**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:**

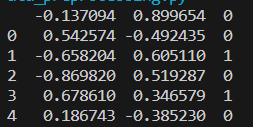
The task involves data classification using Support Vector Machines (SVM), Back-Propagation (BP), and Multiple Linear Regression (MLR) on different datasets. Key aspects include parameter selection via cross-validation, data preprocessing, normalization, and evaluation through error rates, confusion 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**

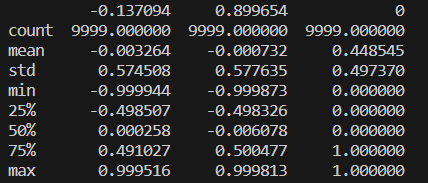
**Ring dataset**

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

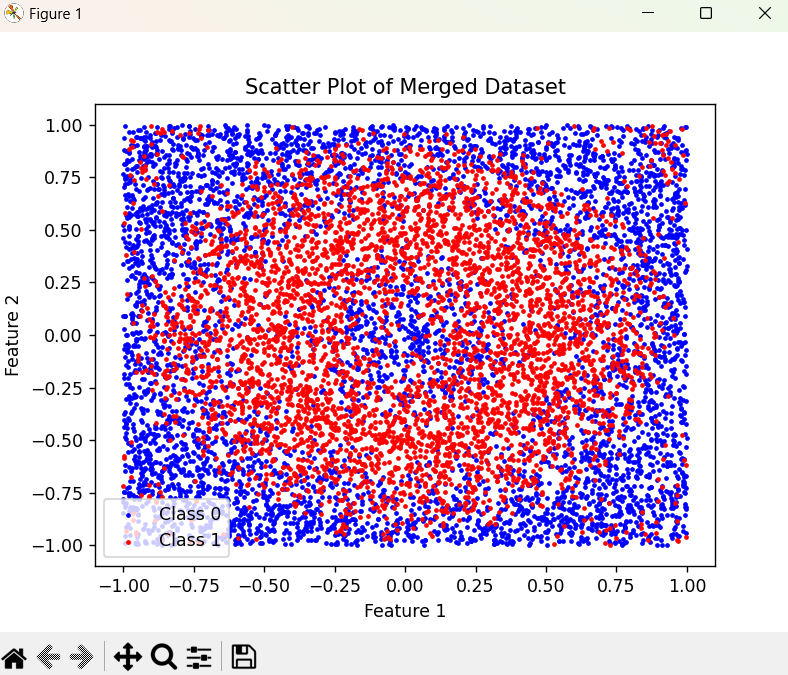
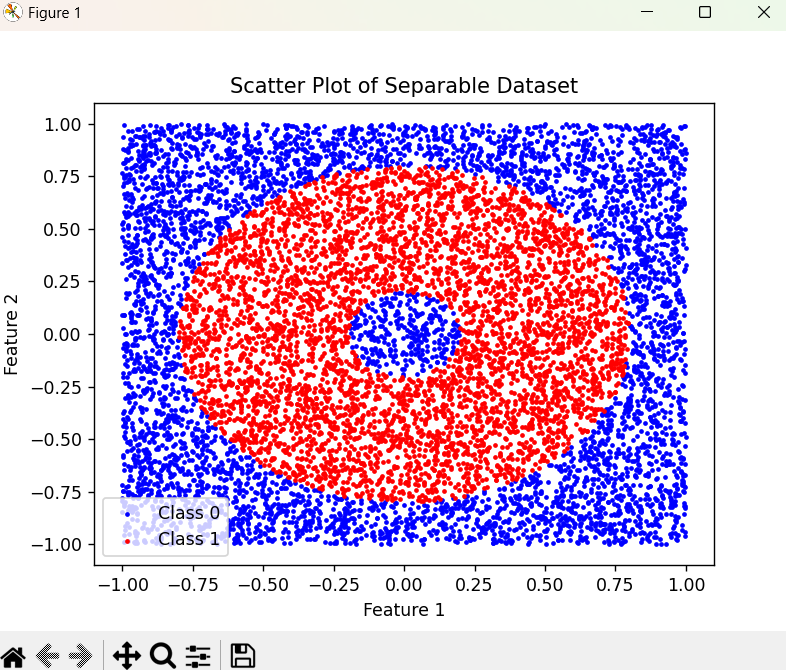
The head of the dataset is the following:



The descriptive statistics of the dataset are the following:



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

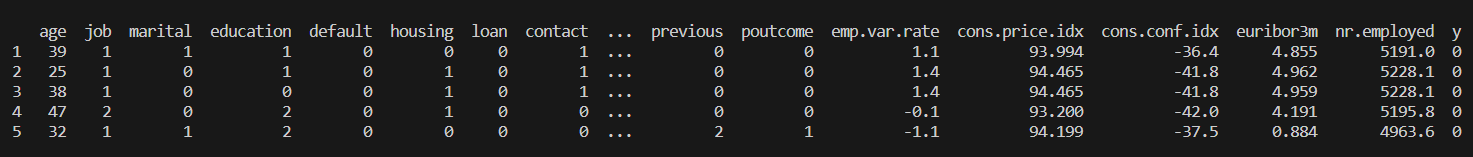
****

The variables of the dataset are:

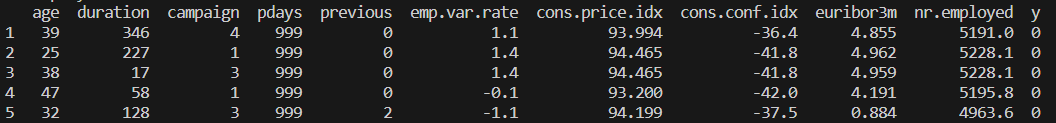
* "age": numerical
* "job":categorical
* "marital":categorical
* "education":categorical
* "default":categorical
* "housing"categorical
* "loan":categorical
* "contact":categorical
* "month":categorical
* "day\_of\_week":categorical
* "duration":numerical
* "campaign":numerical
* "pdays":numerical
* "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 clases, 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:



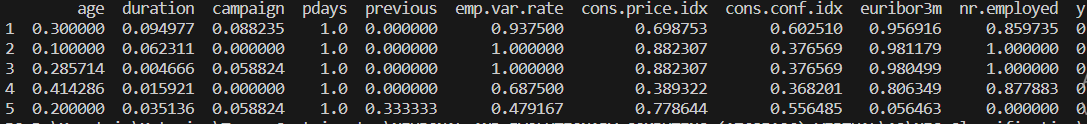
Since, the last column is using as classifier, the rest of categorical variables of the dataset are not considered.

****

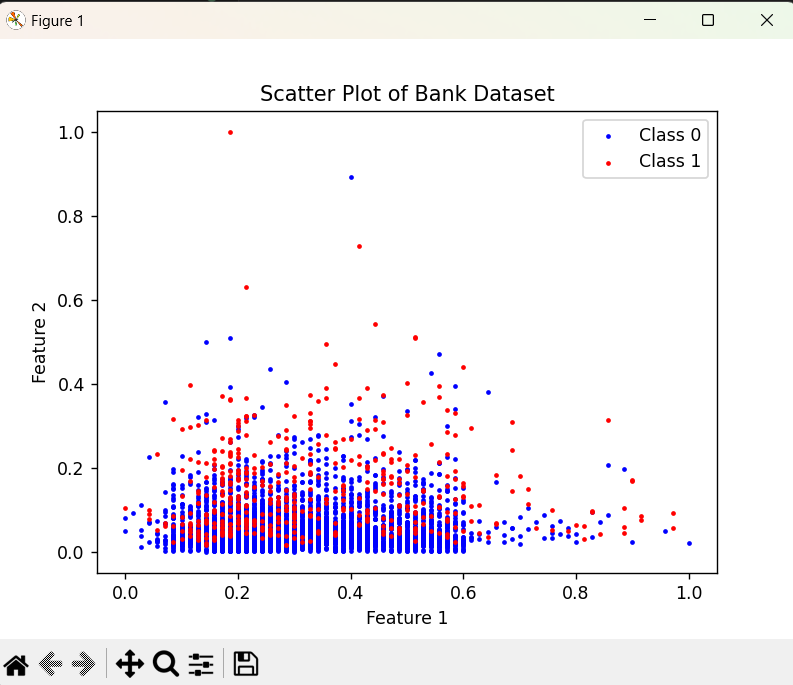
Finally data was normalized as follows:

* 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:



Here below there is the plot of the Banknote dataset:

****

**Personalized dataset: Banknote authentication**

**Dataset variables**

The personalized dataset was obtained from the Univesity of California, Irvine Machine Learning Repository. 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)

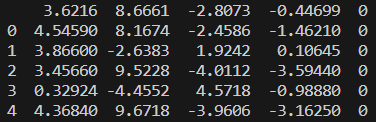
2. skewness of Wavelet Transformed image (continuous)

3. curtosis of Wavelet Transformed image (continuous)

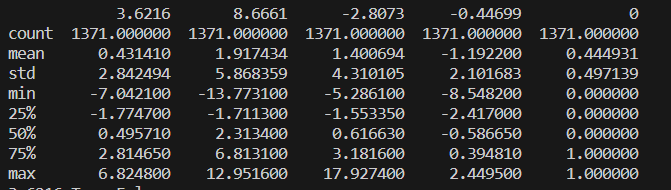
4. entropy of image (continuous)

5. class (integer)

Here below there is the head of the dataset:



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:

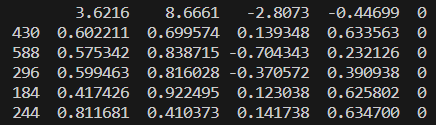


Then data normalization was applied in numerical variables as follows:

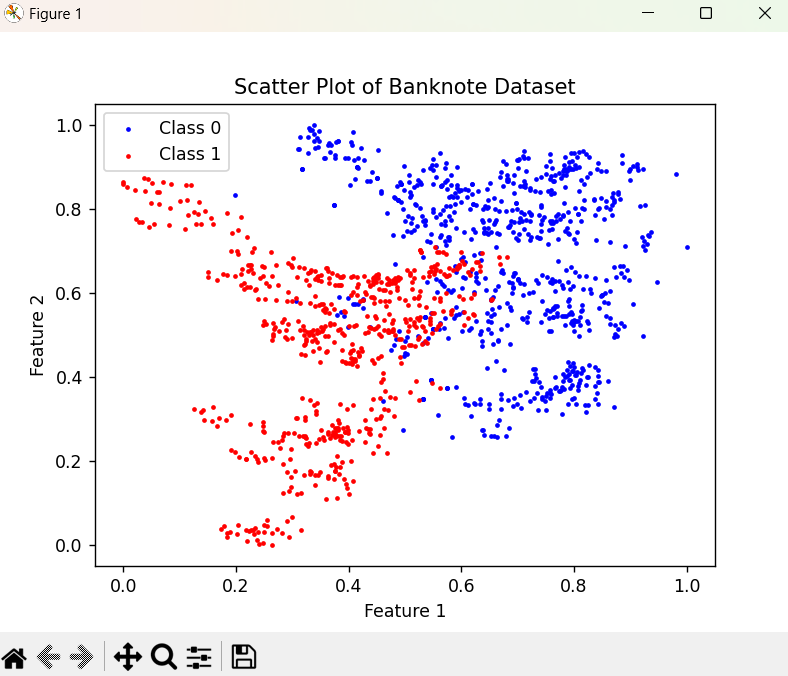
* Variance: There are positive values that vary widely, so it will be normalized with Min-Max scaling between 0 and 1.
* Skewness: There are positive values that vary widely, so it will be normalized with Min-Max scaling between 0 and 1.
* Curtosis: It follows normal distribution so Z-score normalization 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:



Here below there is the plot of the Banknote dataset:



**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.

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:

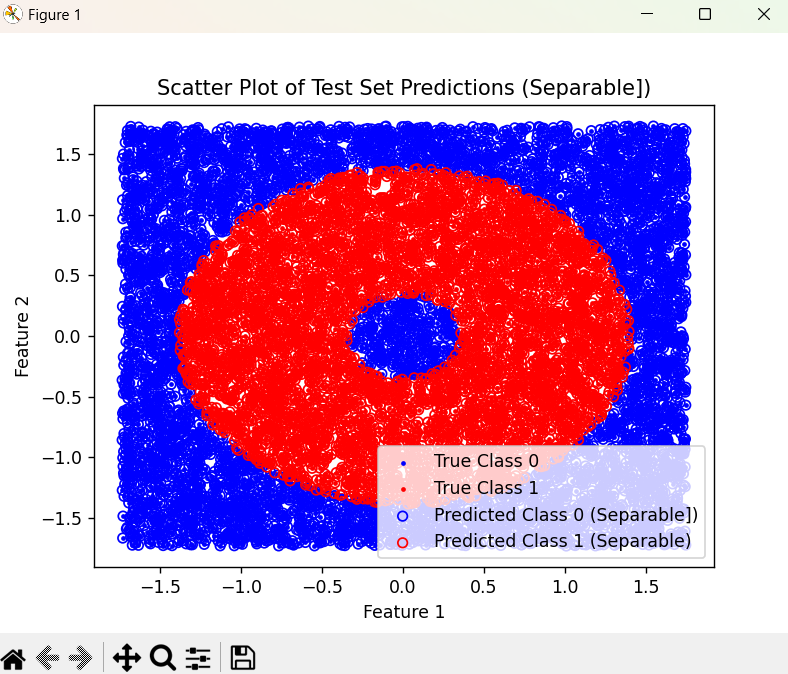
* Kernel:'linear', 'poly', 'rbf', 'sigmoid', 'precomputed'
* Constant:100,200,300,400,500,600,700,800,900

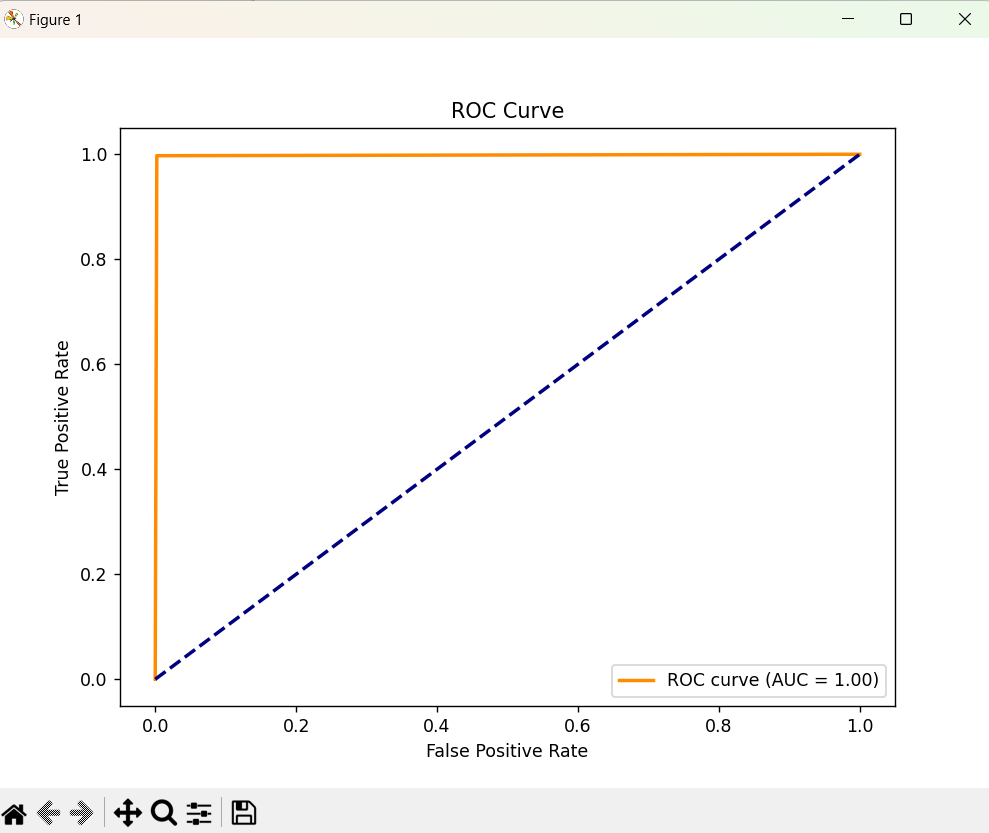
From the execution it was possible to identify that the parameters in which the accuracy is the maximun and the classification error is the minimum is:

* Kernel: rbf
* Constant:701

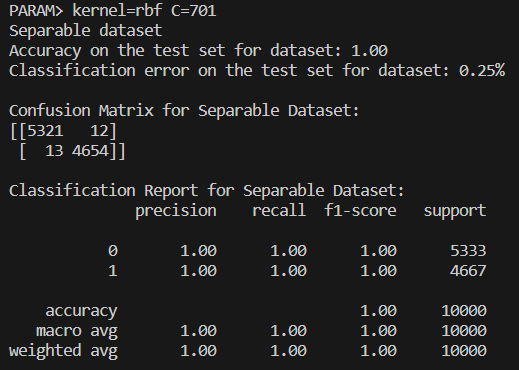
With the best parameters the prediction was performed with the 80% of the dataset for training and the other 20% for test:

* Separable dataset









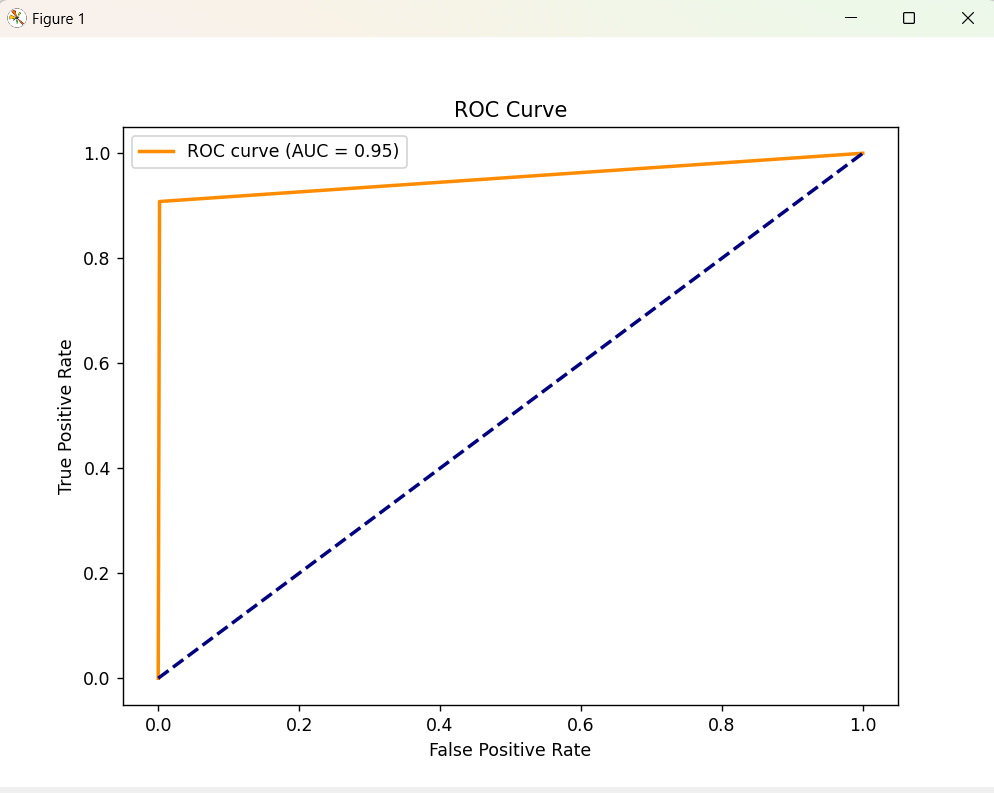
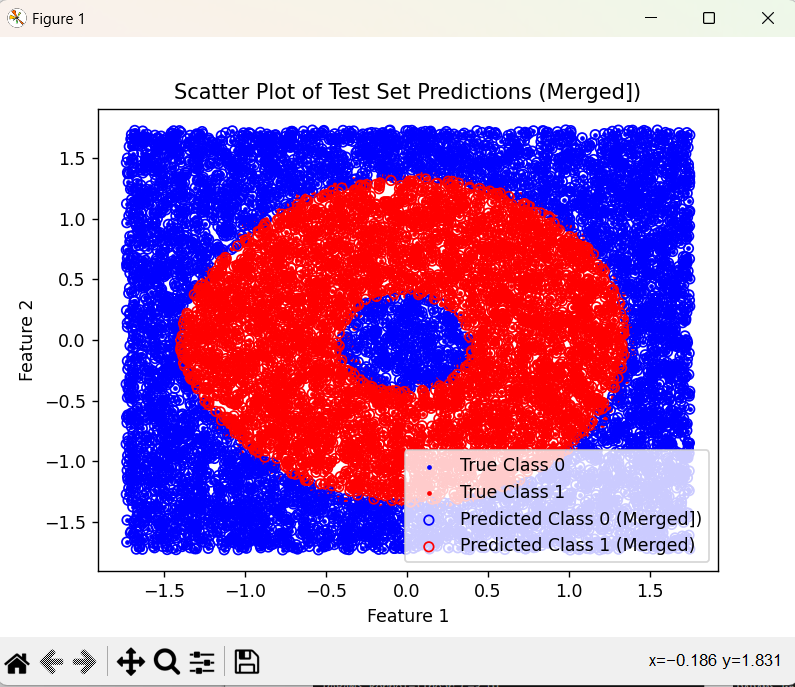
The confusion matrix is detailed as follows:

* 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.

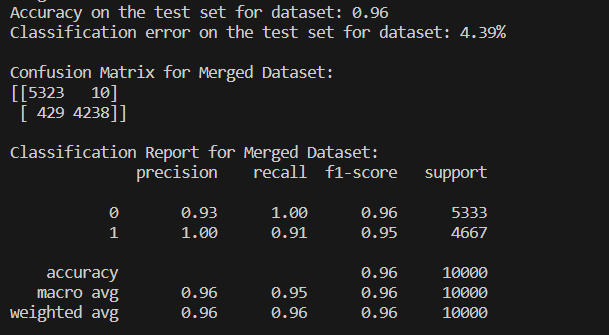
From this definitions, the confusion matrix states that the model predicts more correct values than the incorrect ones, because the values in the upper left corner and in the bottom right corner are higher.

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

* Merged dataset:







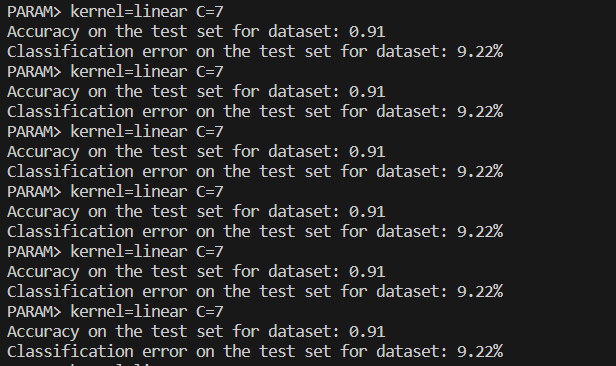
The confusion matrix states that the model predicts more correct values than the incorrect ones, because the values in the upper left corner and in the bottom right corner are higher.

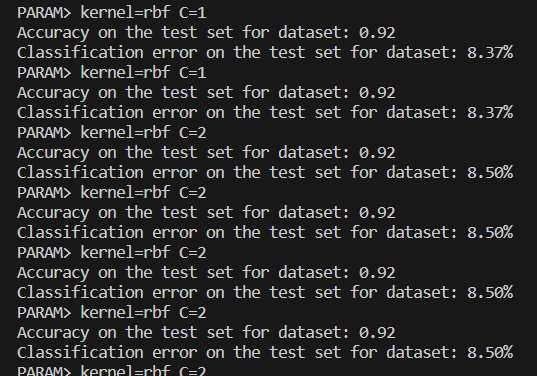
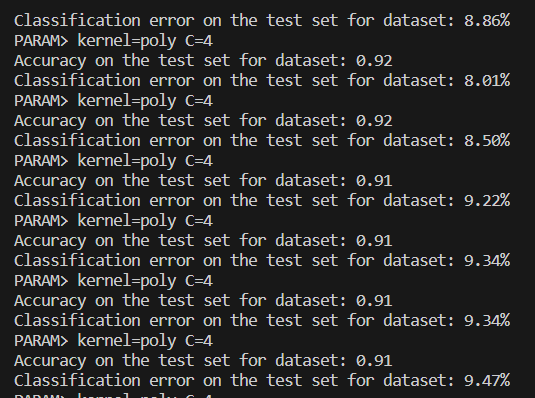
Additionally, the ROC curve shows that there curve is at top-left corner so the model is performing well.

**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:1,2,3,45

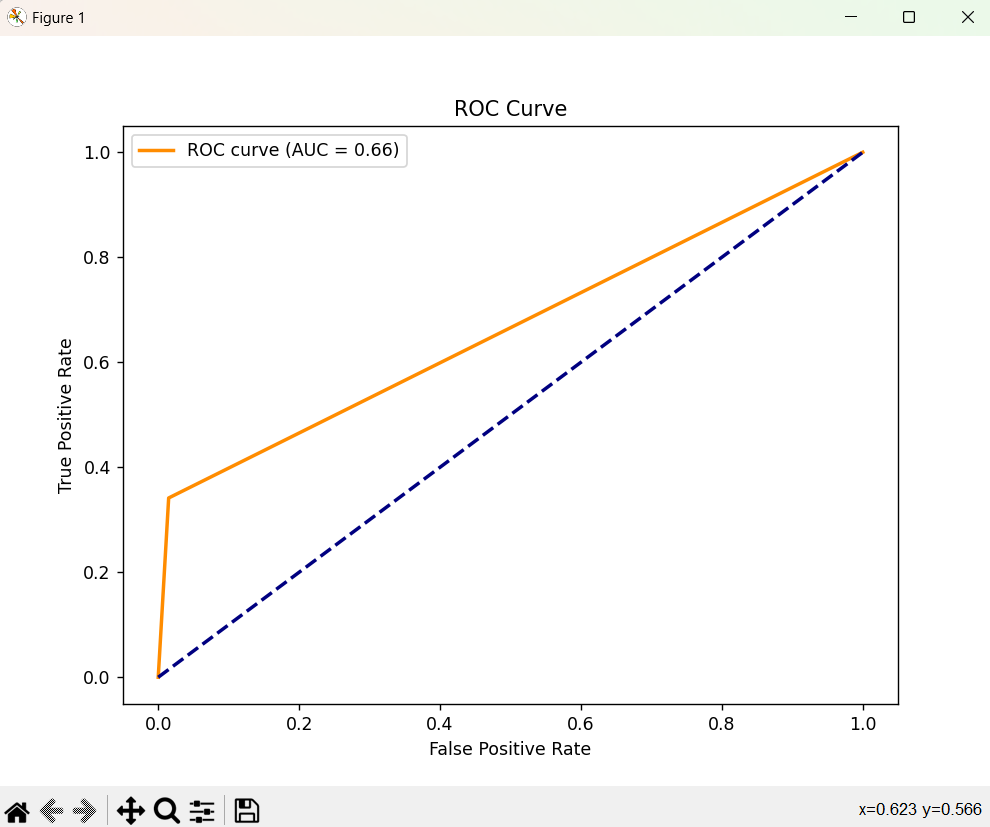


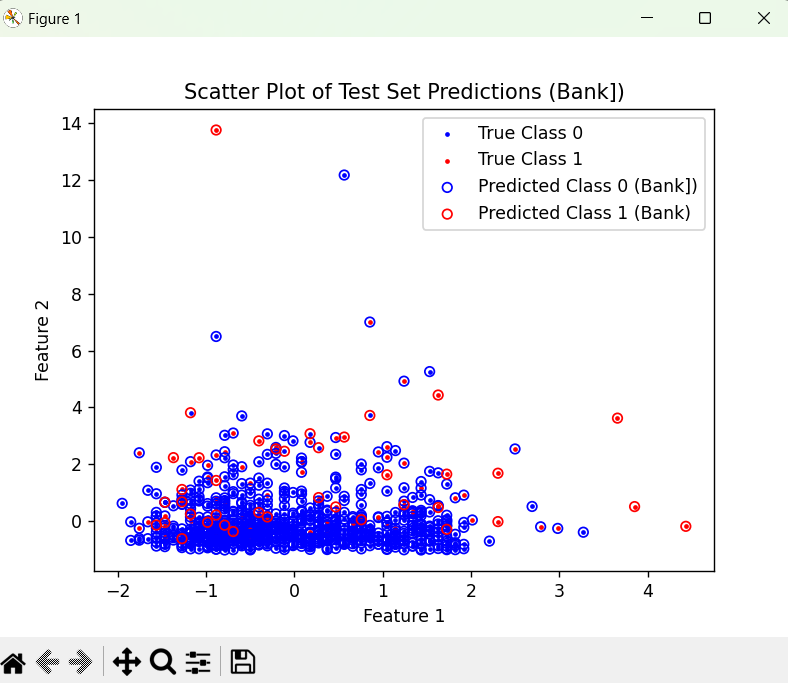


From the execution it was possible to identify that the parameters in which the accuracy is the maximun and the classification error is the minimum is:

* Kernel: poly
* Constant:1
* Degree:4

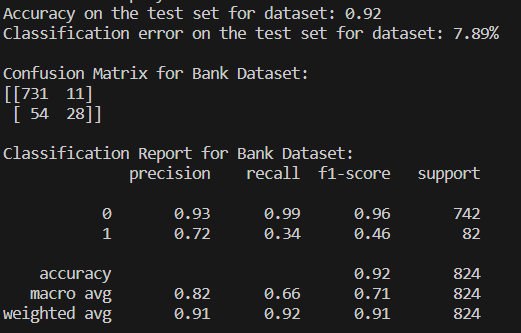
With the best parameters the prediction was performed with the 80% of the dataset for training and the other 20% for test:





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:

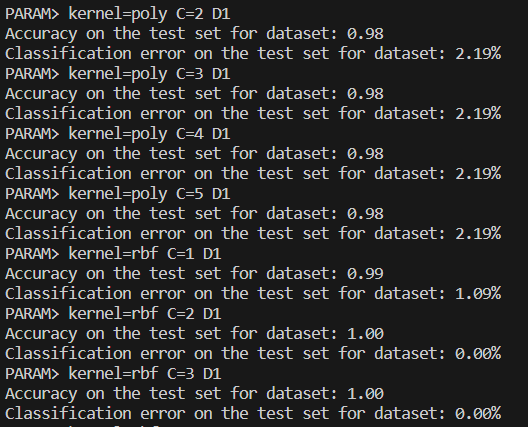
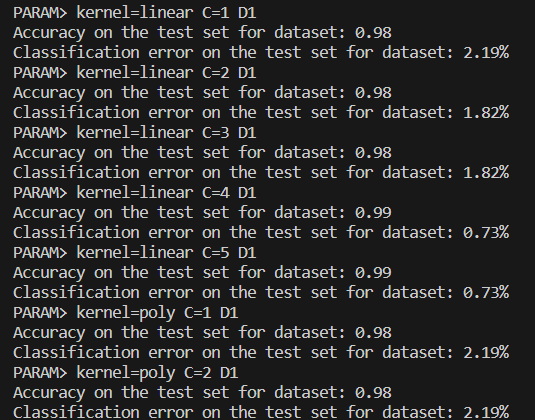


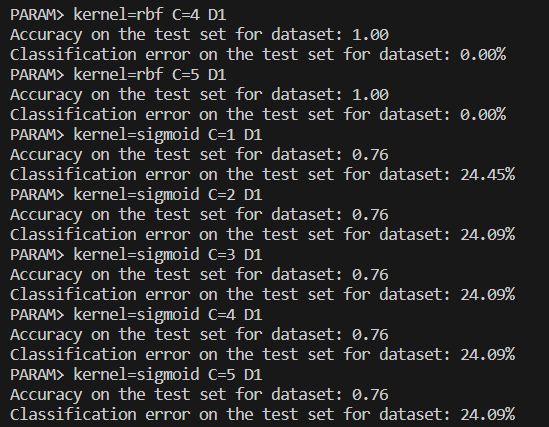


**Banknote dataset**

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

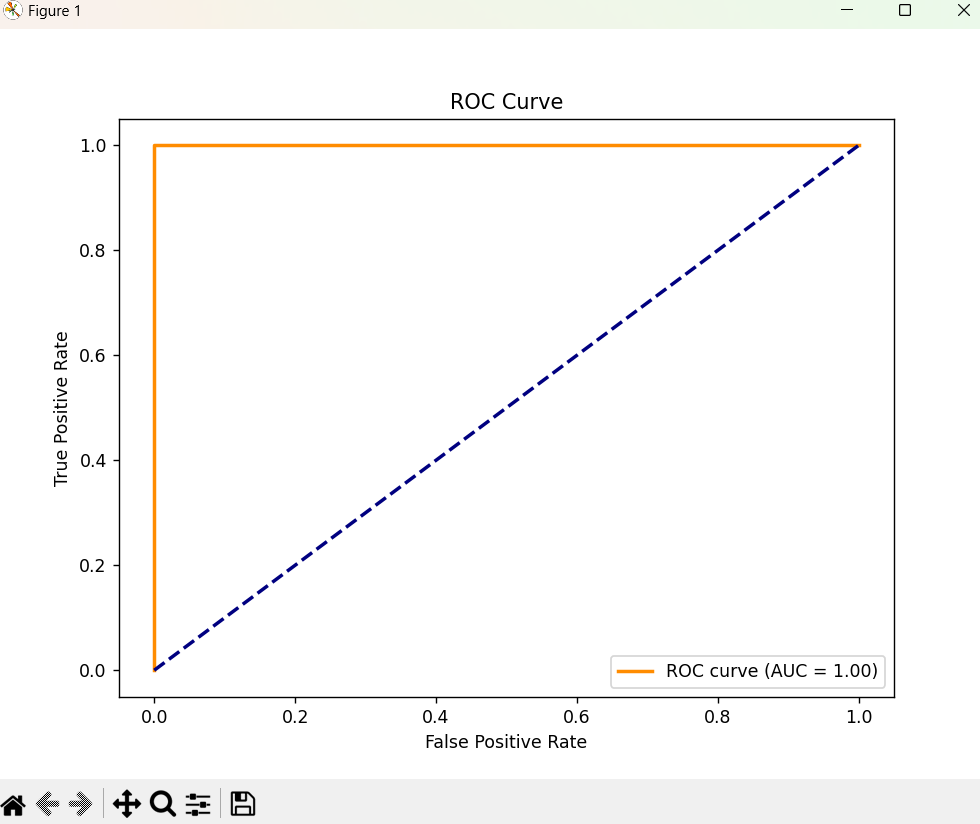
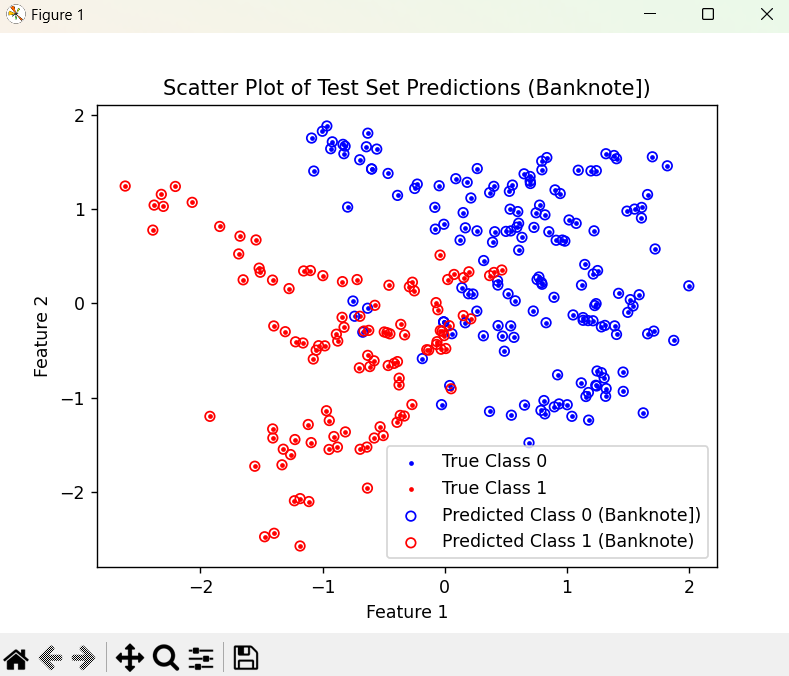




From the execution it was possible to identify that the parameters in which the accuracy is the maximun and the classification error is the minimum is:

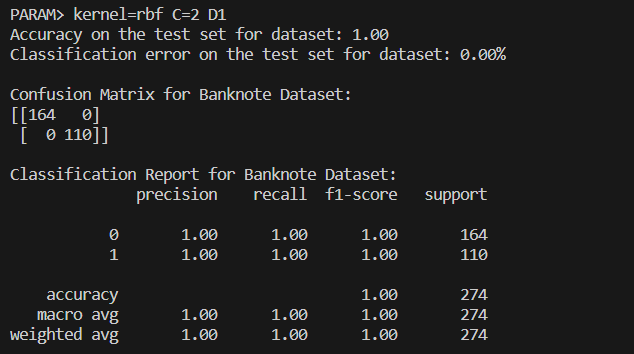
* Kernel: rbf
* Constant:2

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



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:





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. 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.

A screenshot of a computer

Description automatically generated

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:

A screenshot of a computer

Description automatically generated

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

A math equations with numbers

Description automatically generatedA math equations with numbers

Description automatically generated

A screenshot of a computer

Description automatically generated

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”

A screenshot of a computer screen

Description automatically generatedA screen shot of a computer

Description automatically generated

1. **Multiple Linear Regression (MLR)**

A Multi-Layer Perceptron (MLP) consists of multiple layers of nodes, or neurons, that is organized in a feedforward way. 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:Representa 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.

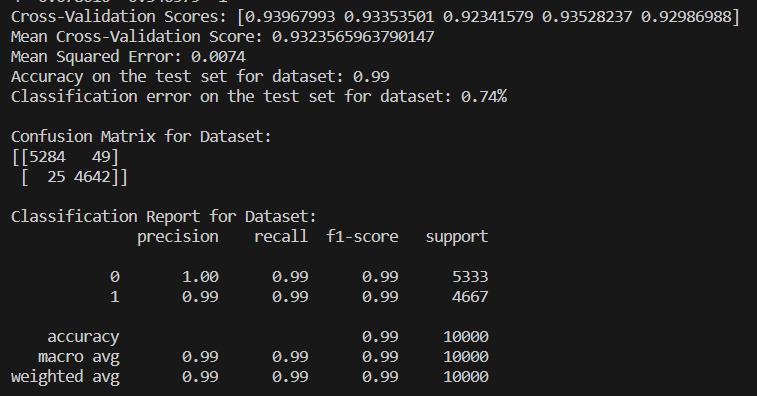
**Ring dataset's**

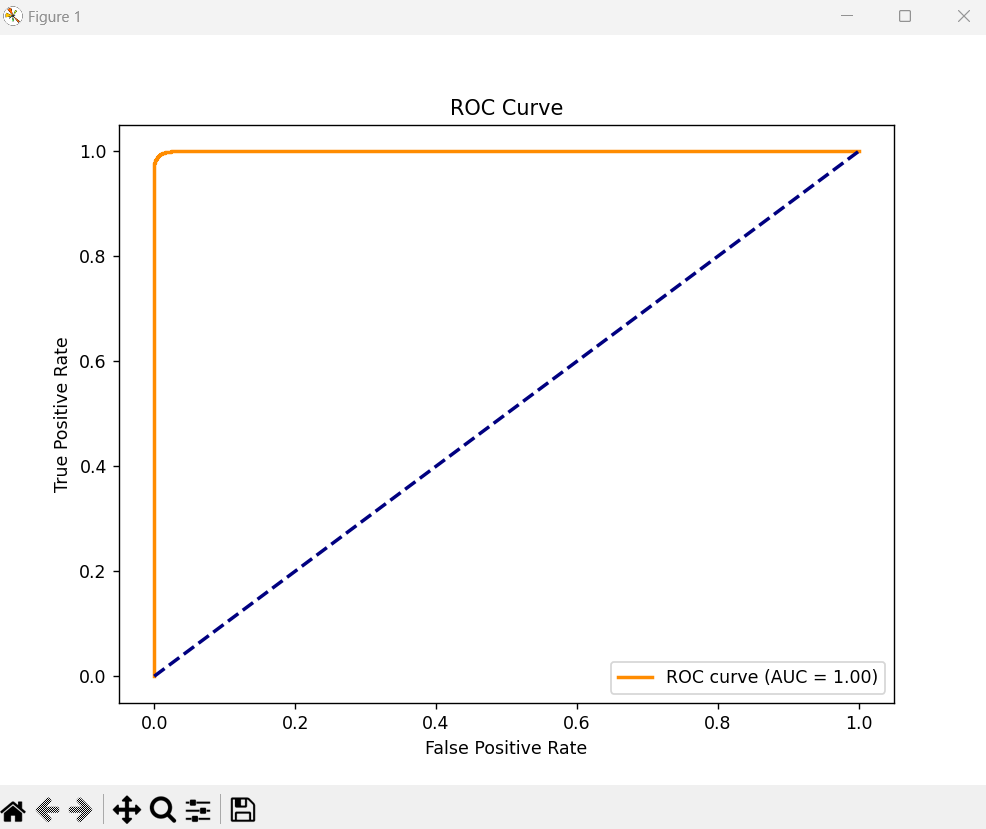
* Ring separated

From the execution it was possible to identify that the parameters in which the accuracy is the 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 :





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.

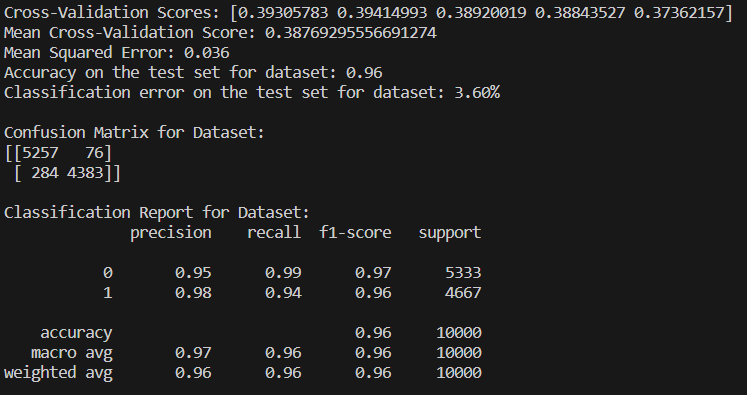
Additionally, the ROC curve shows that there curve is at top-left corner so the model is performing well.

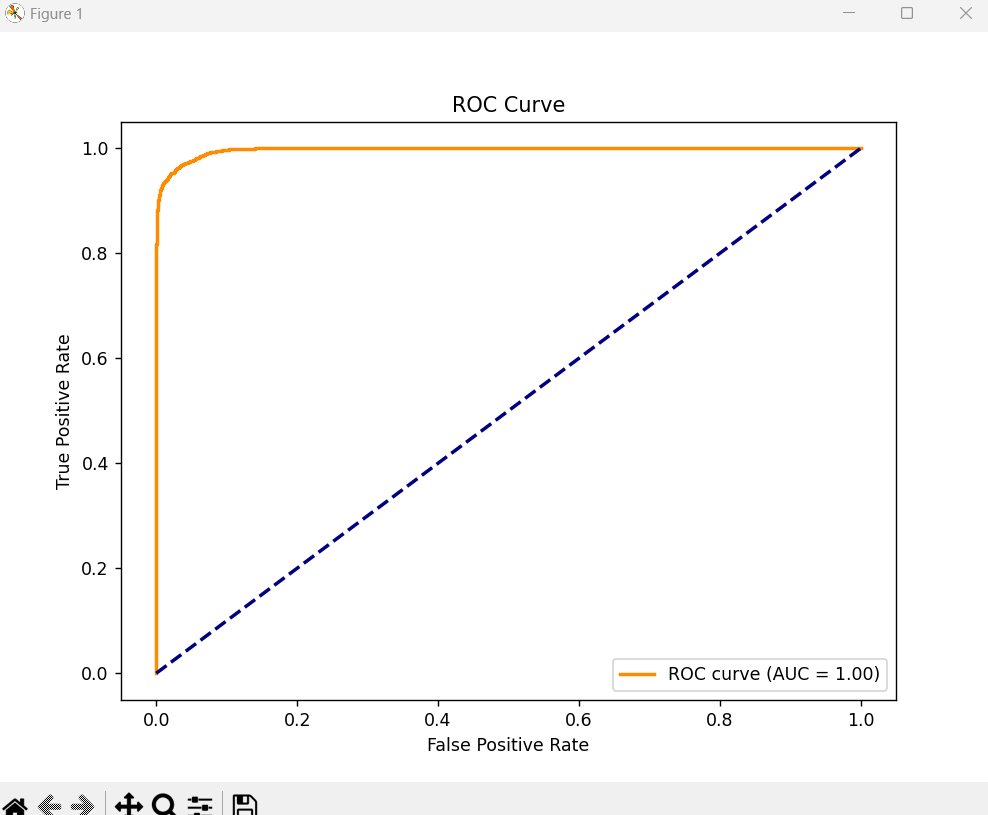
* Ring merged

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 :





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.

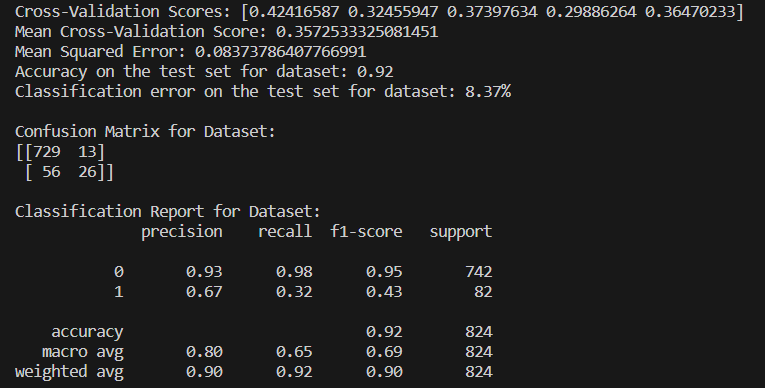
Additionally, the ROC curve shows that there curve is at top-left corner so the model is performing well.

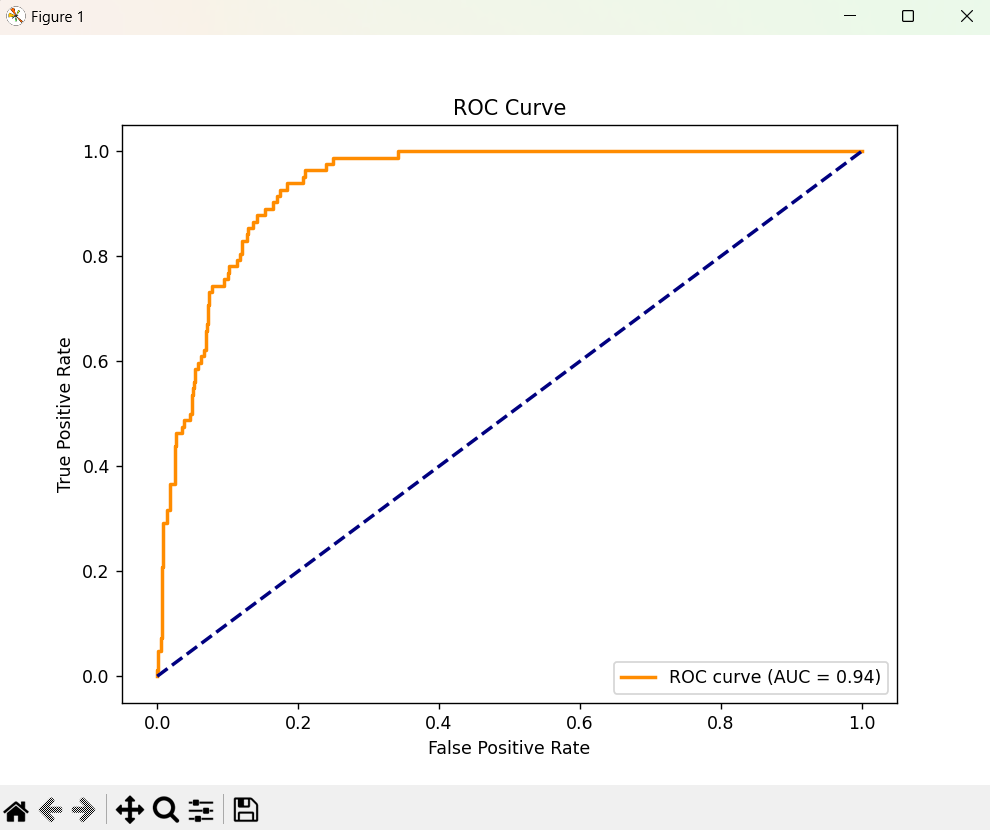
**Bank dataset**

From the execution it was possible to identify that the parameters in which the accuracy is the 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:





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.

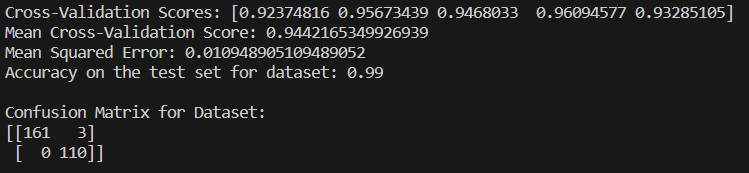
Additionally, the ROC curve shows that there curve is at top-left corner so the model is performing well.

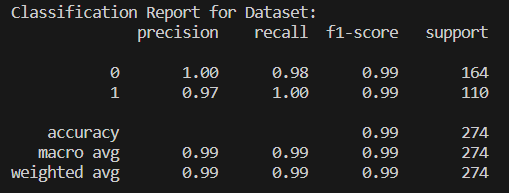
**Banknote dataset**

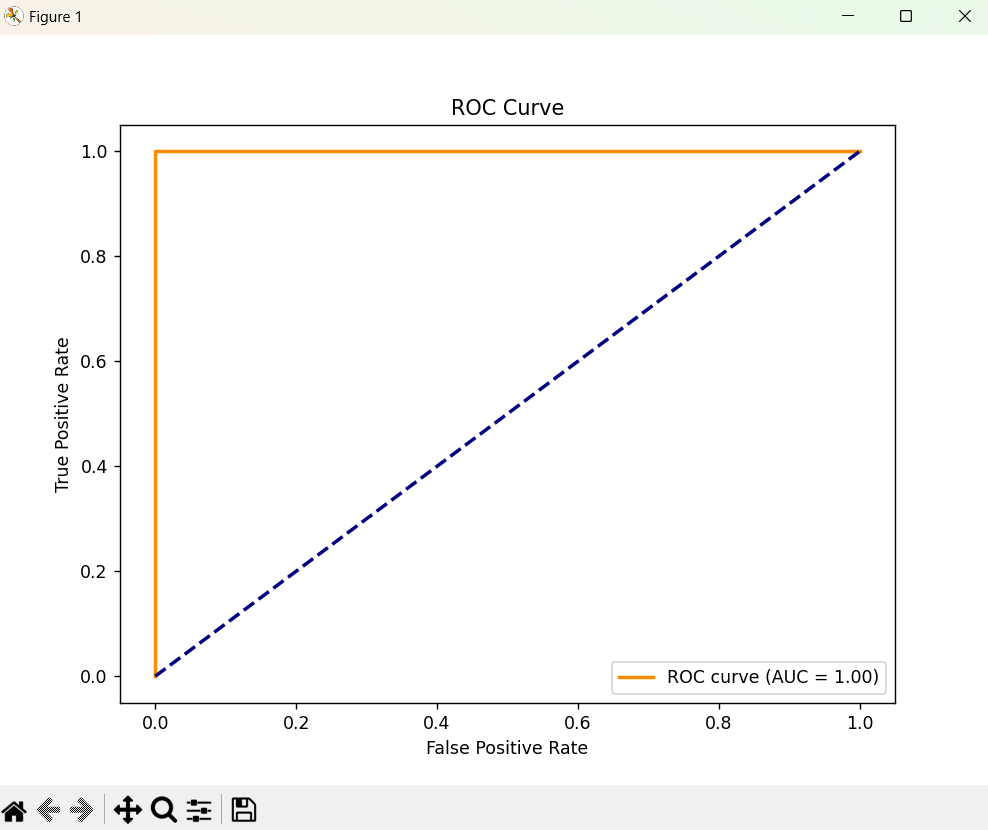
From the execution it was possible to identify that the parameters in which the accuracy is the maximun and the classification error is the minimum is:

* 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:



****

****

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:

A screenshot of a computer

Description automatically generated

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

A screenshot of a computer

Description automatically generated

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

A screenshot of a computer

Description automatically generated

Results from each dataset are in the github repository.

1. **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.

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.

A screenshot of a computer

Description automatically generated

Figure 1 - 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.

A screenshot of a computer

Description automatically generated

Figure 2 - 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.

A screenshot of a computer

Description automatically generated

Figure 3 - 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.

GITHUB – REPOSITORY:

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