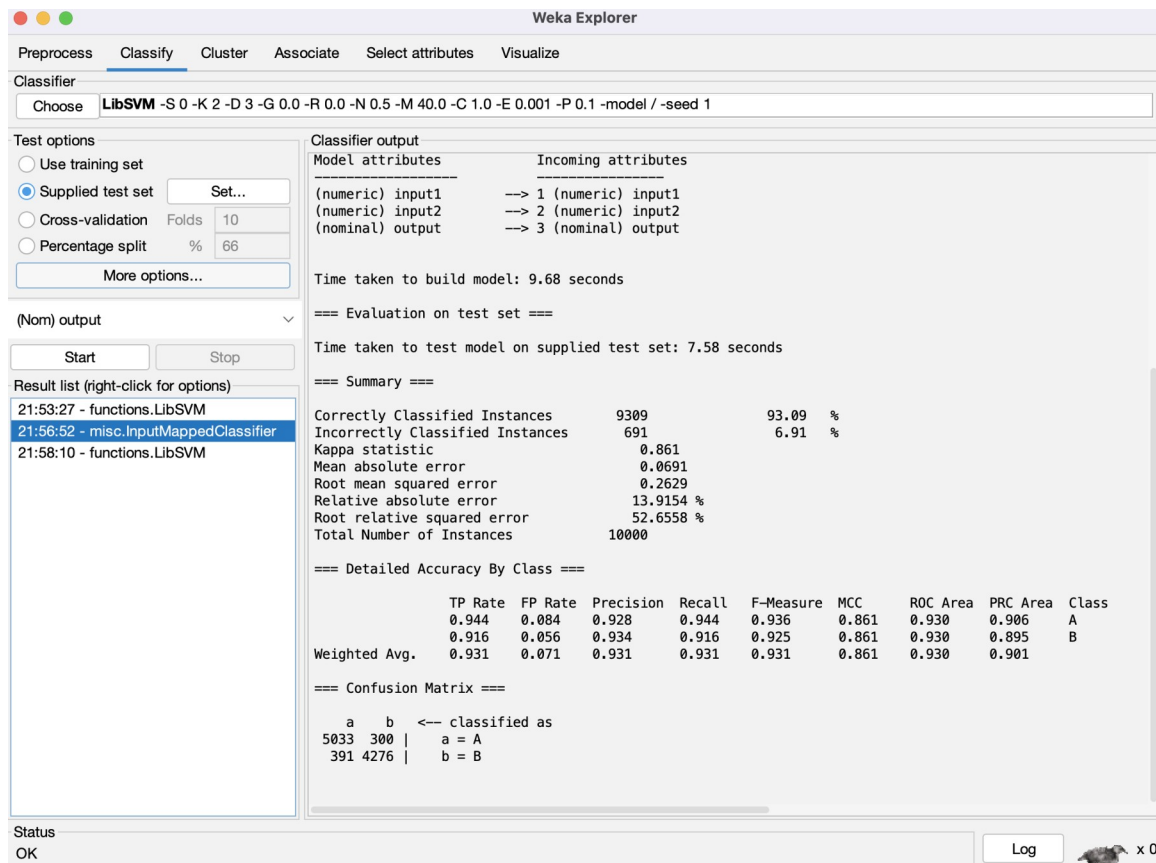


Figure 32: Weka SVM Ring confusion matrix

Below we can see the plot result generated by Weka with this dataset in comparison to the same plot generated but using the “ring-merged-weka-letters”

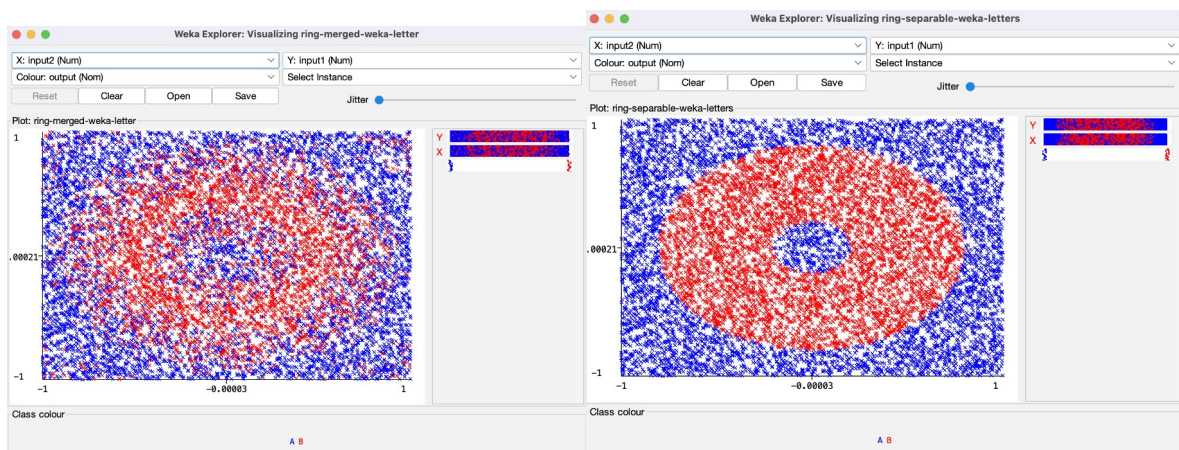


Figure 33: Weka SVM Ring prediction

Figure 34: Weka SVM Ring prediction 2

2. Multiple Linear Regression (MLR)

A Multi-Layer Perceptron (MLP) consists of multiple layers of nodes, or neurons, that is organized in a feedforward way [4]. In this type of artificial neural network (ANN) Each node in one layer connects to every node in the next one, with an associated weights.

- Input Layer: Represents features of the input data. Each node in the input layer corresponds to a feature.
- Hidden Layers: The hidden layers are layers between the input and output layers.
- Output Layer: The output layer produces the final output of the network.
- Neurons: Each node in the network but input. It receives inputs, applies a weighted sum, adds a bias term, and passes the result through an activation function.
- Activation Function: Activation functions allow the model to learn complex relationships.
- Softmax: Used in the output layer for multi-class classification.
- Loss Function: Difference between the predicted output and the target.
- Optimizer: This method helps to adjust weights and biases during training to minimize the loss function.
- Learning Rate: Step size in optimization.
- Hidden Layer Structure: This parameter specifies the number of neurons in each hidden layer.

Ring dataset's

- Ring separated

From the execution it was possible to identify that the parameters in which the accuracy is the maximum and the classification error is the minimum is:

- hidden_layer_sizes: (100, 50). This means that there are two hidden layers with 100 neurons and 50 respectively.

With the best parameters the prediction was performed :

```
Cross-Validation Scores: [0.93967993 0.93353501 0.92341579 0.93528237 0.92986988]
Mean Cross-Validation Score: 0.9323565963790147
Mean Squared Error: 0.0074
Accuracy on the test set for dataset: 0.99
Classification error on the test set for dataset: 0.74%

Confusion Matrix for Dataset:
[[5284  49]
 [ 25 4642]]

Classification Report for Dataset:

```

	precision	recall	f1-score	support
0	1.00	0.99	0.99	5333
1	0.99	0.99	0.99	4667
accuracy			0.99	10000
macro avg	0.99	0.99	0.99	10000
weighted avg	0.99	0.99	0.99	10000

Figure 35: MLP ring separated results

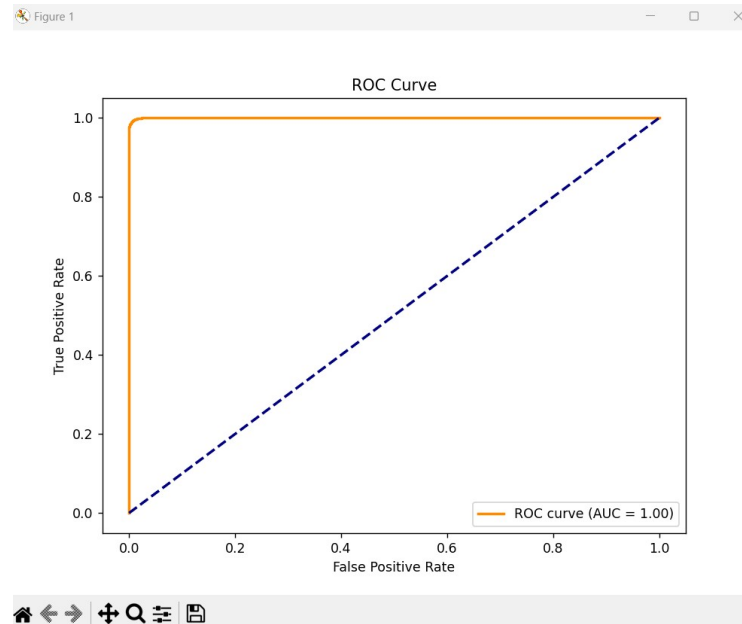


Figure 36: MLP ring separated ROC

The confusion matrix states that the model predicts more correct values than the incorrect ones, because the values in the upper left corner and in the bottom right corner are higher.

Additionally, the ROC curve shows that there curve is at top-left corner so the model is performing well [5],[6].

- Ring merged

From the execution it was possible to identify that the parameters in which the accuracy is the maximum and the classification error is the minimum is:

- hidden_layer_sizes: (100, 50). This means that there are two hidden layers with 100 neurons and 50 respectively.

With the best parameters the prediction was performed :

```
Cross-Validation Scores: [0.39305783 0.39414993 0.38920019 0.38843527 0.37362157]
Mean Cross-Validation Score: 0.38769295556691274
Mean Squared Error: 0.036
Accuracy on the test set for dataset: 0.96
Classification error on the test set for dataset: 3.60%

Confusion Matrix for Dataset:
[[5257  76]
 [ 284 4383]]

Classification Report for Dataset:
              precision    recall  f1-score   support

     0           0.95       0.99       0.97       5333
     1           0.98       0.94       0.96       4667

   accuracy              0.96       10000
  macro avg           0.97       0.96       0.96       10000
 weighted avg           0.96       0.96       0.96       10000
```

Figure 37: MLP ring merged results

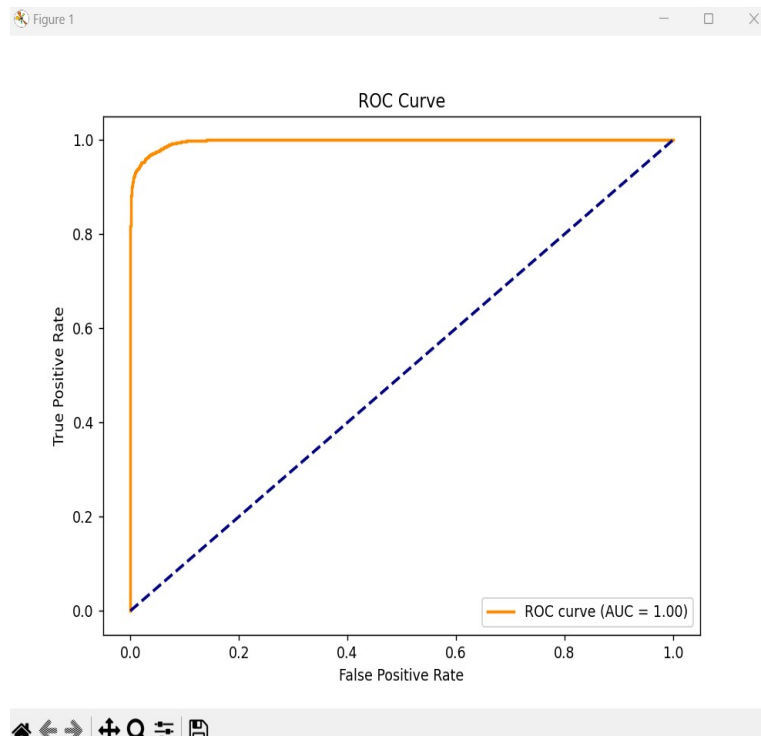


Figure 38: MLP ring merged ROC

The confusion matrix states that the model predicts more correct values than the incorrect ones, because the values in the upper left corner and in the bottom right corner are higher.

Additionally, the ROC curve shows that there curve is at top-left corner so the model is performing well.

Bank dataset

From the execution it was possible to identify that the parameters in which the accuracy is the maximum and the classification error is the minimum is:

- hidden_layer_sizes: (100, 50). This means that there are two hidden layers with 100 neurons and 50 respectively.

With the best parameters the prediction was performed with the 80% of the dataset for training and the other 20% for test:

```
Cross-Validation Scores: [0.42416587 0.32455947 0.37397634 0.29886264 0.36470233]
Mean Cross-Validation Score: 0.3572533325081451
Mean Squared Error: 0.08373786407766991
Accuracy on the test set for dataset: 0.92
Classification error on the test set for dataset: 8.37%

Confusion Matrix for Dataset:
[[729 13]
 [ 56 26]]

Classification Report for Dataset:
      precision    recall  f1-score   support

     0       0.93       0.98       0.95        742
     1       0.67       0.32       0.43         82

   accuracy          0.92          824
  macro avg       0.80       0.65       0.69          824
 weighted avg       0.90       0.92       0.90          824
```

Figure 39: Bank dataset results

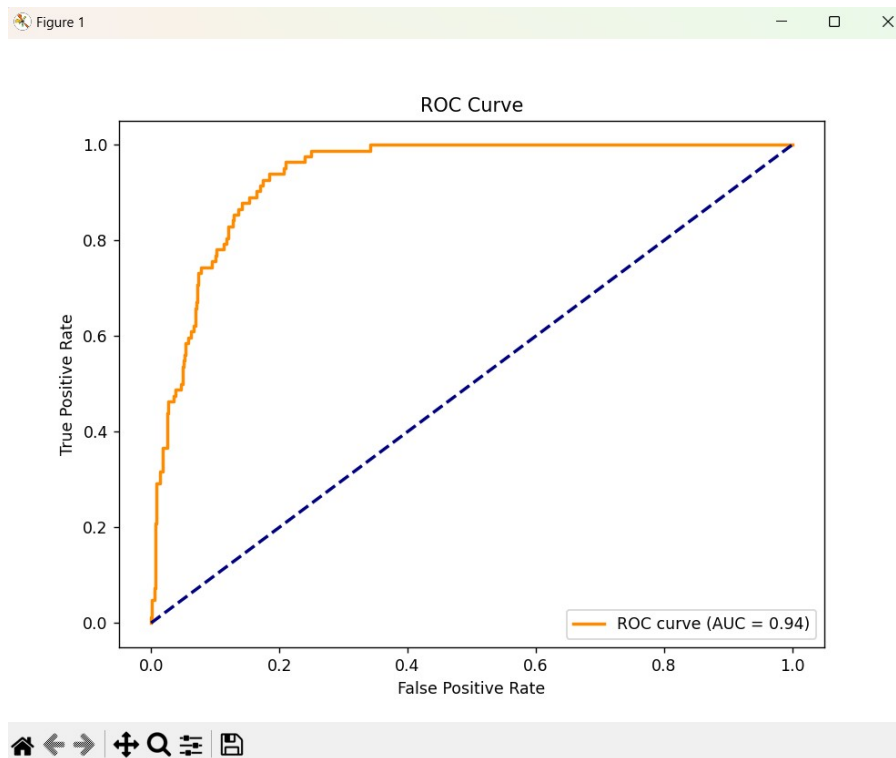


Figure 40: Bank dataset ROC

The confusion matrix states that the model predicts more correct values than the incorrect ones, because the values in the upper left corner and in the bottom right corner are higher.

Additionally, the ROC curve shows that there curve is at top-left corner so the model is performing well.

Banknote dataset

From the execution it was possible to identify that the parameters in which the accuracy is the maximun and the classification error is the minimum is:

- hidden_layer_sizes: (100, 50). This means that there arw two hidden layers with 100 neurons and 50 respectively.

With the best parameters the prediction was performed with the 80% of the dataset for training and the other 20% for test:

```
Cross-Validation Scores: [0.92374816 0.95673439 0.9468033 0.96094577 0.93285105]
Mean Cross-Validation Score: 0.9442165349926939
Mean Squared Error: 0.010948905109489052
Accuracy on the test set for dataset: 0.99

Confusion Matrix for Dataset:
[[161  3]
 [ 0 110]]
```

Classification Report for Dataset:				
	precision	recall	f1-score	support
0	1.00	0.98	0.99	164
1	0.97	1.00	0.99	110
accuracy			0.99	274
macro avg	0.99	0.99	0.99	274
weighted avg	0.99	0.99	0.99	274

Figure 41: Banknote results

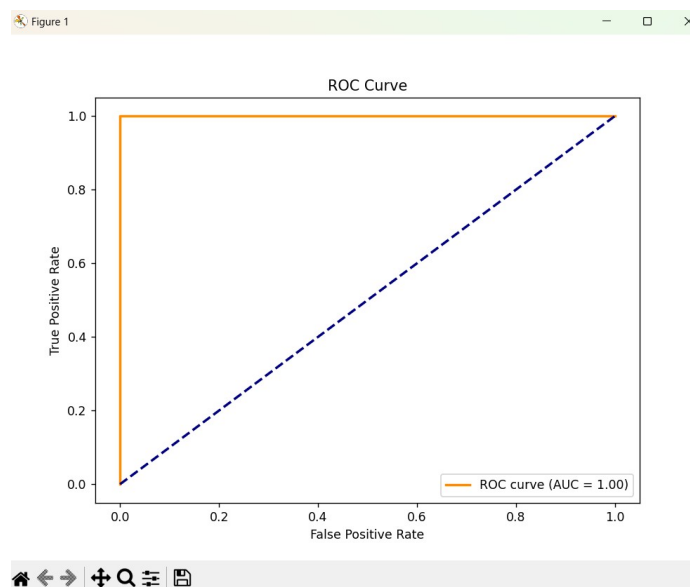


Figure 42: Banknote dataset ROC

The confusion matrix states that the model predicts more correct values than the incorrect ones, because the values in the upper left corner and in the bottom right corner are higher.

Additionally, the ROC curve shows that there curve is at top-left corner so the model is performing well.

There was used weka but other libraries this time (LinearRegression) to have a better understanding of data. However in this case the original CSV /txt file could be used. The training was performed using the default method:

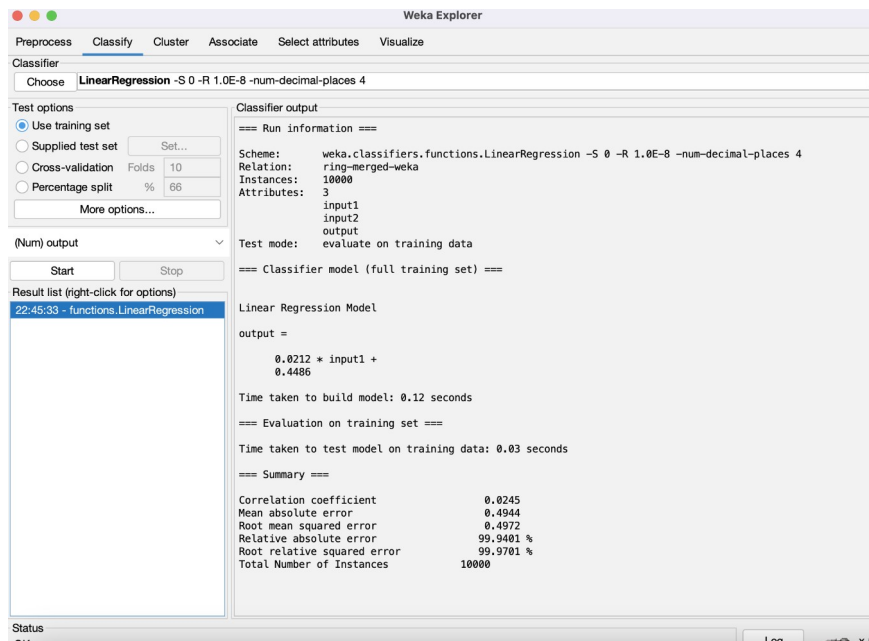


Figure 43: MLP Ring in Weka

Then we compare the results with the one for the cross-validation with 10 folds:

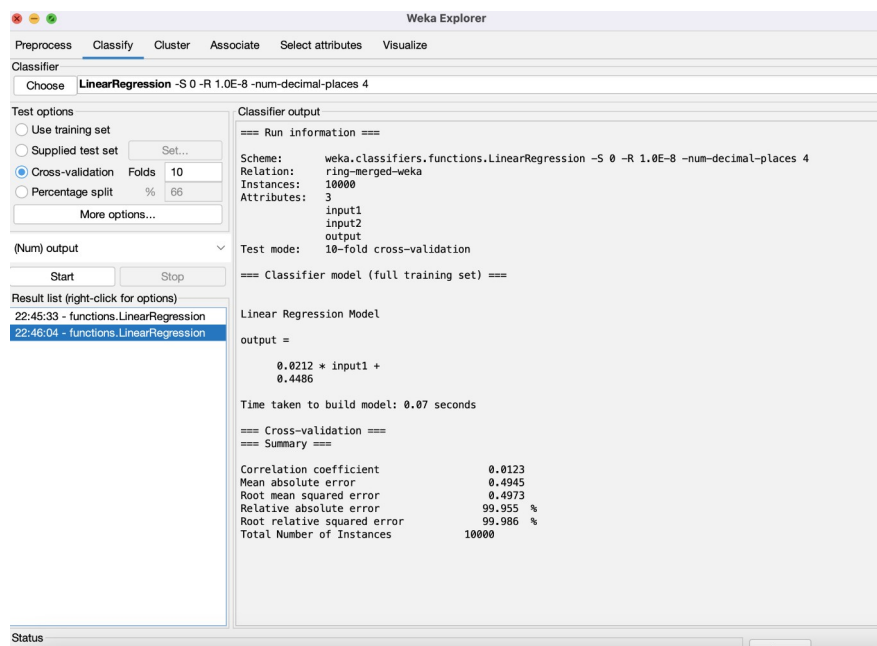


Figure 44: MLP Ring in Weka 2

By creating a table to compare the values if they are bigger or smaller than 0.5 (threshold value) we can detect the approximation of the function. For 0.5 => 4667 (sum of errors) – which is very fast.

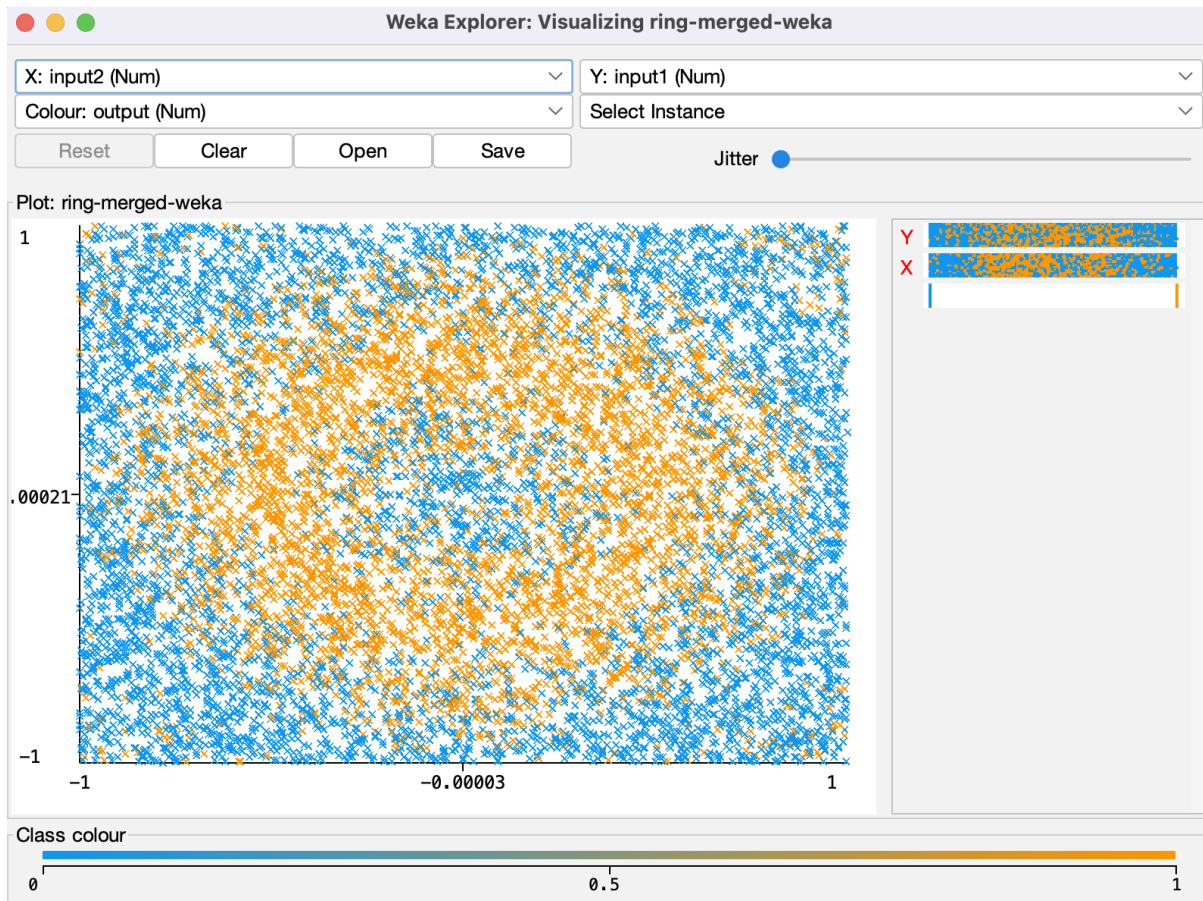


Figure 45: MLP Ring in Weka result

Results from each dataset are in the github repository.

3. Back-Propagation (BP)

We have used TensorFlow, one of the most popular when it comes to BP algorithm and Keras which is a library that provides the interface for artificial neural networks [6].

The classification error is a critical metric in evaluating the performance of our model. It measures the proportion of incorrect predictions out of the total number of instances. In a binary classification task, an error occurs when the predicted class does not match the actual class. For example, predicting that a client will subscribe ('yes') when they actually will not ('no'), or vice versa.



Figure 46 - Bank dataset

This matrix reveals that:

- 698 instances were correctly predicted as 'no' (True Negatives).
- 43 instances were incorrectly predicted as 'yes' (False Positives).
- 43 instances were incorrectly predicted as 'no' (False Negatives).
- 40 instances were correctly predicted as 'yes' (True Positives).

The classification error from the same fold is calculated to be approximately 10.44%. This error rate is a significant metric, considering the imbalance in the dataset; it reflects the proportion of misclassifications.

For the **banknote_authentication** dataset, we've designed a neural network with the following specifications:

- **Learning Rate:** 0.05, which governs the speed at which the model learns. A moderate rate helps in converging to a global minimum without overshooting.
- **Momentum:** 0.1, to help accelerate the optimizer in the correct direction and prevent oscillations.
- **Neural Network Architecture:** The network consists of an input layer with 4 neurons (one for each feature), two hidden layers with 20 and 5 neurons respectively, and an output layer with a single neuron.



Figure 47 - Banknote dataset

This confusion matrix tells us that:

- 156 instances of genuine banknotes were correctly identified (True Negatives).
- 5 instances of genuine banknotes were incorrectly classified as forgeries (False Positives).
- 0 instances of forged banknotes were incorrectly classified as genuine (False Negatives).
- 114 instances of forged banknotes were correctly identified (True Positives).

From the given fold, the classification error is remarkably low, at approximately 1.82%. This low error rate indicates a high level of accuracy in the model's ability to classify banknotes as genuine or counterfeit.

A 4-fold cross-validation technique is also used for the ring dataset to validate the model's performance and ensure its ability to generalize to new data. This technique involves partitioning the dataset into four subsets, training on three subsets, and validating on the fourth. This process is repeated four times, with each subset used once as the validation set.



Figure 48 - Ring dataset

Interpreting the confusion matrix:

- 1233 True Negatives (TN): Correctly predicted negative class instances.
- 82 False Positives (FP): Negative class instances incorrectly predicted as positive.
- 7 False Negatives (FN): Positive class instances incorrectly predicted as negative.
- 1178 True Positives (TP): Correctly predicted positive class instances.

The classification error, derived from the confusion matrix, is approximately 3.56%, indicating the model's high accuracy in classifying the instances. This low error rate is particularly noteworthy in a dataset with a complex, non-linear decision boundary like that of the ring dataset.

The custom BP algorithm has been updated from its original form in the first assignment (A1), leading to enhanced functionality and now works. Here are the key improvements:

- **Enhanced Cross-Validation Methodology:** The addition of the 'trainingFolds' parameter to the BP algorithm is a major change. The 'folds' refer to the different segments that this new feature creates from the data and epochs. Training and validation sets are rotated using each fold in a methodical manner, which results in a more complete and reliable training procedure.
- **Modification to the "CostFunction" Method:** A new parameter has been added to the algorithm's "costFunction" method. This modification is essential for precisely computing the classification error, particularly in light of the 'forward' method's

adoption of a threshold. The formula..... is used to calculate the error, where..... stands for various classification counts.

- **Normalization Procedures Are Removed:** The normalization and denormalization stages of data processing are eliminated in the current version of the BP algorithm. The 'readDataAndNormalize' method has been modified due to the dataset being utilized already having a normalized format.
- **Threshold Application in the 'Forward' Method:** A notable change is the 'forward' method's addition of a new parameter. This update is necessary for testing since it applies a threshold to the final layer's output, turning it into binary form (0 or 1).

Part 2.1: Parameter selection

SVM:

- Ring dataset:
 - Separable dataset
 - Kernel: rbf
 - Constant:701
 - Merged dataset
 - • Kernel: rbf
 - • Constant:701
- Bank dataset
 - Kernel: poly
 - Constant:1
 - Degree:4
- Banknote dataset
 - Kernel: rbf
 - Constant:2

MLP:

- Ring dataset
 - Separable dataset
 - hidden_layer_sizes=(100, 50)
 - Merged dataset
 - hidden_layer_sizes=(100, 50)
- Bank dataset
 - hidden_layer_sizes=(100, 50)
- Banknote dataset
 - hidden_layer_sizes=(100, 50)

BP:

- Ring dataset
 - Separable dataset
 - learning_rate = 0.05
 - momentum = 0.1
 - nn = [2, 16, 8, 1]
 - Merged dataset
 - learning_rate = 0.05
 - momentum = 0.1
 - nn = [2, 16, 8, 1]
- Bank dataset
 - learning_rate = 0.01
 - momentum = 0.05
 - nn = [20, 100, 25, 5, 1]
- Banknote dataset
 - learning_rate = 0.05
 - momentum = 0.1
 - nn = [4, 20, 5, 1]

Part 2.2: Evaluation of the results

SVM:

- Ring dataset

	Separable dataset	Merged dataset
Cross-validation score	0.997	0.7789
Classification error on test set	0.25%	4.39%
Accuracy	1.0	0.96

- Bank dataset

Cross-validation score	0.9077
Classification error on test set	7.89%
Accuracy	0.92

- Banknote dataset

Cross-validation score	0.9973
Classification error on test set	0.00%
Accuracy	1.00

MLP:

- Ring dataset

	Separable dataset	Merged dataset
Cross-validation score	0.9323	0.387
Classification error on test set	0.74%	3.60%
Accuracy	0.99	0.96

- Bank dataset

Cross-validation score	0.3573
Classification error on test set	8.37%
Accuracy	0.92

- Banknote dataset

Cross-validation score	0.9442
Classification error on test set	1.09%
Accuracy	0.99

BP:

- Ring dataset

	Separable dataset	Merged dataset
Cross-validation score	0.9774	0.774
Classification error on test set	2.23%	22.36%
Accuracy	0.9777	0.7876

- Bank dataset

Cross-validation score	0.9174
Classification error on test set	10.80%
Accuracy	0.8920

- Banknote dataset

Cross-validation score	0.9936
Classification error on test set	0.73%
Accuracy	0.9927

Confusion matrix comparation:

	SVM	MLP	BP
Ring	[[5321 12] [13 4654]] [[5323 10] [429 4238]]	[[5284 49] [25 4642]] [[5257 76] [284 4383]]	[[5188 145] [40 4627]] [[983 402] [192 923]]
Bank	[[731 11] [54 28]]	[[729 13] [56 26]]	[[698 43] [43 40]]
Banknote	[[164 0] [0 110]]	[[161 3] [0 110]]	[[156 5] [0 114]]

For all cases it is possible to identify that the left upper corner and the right bottom one vales for all datasets and methods have higher values compared to the other two corners. This means that most values are TP and TN, so that the model correctly predicted positive instances and that the model correctly predicted negative instances.

The execution of the models for the datasets and the selected parameters is highly accurate and reliable and the risk of false positives and false negative is minimal.

As detailed in each section the ROC Curve for each model and dataset it shows that there curves are at top-left corner so models are performing well.

Even if they all got good predictions, depending on the parameters used the results could be even better.

GITHUB – REPOSITORY:

<https://github.com/sebastianbuzdugan/A2-NeuralNetworks>

References

- [1] UC Irvine Machine Learning. (2023). *banknote authentication - UCI Machine Learning* Accessed: Dec. 02, 2023. [Online]. Available: <https://archive.ics.uci.edu/dataset/267/banknote+authentication>
- [2] scikit-learn, “sklearn.svm.SVC — scikit-learn 1.3.2 documentation.” Accessed: Dec. 02, 2023. [Online]. Available: <https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html>
- [3] The University of Waikato, “Weka 3 - Data Mining with Open Source Machine Learning Software in Java.” Accessed: Dec. 02, 2023. [Online]. Available: <https://www.cs.waikato.ac.nz/ml/weka/>
- [4] scikit-learn, “sklearn.neural_network.MLPRegressor — scikit-learn 1.3.2 documentation.” Accessed: Dec. 02, 2023. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPRegressor.html
- [5] scikit-learn, “3.1. Cross-validation: evaluating estimator performance — scikit-learn 1.3.2 documentation.” Accessed: Dec. 02, 2023. [Online]. Available: https://scikit-learn.org/stable/modules/cross_validation.html
- [6] scikit-learn, “Multiclass Receiver Operating Characteristic (ROC) — scikit-learn 1.3.2 documentation.” Accessed: Dec. 02, 2023. [Online]. Available: https://scikit-learn.org/stable/auto_examples/model_selection/plot_roc.html
- [7] Tensorflow, “Keras: la API de alto nivel para TensorFlow | TensorFlow Core.” Accessed: Dec. 02, 2023. [Online]. Available: <https://www.tensorflow.org/guide/keras?hl=es>