Machine Learning
University of La Sapienza
Fall 2023
Rome, Italy

Homework 1

Alessia Pontiggia 1892079

1 Introduction

The task of this first homework is a multiclass (10 classes) classification problem on two different datasets. The methodology and solutions used are explained following a sequential order, corresponding also to the code order (code available at colab). As the assignment requires, no neural network model has been used.
The workflow followed is:

1. Data Preprocessing
2. Models selection
    - Tuning
    - Evaluation
    - Comparative Analysis
3. Conclusions

NB: For each model, both datasets are tested and evaluated in parallel and in corresponding sections. Due to high time computation of the tuning process, the best hyperparameters returned are saved. Models are tested more than once and since the splitting process is random, thus there might occur some inaccuracies from the report's results and those on the notebook.

2 Data Preprocessing

The preprocess of both datasets is indeed a crucial aspect to consider. Let's analyize in detail the operations done at this step.

- **Dataset loading**: in order to use both datasets, they need to be downloaded in a proper format.
- **Dataset visualization**: when dealing with huge datasets, it is necessary to plot data so that some considerations can be done before choosing the models.
- **Dataset distributions**: it is convenient to know how data is distributed, so the range of values features take, in order to normalize data, if necessary.

- **Dataset normalization**: when the range of values on which data is distributed is too big, then there might be instability in the training, so it is a good practice normalizing data.
- **Dataset balance**: typically we want the training process not to be affected by the number of samples belonging to one class wrt another. (E.g. if the dataset has few samples for a class, so few information available, then the learning process for that label can be incomplete and can produce some errors in test step).
- **Dataset splitting**: As to perform training, validation and test steps, the separation of the dataset into training and test sets in a random way is done. The validation set will be set in the K-Fold Cross Validation. The randomness guarantees same probability distribution in each set and an unbiased model.

2.1 Datasets

2.1.1 Loading

After the download of the dataset, let's distinguish between Xi and Yi. Xi contains the samples of dataset i and Yi the label associated to each sample.

- **Size of X1**: (50000, 100), so it has 50000 examples and each sample has 100 features.
- **Size of Y1**: (50000,), so, as X, it has 50000 examples and each sample has a label going from 0 to 9.
- **Size of X2**: (50000, 1000), so it has 50000 examples and each sample has 1000 features.
- **Size of Y2**:(50000,), so, as X, it has 50000 examples and each sample has a label going from 0 to 9.

2.1.2 Visualization

As the following figures show, a lot of features (in two random samples) have 0 as value, so they have no contribution in the learning process. In particular, in dataset 1, for this particular sample, there are more than 50 features (out of 100) with zero values, while in dataset 2 there are more than 500 features (out of 1000) with zero values.

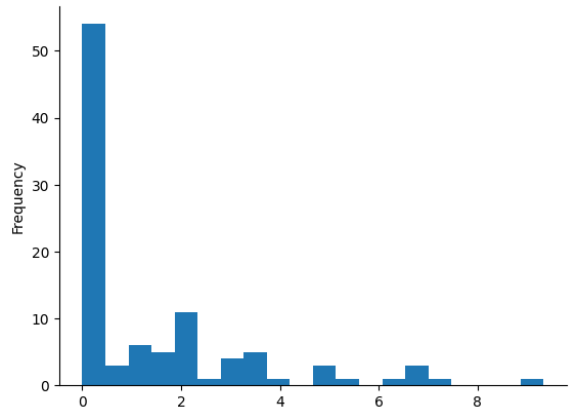

Figure 1: Dataset1

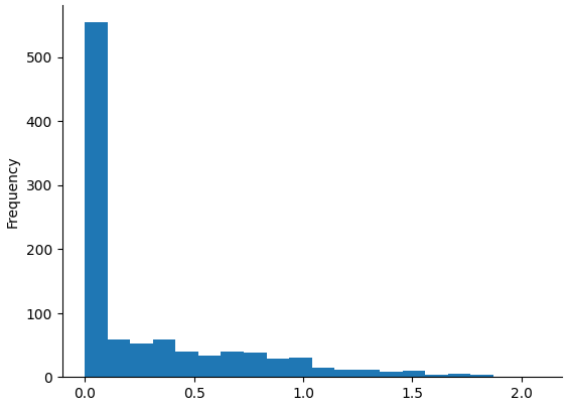

Figure 2: Dataset2

Since both datasets have a huge number of features, PCA is used to reduce dimensionality of both datasets and display how the samples are distributed in 2D-space.

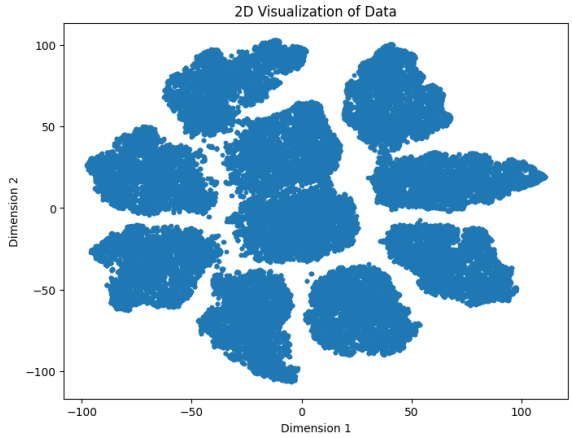

Figure 3: Dataset1

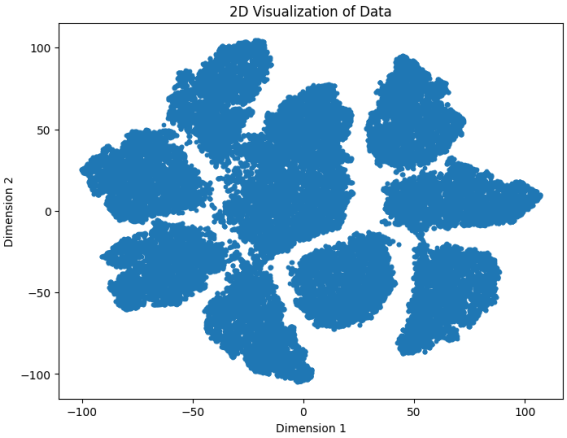

Figure 4: Dataset2

As the figures show, the samples are clustered and this behaviour should be an advantage for classification.

2.1.3 Distributions

For computational and stylistic purposes, the distributions are plotted only for random and well-balance subsets of both dataset.

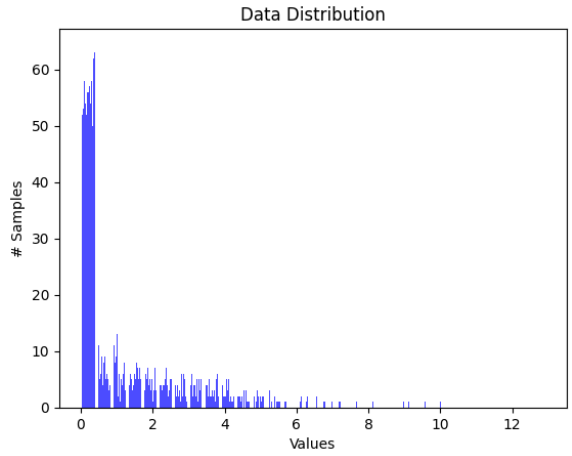

Figure 5: Dataset1

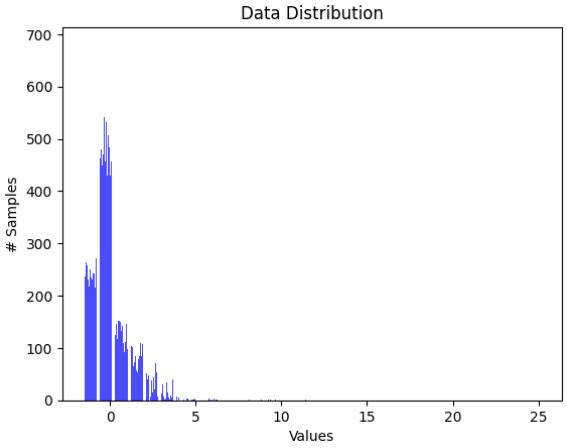

Figure 6: Dataset2

2.1.4 Normalization

As normalization technique, the Standard Scaler has been used, which first calculates the mean and the standard deviation of each feature in the dataset, then it subtracts the mean from each data point in the feature. This centers the data around zero (so the mean of the feature is zero). After subtracting the mean, it divides each data point by the standard deviation. This scales the data so that the standard deviation becomes 1.

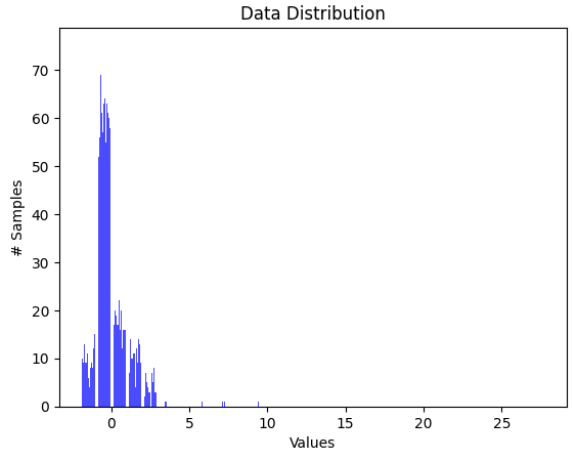

Figure 7: Dataset1

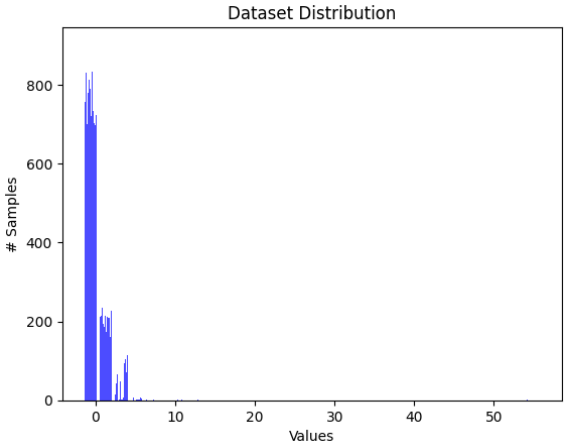

Figure 8: Dataset2

2.1.5 Balance

As the Figures suggest, both datasets are perfectly balanced.

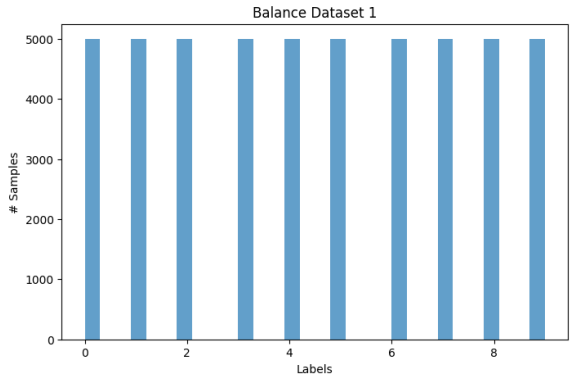

Figure 9: Dataset1

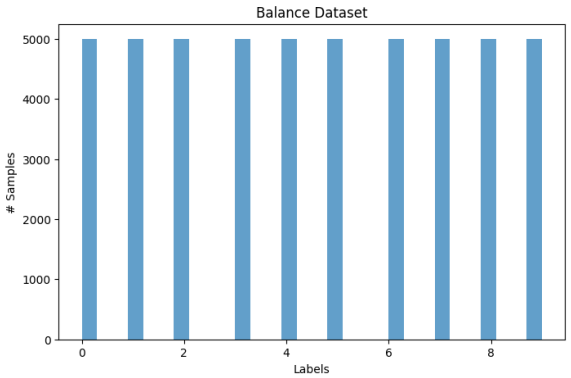

Figure 10: Dataset2

2.1.6 Splitting

Since both datasets have the same size, then the training and test sets too will have same dimensions wrt the number of samples:

- **Training set size**: 40000 samples and 100 features for dataset1, 40000 samples and 1000 features for dataset 2.
- **Test set size**: 10000 samples and 100 features for dataset1, 10000 samples and 1000 features for dataset 2.

As said, the validation set is obtained when doing the K-Fold Cross Validation.

3 Models

To face this multiclass classification problem, 4 models have been used and tested. Here is the list:

1. Decision Tree
2. Logistic Regression
3. Support Vector Machine
4. Ensamble Methods

In order to get the best models, their hyperparameters are tuned. The procedure is implemented with a 5-Fold Cross Validation.
For every hyperparameter a set of possible values is defined and each combination of them is passed to the model. Then, the resulting model is trained on the training set folder and the accuracy on the validation set folder is used to keep the best combination got so far. This "handmade" method allows to control the flow of some metrics while finding the best model, for this reason the classic GridSearchCV is not used.
The hyperparameters leading to the best performance on the validation set are then used to evaluate the model and compute the metrics.
For a multiclass classification problem the metrics used are: **accuracy**, **precision**, **recall**, **f1-score**. In addition, **execution time** and **noise robustness** are analyzed. The robustness to noise is measured by adding random noise labels only in the training set. The model is then trained and tested. The accuracies over different percentages of noise presence are kept and plot.

3.1 Decision Tree

Decision trees are used because of their good properties: they can model complex data, they make no strong assumption about the distribution of data, they perform implicitly features selection. The main drawback of decision trees is the overfitting problem, but it is managed by defining the best values of hyperparameters.

3.1.1 Tuning

As said the selection of hyperparameters is a crucial step. The hyperparameters of decision trees that are tuned are: **max depth** (maximum depth of the tree), **min samples split** (minimum number of samples required to split an internal node), **min samples leaf** (minimum number of samples required to be in a leaf node). Since the overfitting's probability to occur is proportional to the deepness of the tree, in the tuning process described above, it is also defined a variable "patience", set to 3, which is decremented each time the validation accuracy doesn't increase its value wrt the best accuracy got so far. Once patience arrives to zero, the tuning ends.

The set of values for which the model is tested are:

- max depth [1, 15]
- min samples split [2, 5]

- min samples leaf [1, 2]

Let's see how the choice of parameters affects the performances in the graphics below. They represent the courses of the validation accuracy while modelling the best model of decision tree. On the y-axis there is the accuracy, on the x-axis there is the number of combinations tried. The marker "x" in red represents the optimal performance.

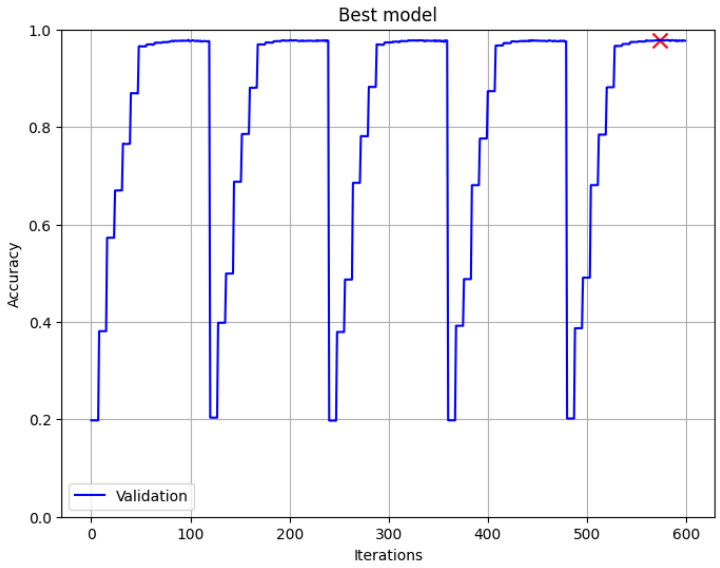

Figure 11: Best decision tree on Dataset 1

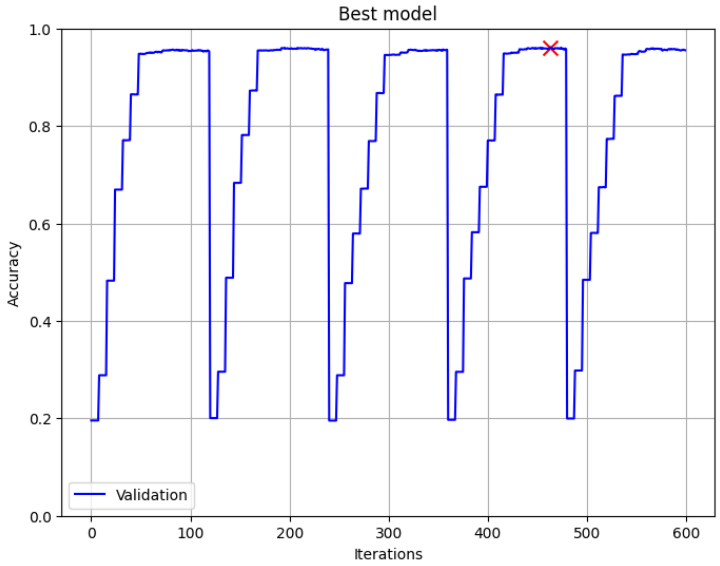

Figure 12: Best decision tree on Dataset 2

For **dataset 1**:

- the best combination of hyperparameters is: **max depth = 12, min samples split = 4, min samples leaf = 1**.
- The optimum value of **validation accuracy** is 0.978 obtained at combination number 573 over the total 600 combinations tried.
- The **time of execution** of the method fit on such model is 3.99 s.

For **dataset 2**:

- the best combination of hyperparameters is: **max depth = 13, min samples split = 5, min samples leaf = 2**.
- The optimum value of **validation accuracy** is 0.961 obtained at combination number 463 over the total 600 combinations tried.
- The **time of execution** of the method fit on such model is 51.33 s.

3.1.2 Evaluation and Metrics

Classification Reports:

Class	Precision	Recall	f1-score
0	0.99	0.98	0.98
1	0.98	0.99	0.99
2	0.98	0.98	0.98
3	0.95	0.94	0.94
4	0.97	0.98	0.98
5	0.95	0.94	0.95
6	0.98	0.99	0.99
7	0.99	0.99	0.99
8	0.99	0.99	0.99
9	0.98	0.98	0.98

Table 1: Classification Report Dataset1

Class	Precision	Recall	f1-score
0	0.98	0.96	0.97
1	0.98	0.98	0.98
2	0.95	0.95	0.95
3	0.89	0.89	0.89
4	0.96	0.96	0.96
5	0.90	0.90	0.90
6	0.98	0.97	0.97
7	0.96	0.98	0.97
8	0.98	0.99	0.99
9	0.98	0.98	0.98

Table 2: Classification Report Dataset 2

	Dataset 1	Dataset 2
Test Accuracy	0.98	0.96

Table 3: Accuracy

Confusion matrix:

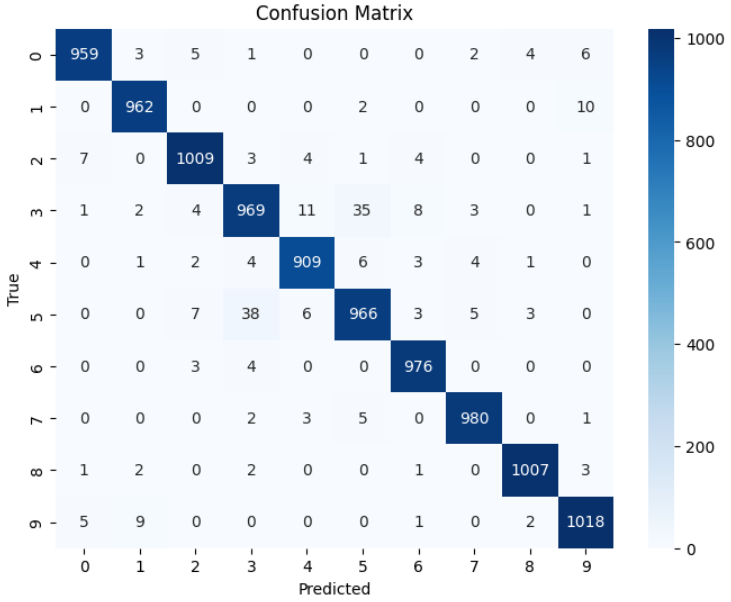

Figure 13: Confusion Matrix Dataset 1

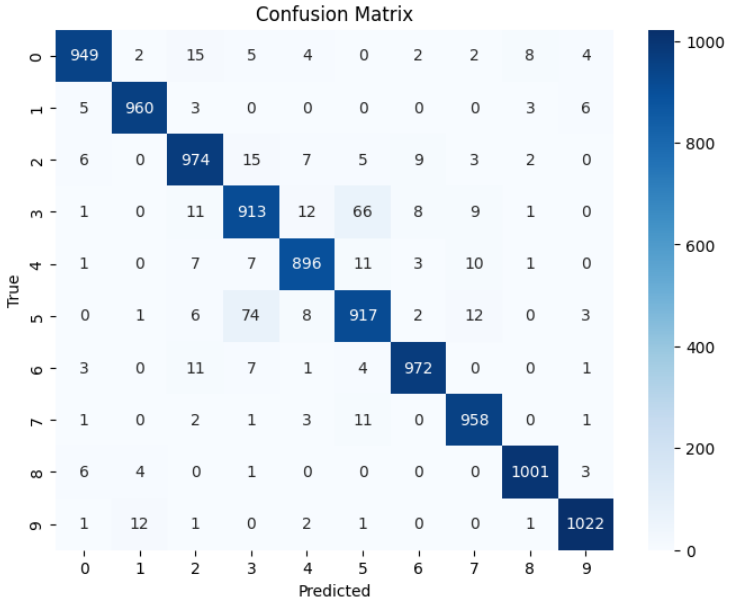

Figure 14: Confusion Matrix Dataset 2

Robustness

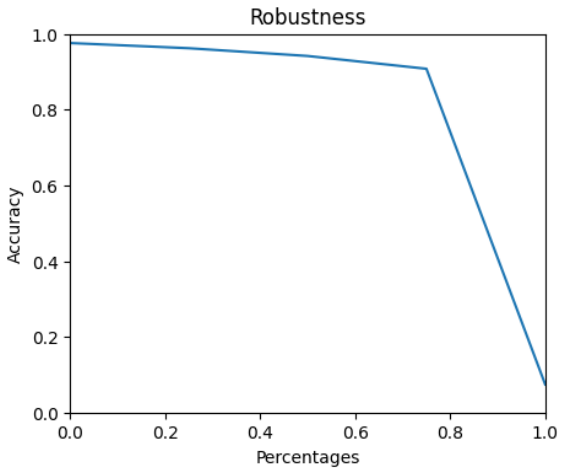

Figure 15: Dataset1

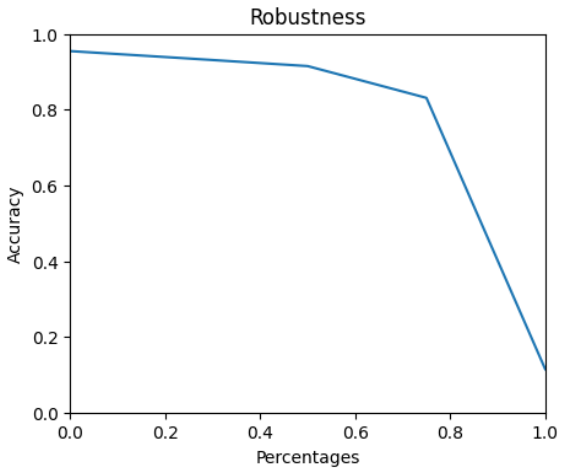

Figure 16: Dataset2

3.1.3 Comparative analysis

The *Figure 11* and *Figure 12* show the curves of the tuning process depending on hyperparameters selection. The five curves, as many as the folders are, have a stair-step shape. The step length is proportional to the number of combinations of min samples split and min samples leaf when max depth is fixed. In this interval the validation accuracy is almost constant. This means that the different values of min samples split and min samples leaf don't affect in a significant way the performance of the model. What makes the accuracy grow in a semi-vertical way is the max depth hyperparameter. In fact, from 0.2 of validation accuracy, corresponding to max depth equal to 1, it is reached 0.978 in the first dataset and 0.961 in the second dataset, when max depth = 12 and max depth = 13 respectively.
Once all the combinations have been tried the best hyperparameters are going to be used by the Decision Tree model.

Then, the model is fit. It is also considered the time of training the model as a metric of evaluation.
The time of execution of the model training for dataset 1 is 3.99 seconds, while the time of execution of the model training for dataset 2 is 51.3 seconds. So the time of execution in the last case is 12 times bigger than the first one. Infact, high-dimensional feature spaces can be computationally intensive and prone to overfitting with decision trees.

After the training of this model on both datasets, the evalutaion process starts. The *Table 1* and *Table 2* report the precision, recall and f1-score. These values are computed based on the label. Precision and Recall are near with each other for all classes, this means that there are more or less the same number of false positives and false negatives. Thus these metrics are good, the smallest precision, recall and f1-score correspond to label 3 and 5 for both datasets. This can be seen also in the *confusion matrices*, where:

- In the dataset 1: the number of samples whose real label is "3" and classified as "5" is 35, while the number of samples whose real label is "5" and classified as "3"

is 38. Their sum is 73 and this contribution represents nearby the 30% of total 245 misclassifications.

- In the dataset 2: the number of samples whose real label is "3" and classified as "5" is 66, while the number of samples whose real label is "5" and classified as "3" is 74. Their sum is 140 and this contribution represents nearby the 32% of total 438 misclassifications.

The *Figure 15* and *Figure 16* show how the performances decreases in terms of accuracy when the noisy increases.
For dataset 1, the decision tree still provides good result up to 75 % of noisy labels, but when all the training set is full of noise (i.e. all labels are flipped randomically), the accuracy goes to a small value.
For dataset 2, the decision tree still provides good result up to 50 % of noisy labels, but when the percentage of noise arrives at 75 %, the accuracy goes to a small value.

3.2 Logistic Regression

The logistic regression is used because of its simple interpretation and efficiency when dealing with big datasets. It can prevent overfitting with regularization term (L1 or L2). Moreover it is robust and stable.

3.2.1 Tuning

The hyperparameters of logistic regression that are tuned are: **Max iter**, the number of iterations necessary for the model to converge, **C**, interpreted as the inverse of regularization term, **Penalty**, the regularization type.
The set of values for which the model is tested are:

- max iter [1,10]
- C [0.01, 0.1, 1.0]
- penalty ['l1', 'l2']

Let's see how the choice of parameters affects the performances in the graphics below. They represent the courses of the validation accuracy while modelling the best model of decision tree. On the y-axis there is the accuracy, on the x-axis there is the number of combinations tried. The marker "x" in red represents the optimal performance.

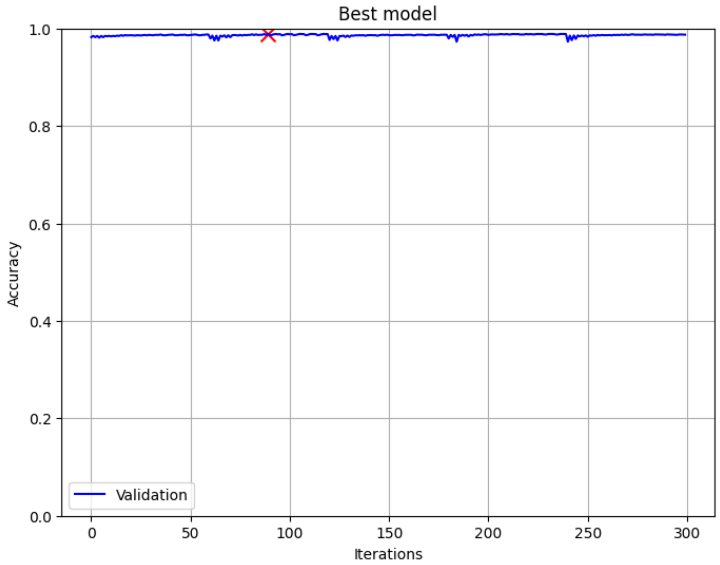

Figure 17: Best logistic regression on Dataset 1

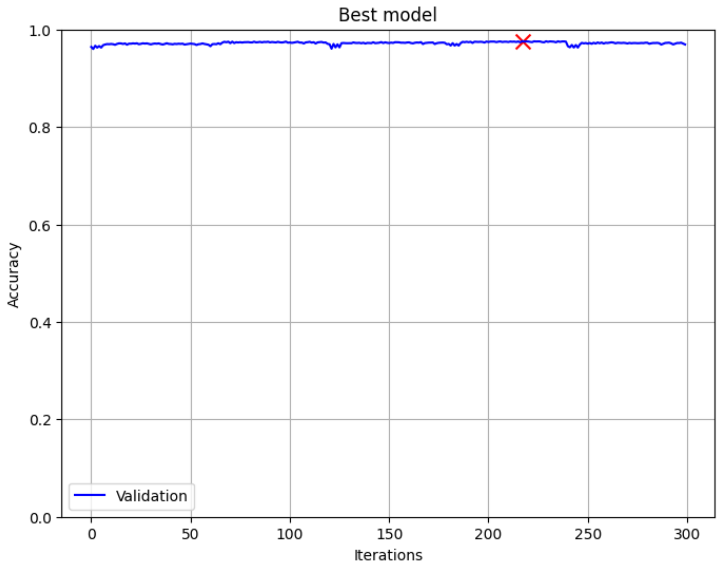

Figure 18: Best logistic regression on Dataset 2

For **dataset 1**:

- the best combination of hyperparameters is: **max iter = 5, C = 1, penalty = l2**.
- The optimum value of **validation accuracy** is 0.988 obtained at combination number 89 over the total 300 combinations tried.
- The **time of execution** of the method fit on such model is 15.40 s.

For **dataset 2**:

- the best combination of hyperparameters is: **max iter = 7, C = 0.01, penalty = l2**.
- The optimum value of **validation accuracy** is 0.976 obtained at combination number 217 over the total 300 combinations tried.
- The **time of execution** of the method fit on such model is 219 s.

3.2.2 Evaluation and Metrics

Classification Reports:

Class	Precision	Recall	f1-score
0	0.99	0.99	0.99
1	0.99	0.99	0.99
2	0.99	0.99	0.99
3	0.96	0.96	0.96
4	0.99	0.98	0.99
5	0.96	0.96	0.96
6	0.99	0.99	0.99
7	0.99	0.99	0.99
8	1.00	1.00	1.00
9	1.00	1.00	1.00

Table 4: Classification Report Dataset1

Class	Precision	Recall	f1-score
0	0.99	0.98	0.98
1	0.98	0.99	0.99
2	0.96	0.98	0.97
3	0.91	0.93	0.92
4	0.98	0.96	0.97
5	0.93	0.92	0.92
6	0.99	0.99	0.99
7	0.98	0.98	0.98
8	0.99	0.99	0.99
9	0.99	0.99	0.99

Table 5: Classification Report Dataset 2

	Dataset 1	Dataset 2
Test Accuracy	0.99	0.97

Table 6: Accuracy

Confusion matrix:

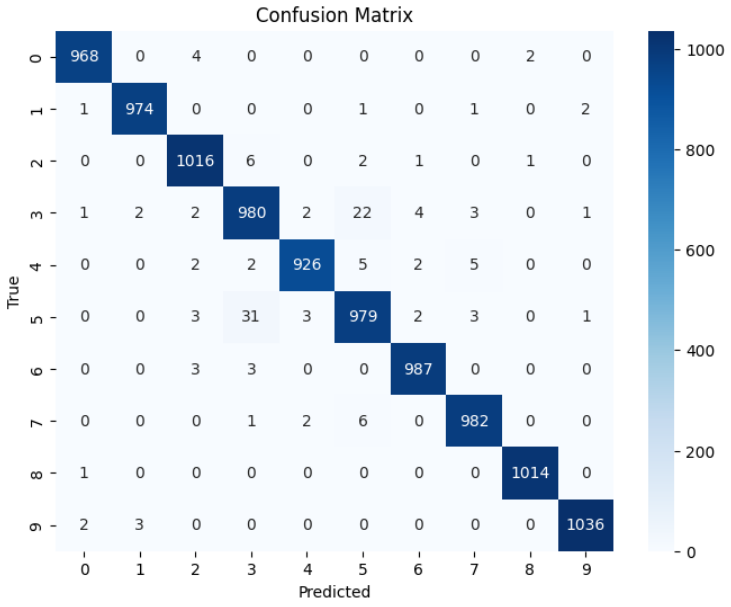

Figure 19: Confusion Matrix Dataset 1

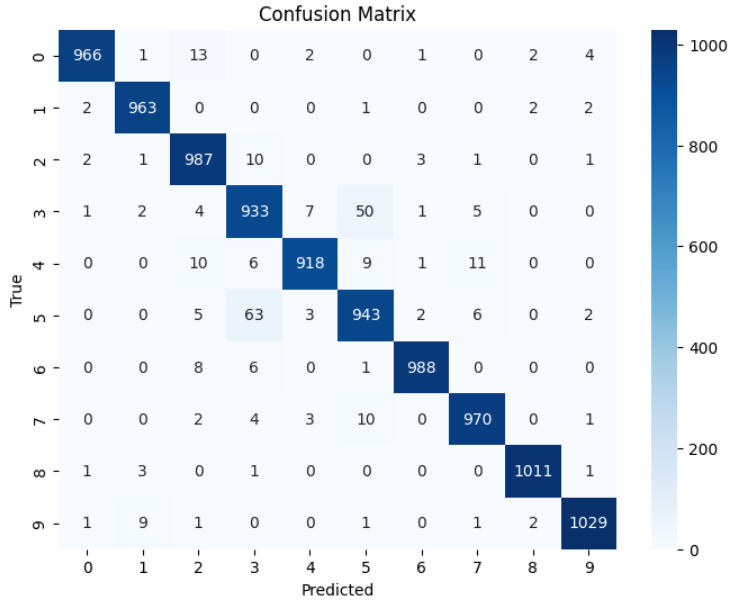

Figure 20: Confusion Matrix Dataset 2

Robustness

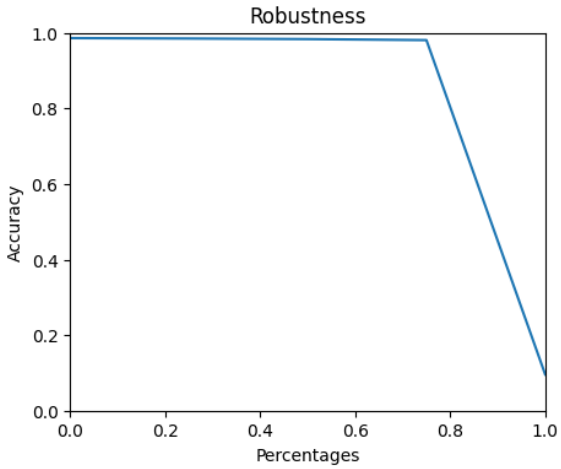

Figure 21: Dataset1

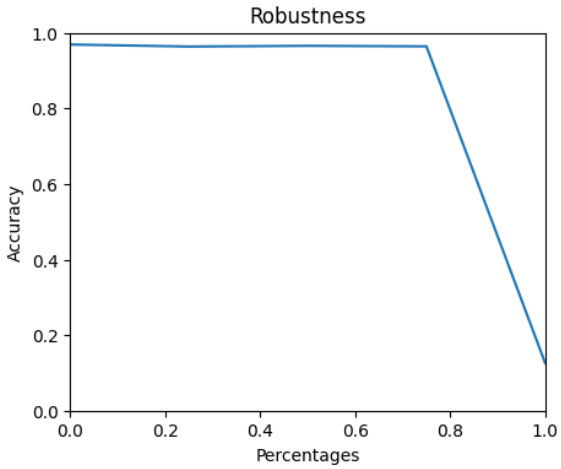

Figure 22: Dataset2

3.2.3 Comparative analysis

The *Figure 17* and *Figure 18* show the curves of the tuning process depending on hyperparameters selection. The five curves, as many as the folders are, are quite stable over good values of validation accuracy reaching in few number of combinations (89) 0.988 of accuracy for dataset 1, and in 213 combinations 0.976 of accuracy for dataset 2.
Once all the combinations have been tried the best hyperparameters are going to be used by the Logistic Regression model.

Then, the model is fit. It is also considered the time of training the model as a metric of evaluation.

The time of execution of the model training for dataset 1 is 15.40 seconds, while the time of execution of the model training for dataset 2 is 219.13 seconds. So the time of execution in the last case is 14 times bigger than the first one.

After the training of this model on both datasets, the evalutaion process starts. The *Table 4* and *Table 5* report the precision, recall and f1-score. These values are computed based on the label. Precision and Recall are near with each other for all classes, this means that there are more or less the same number of false positives and false negatives. Thus these metrics are good, the smallest precision, recall and f1-score correspond to label 3 and 5 for both datasets. This can be seen also in the *confusion matrices*, where:

- In the dataset 1: the number of samples whose real label is "3" and classified as "5" is 22, while the number of samples whose real label is "5" and classified as "3" is 31. Their sum is 53 and this contribution represents nearby the 38% of total 138 misclassifications.
- In the dataset 2: the number of samples whose real label is "3" and classified as "5" is 50, while the number of samples whose real label is "5" and classified as "3" is 63. Their sum is 113 and this contribution represents nearby the 38% of total 292 misclassifications.

This means that, given the accuracies on both dataset, the error in classifying those 2 features affect in a significant way the total error.
The *Figure 21* and *Figure 22* show how the performances decreases in terms of accuracy when the noisy increases.
For dataset 1, the logistic regression still provides very good result up to 75 % of noisy labels and the accuracy seems not to decrease when increasing noise, but when all the training set is full of noise (i.e. all labels are flipped randomically), the accuracy goes to a small value.
For dataset 2, the logistic regression behaves as in the first case.

3.3 Support Vector Machine

SVM models are known for their ability to: perform well in high-dimensional features spaces, be robust in presence of outliers, to have high accuracy.

3.3.1 Tuning

The hyperparameters of Support Vector Machine that are tuned are: **kernel** and the **degree of the polynomial**. The set of values for which the model is tested are:

- kernel [linear, poly]
- degree [1,2,3]

Let's see how the choice of parameters affects the performances in the graphics below. They represent the courses of the validation accuracy while modelling the best model of decision tree. On the y-axis there is the accuracy, on the x-axis there is the number of combinations tried. The marker "x" in red represents the optimal performance.

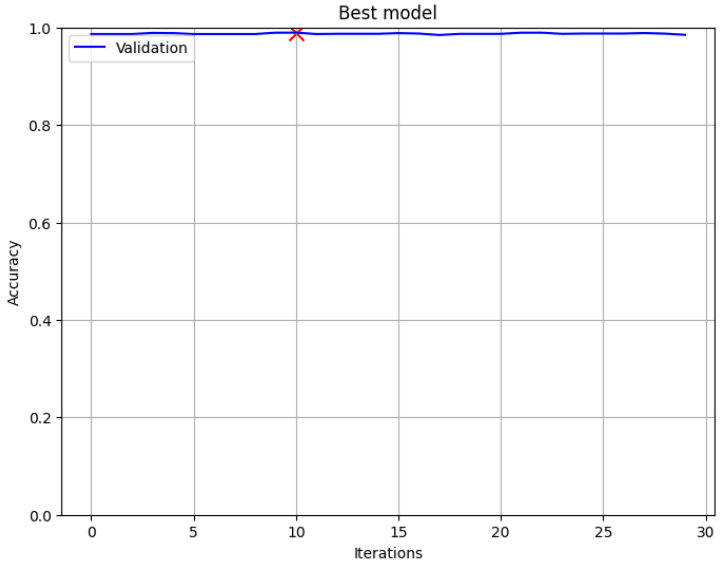

Figure 23: Best svm on Dataset 1

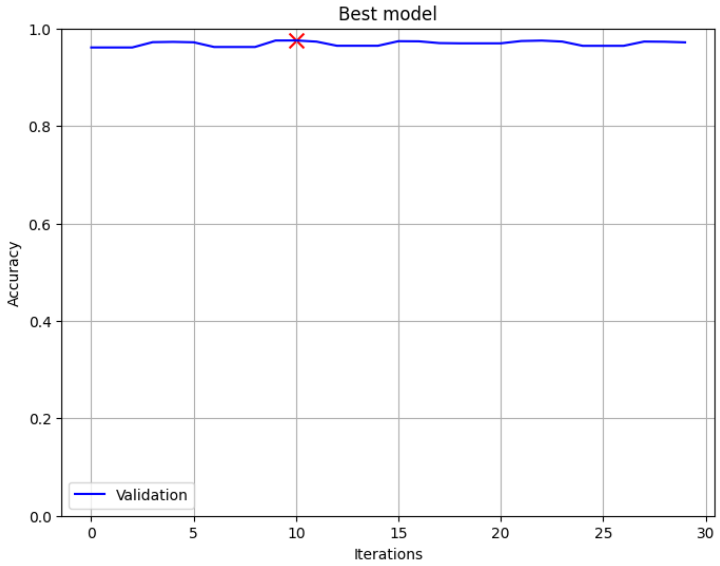

Figure 24: Best svm on Dataset 2

For **dataset 1**:

- the best combination of hyperparameters is: **kernel = polynomial, degree = 2**.
- The optimum value of **validation accuracy** is 0.989 obtained at combination number 10 over the total 30 combinations tried.
- The **time of execution** of the method fit on such model is 8.89 s.

For **dataset 2**:

- the best combination of hyperparameters is: **kernel = polynomial, degree = 2**.
- The optimum value of **validation accuracy** is 0.975 obtained at combination number 10 over the total 30 combinations tried.

- The **time of execution** of the method fit on such model is 69.9 s.

3.3.2 Evaluation and Metrics

Classification Reports:

Class	Precision	Recall	f1-score
0	0.99	0.99	0.99
1	0.99	1.00	1.00
2	0.99	1.00	0.99
3	0.96	0.95	0.96
4	0.99	0.99	0.99
5	0.97	0.96	0.96
6	0.99	1.00	0.99
7	0.99	0.99	0.99
8	1.00	1.00	1.00
9	0.99	0.99	0.99

Table 7: Classification Report Dataset1

Class	Precision	Recall	f1-score
0	0.99	0.98	0.99
1	0.99	0.99	0.99
2	0.96	0.98	0.97
3	0.92	0.92	0.92
4	0.98	0.96	0.97
5	0.92	0.93	0.93
6	0.99	0.99	0.99
7	0.98	0.98	0.98
8	1.00	0.99	0.99
9	0.99	0.99	0.99

Table 8: Classification Report Dataset 2

	Dataset 1	Dataset 2
Test Accuracy	0.99	0.97

Table 9: Accuracy

Confusion matrix:

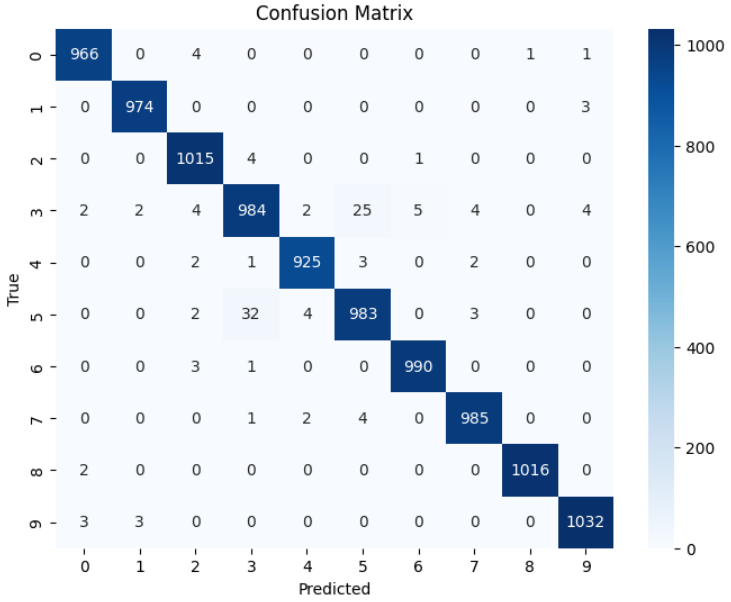

Figure 25: Confusion Matrix Dataset 1

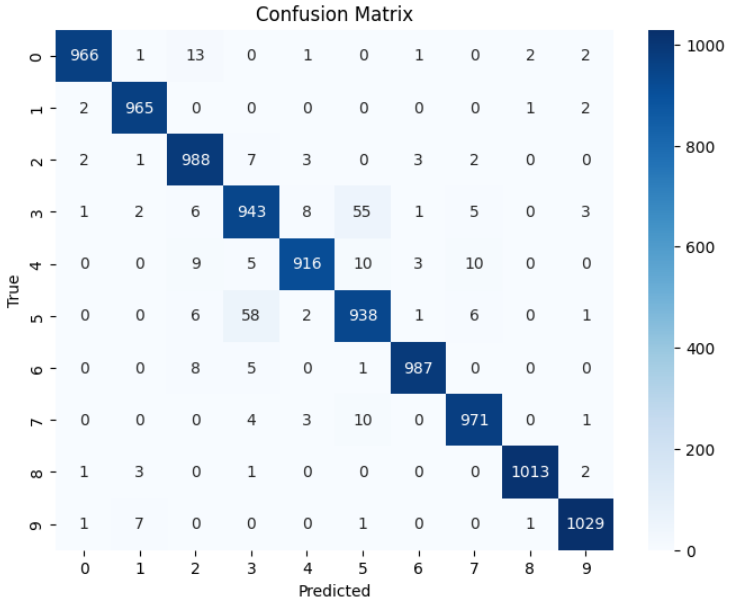

Figure 26: Confusion Matrix Dataset 2

Robustness

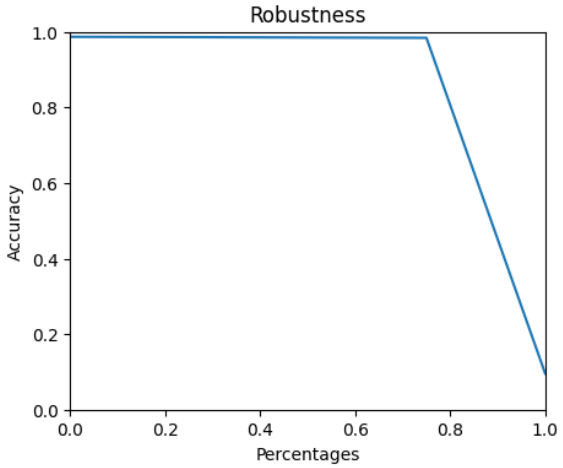

Figure 27: Dataset1

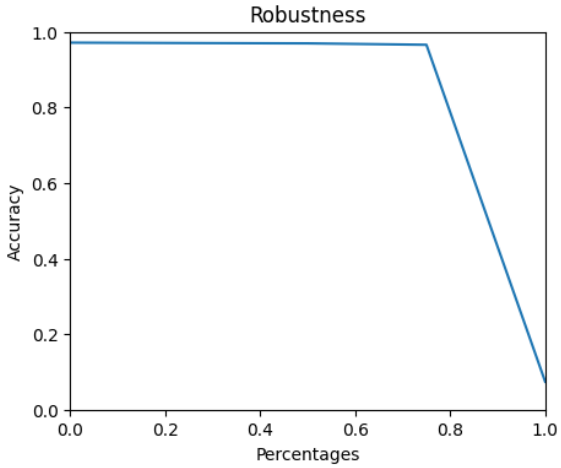

Figure 28: Dataset2

3.3.3 Comparative analysis

The *Figure 23* and *Figure 24* show the curves of the tuning process depending on hyperparameters selection. The five curves, as many as the folders are, are quite stable over good values of validation accuracy reaching in few number of combinations (10) 0.975 of accuracy for dataset 1, and in 10 combinations 0.976 of accuracy for dataset 2. The validation accuracy when the kernel is "linear" seems to be constant for each degree tried. It increases in a (almost) linear way when the kernel is switched to polynomial.
Once all the combinations have been tried the best hyperparameters are going to be used by the Support Vector Machine model.

Then, the model is fit. It is also considered the time of training the model as a metric of evaluation.
The time of execution of the model training for dataset 1 is 8.89 seconds, while the time of execution of the model training for dataset 2 is 69.91 seconds. So the time of execution in the last case is 8 times bigger than the first one.

After the training of this model on both datasets, the evalutaion process starts. The *Table 7* and *Table 8* report the precision, recall and f1-score. These values are computed based on the label. Precision and Recall are near with each other for all classes, this means that there are more or less the same number of false positives and false negatives. Thus these metrics are good, the smallest precision, recall and f1-score correspond to label 3 and 5 for both datasets. This can be seen also in the *confusion matrices*, where:

- In the dataset 1: the number of samples whose real label is "3" and classified as "5" is 22, while the number of samples whose real label is "5" and classified as "3" is 32. Their sum is 54 and this contribution represents nearby the 41,5% of total 130 misclassifications.
- In the dataset 2: the number of samples whose real label is "3" and classified as "5" is 55, while the number of samples whose real label is "5" and classified as "3" is

58. Their sum is 113 and this contribution represents nearby the 40% of total 284 misclassifications.

This means that, given the accuracies on both dataset, the error in classifying those 2 features affect in a significant way the total error. The *Figure 27* and *Figure 28* show how the performances decreases in terms of accuracy when the noisy increases.
For dataset 1, the svm still provides good results up to 75 % of noisy labels, and accuracy remains stable, but when all the training set is full of noise (i.e. all labels are flipped randomically), the accuracy goes to a small value.
For dataset 2, the same considerations of the first case can be done.

3.4 Ensamble methods

Ensamble methods are widely used because they increase accuracy and combine different learners properties and advantages. There are many types of ensamble methods, but in this case it is used the hard-voting ensamble methods: each individual classifier in the ensemble predicts a class label, and the class that receives the most votes is the final prediction. The models which will be used are: Decision Trees, Logistic Regression and SVM.

3.4.1 Tuning

The tuning process here is not done since the best hyperparameters of the models used in the ensamble method are already obtained.

3.4.2 Evaluation and Metrics

This last technique is tested both when the dataset is normalized and when not so that it is possible to see differences. However, these models work fine with or without normalization and the value of accuracy and time of execution is nearby the same.
Classification Reports:

Class	Precision	Recall	f1-score
0	0.99	0.99	0.99
1	0.99	1.00	0.99
2	0.99	1.00	0.99
3	0.96	0.95	0.96
4	0.99	0.99	0.99
5	0.97	0.96	0.97
6	0.99	1.00	0.99
7	0.99	0.99	0.99
8	1.00	1.00	1.00
9	0.99	0.99	0.99

Table 10: Classification Report Dataset1

Class	Precision	Recall	f1-score
0	0.99	0.97	0.97
1	0.99	0.99	0.99
2	0.96	0.98	0.97
3	0.92	0.92	0.92
4	0.98	0.96	0.97
5	0.93	0.92	0.92
6	0.99	0.99	0.99
7	0.98	0.99	0.99
8	0.99	0.99	0.99
9	0.99	0.99	0.99

Table 11: Classification Report Dataset 2

	Dataset 1	Dataset 2
Test Accuracy	0.99	0.97

Table 12: Accuracy

Confusion matrix:

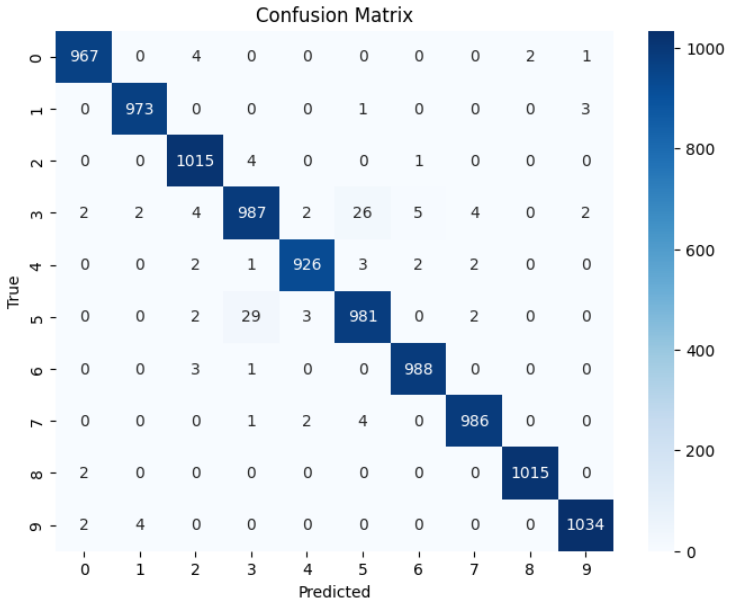

Figure 29: Confusion Matrix Dataset 1

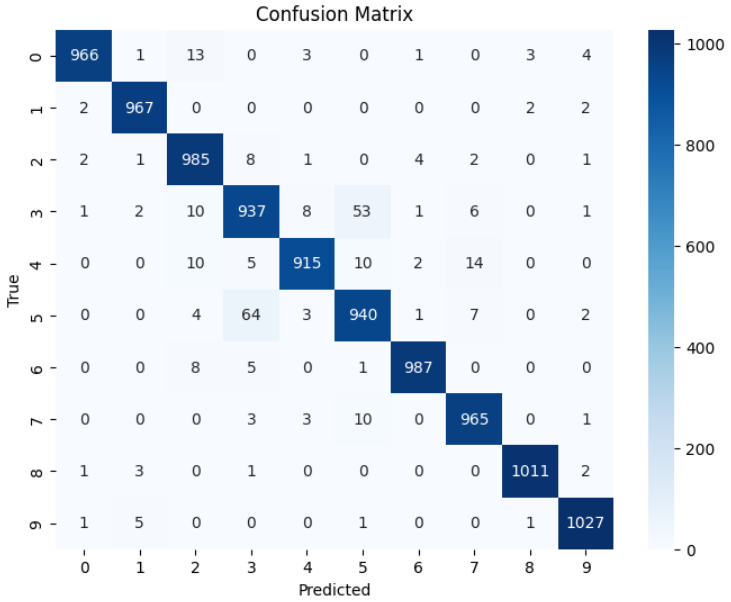

Figure 30: Confusion Matrix Dataset 2

3.4.3 Comparative analysis

As can be guessed, the results are the best obtained so far in terms of accuracy, precision, recall and f1-score.

In the first dataset:

- The time of execution of the ensamble method fit is 12.6 seconds
- The total number of misclassification is 128: 55 errors occur when classifying the samples labeled with 3 (and classified as 5) and 5 (classified as 3), so 43% of total error.

In the second dataset:

- The time of execution of the ensamble method fit is 143 seconds.
- The total number of misclassification is 300: 117 errors occur when classifying the samples labeled with 3 (and classified as 5) and 5 (classified as 3), so 39% of total error.

4 Conclusions

After having seen the methods above, we can draw some conclusions.

For dataset 1:

- The **quickest method** is Decision Trees model
- The **highest accuracy** is obtained with Logistic Regression and SVM
- The **most robust methods** are SVM and Logistic Regression.

For dataset 2:

- The **quickest method** is Decision Trees model
- The **highest accuracy** is obtained with Logistic Regression and SVM
- The **most robust method** are SVM and Logistic Regression.

5 Blind Tests

In addition to this part, there are also reported in files **"d1_1892079.csv"** and **"d2_1892079.csv"** the predictions on the blind tests for dataset 1 and dataset 2, respectively.
In order to predict the labels on these two test sets, the last technique of Ensable Methods with the best hyperparameters of each model is used.
The training process is done on the training sets of the starting datasets and then the test is done on the blind sets.
The predictions are kept and saved in the csv files.