Data Mining - Assignment 1

(CSC 503)

François Lédée - V00956813

June 26, 2020

1 Exercise 1: Experiments and Analysis

1.1 Introduction

In this first part, the three following methods are implemented: Decision tree, Random forest, Neural Network. The implementation consists in the use of the existing library for machine learning *Scikit-learn*. The analysis are realized with Python3.7. Each method is investigated on the basis of two binary classification exercises. The first data set is the *heart disease* dataset from UCI repository[1]. The second is the *digits* set available in *scikit-learn 0.23.1*. (When choosing this dataset, I did not know this would be the dataset used in the next TA...)

The exercises consist in binary classifications. Therefore, a prediction can be right or wrong labeled. The distinguish between the false negatives and the false positives lies out of the scope of this work. In this paper, the only focus is made on whether the sample is good labeled or not. Consequently, the metric used to assess the score of the model is the ***accuracy*** (portion of good labeled samples among the whole classified data).

In each part, the construction and results of each model are presented, then a fourth subsection compares the results and formulate an explicit choice.

1.2 Part 1: Dataset from UCI on heart decease

The first dataset investigated is the *heart disease* dataset from UCI repository containing 297 individuals of 13 features. This dataset has already been transformed to best suit the machine learning exercise. A quick analysis reveals the presence of tree binary variables (fields 2, 6 and 9), five categorical variables (fields 3,7,11,12 and 13) and five numerical variables with various ranges in three different order of magnitude.

The decision tree and random forest models treat features separately, the scaling of the data is not supposed to affect their performance or the way they are built. This is not the case of the neural network, therefore, the effect of the standardization on its performances will be explored in that third model only.

1.2.1 Decision trees

A decision tree is a method consisting in a sequences of choices made based on the characteristics of an individual. The training process consists in choosing the best feature to split the data on at each step of the implementation. Such a model can rapidly become specific to the data used for its training, therefore a *pruning* process can be used to shorten the tree and avoid so called *overfitting*. The most instinctive pruning method is the *reduced error pruning*: the tree is first grown as it maximum to maximize its accuracy when applied on the training data, then it is shorten by cutting off some of its leaves to maximize its accuracy when applied on *validation data*.

The *Scikit-learn* library proposed most advanced methods for the pruning, therefore the *reduced error pruning* method has been implemented and is available in the file ***pruning.py***. Figure 1 represent the effect of this pruning algorithm on the present dataset.

The evolution of the accuracy during the two phases of the training is represented in Figure 2. During the first phase (grown of the tree), the accuracy on the training set (blue) and validation set (red) move apart. The pruning (pink) tend to increase the accuracy again. The horizontal portion of the line (between 22 and 31 leaves), and more generally its stairs shaped structure is due to the suppression of nodes where none of the validation points are falling. These nodes therefore appeared useless to the algorithm during the pruning.

The methodology to apply to the decision tree is straightforward when the pruning is to be run from a pure tree, therefore, only the criterion related to the information gain (entropy or Gini

[1] https://archive.ics.uci.edu/ml/datasets/Heart+Disease

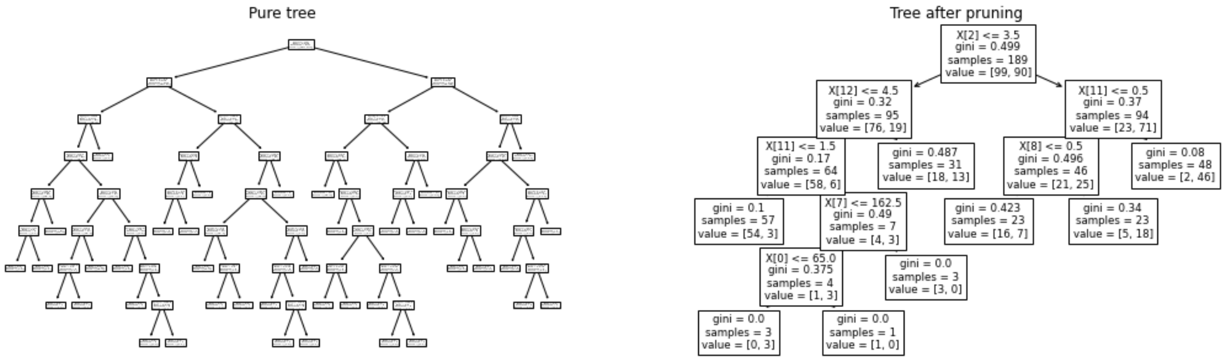


Figure 1: The pruning shorten the initial tree to prevent the model to overfit. The right tree is the pure tree, when grown at its maximum. The left tree is the pruned one to maximize the accuracy on a validation set.

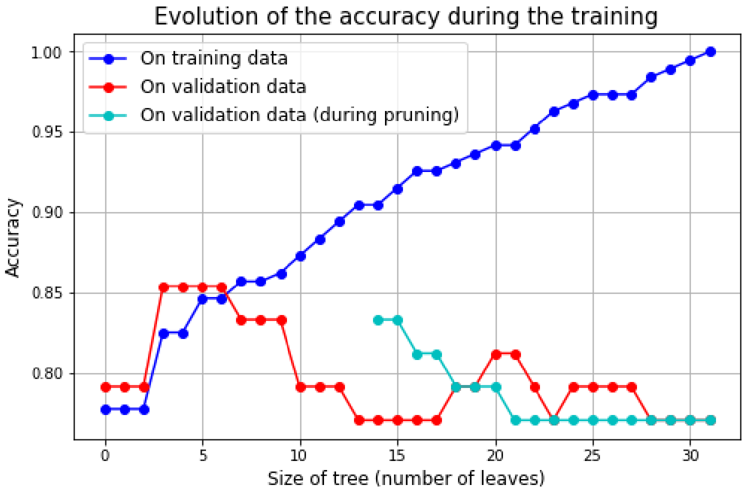


Figure 2: Evolution of the accuracy of the model during the two phases of it's training.

index) is explored. The Figures 3 and 4 show the two trees at the pure and the final stages. The first noticed difference is the homogeneous shape of the pure tree reached with the Gini index. However the first separation criteria and threshold are the same. Consequently, both information criteria lead to trees more pruned on their right side (Figure 4), probably due to a non-perfect homogeneity of the validation sample.

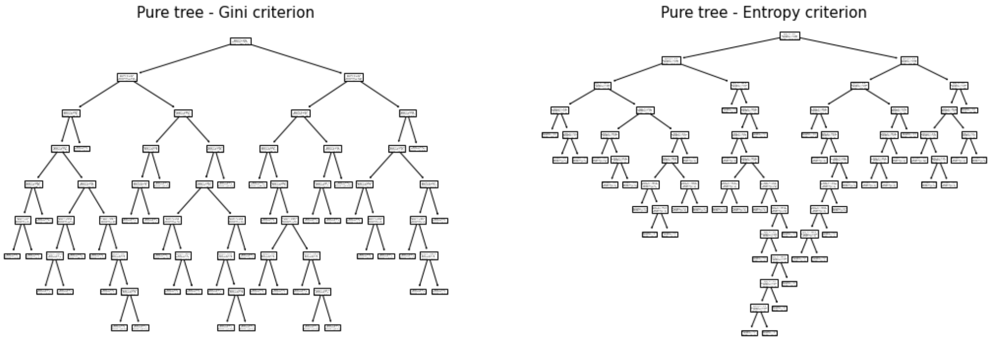


Figure 3: Pure trees before pruning, grown based on the Gini index (left) and the Entropy (right) information criteria.

The accuracy reached with the *entropy* information criterion leads to slightly better final results (76.6% with the entropy, against 73% with the Gini criterion), so further analysis is led with the entropy criterion.

To train the model and validate it, the initial dataset is divided into two parts: the training set and the testing set. Figure 5a represents the accuracy of the model for various portions of testing set. The more testing data, the less training data. Consequently the accuracy slightly decrease with increasing portion of testing set. This accuracy drops for very high portions of training set, as there is not enough information left to properly grow the tree at first.

Data is not always available. Moreover training with less data can lead to better generalization (even if it is rarely the case). Figure 5b shows that the decision tree performs poorly with very few data but stabilizes rapidly its accuracy with only half of the dataset. Figure 6 represents the trees for three different sizes of used dataset.

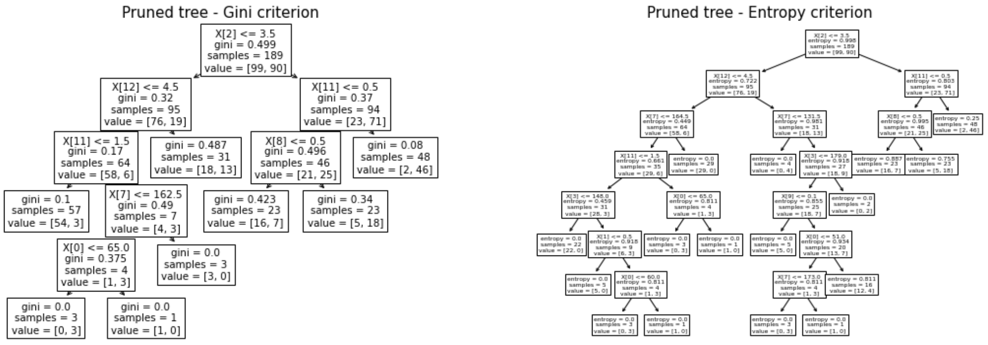
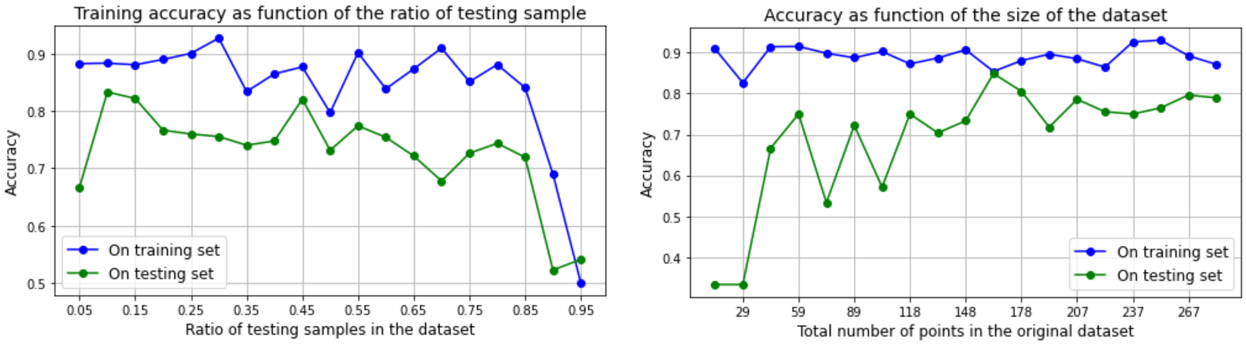


Figure 4: Pruned trees, grown based on the Gini index (left) and the Entropy (right) information criteria.



(a) Ratio training-testing

(b) Total quantity of available data

Figure 5: Effect of the number of ratio of testing samples 5a and of the quantity of samples available 5b on the accuracy of a decision tree model.

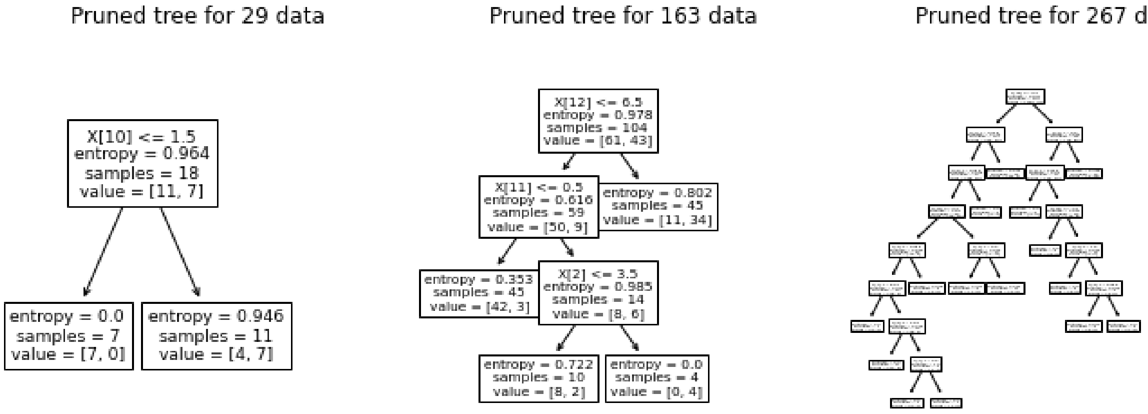


Figure 6: Visual representation of the pruned tree for tree different number of available data points

The decision tree is a very visual method, however the number of parameters is limited. The best accuracy reached by the model is 76.6% of accuracy on the testing set with the full use of the dataset and a portion of 80% of training data.

1.2.2 Random Forest

The random forest method consists in building numerous of decision trees and consider as a result the average prediction of each tree. To avoid having systematically identical trees in the forest, stochasticity is involved twice in the process. First, each tree is build based on a randomly selected subset of the original training dataset. Secondly, to build each node of a tree, only some random selected features are selected to be potential candidates to find the best split. Finally, the trees in a forest are not necessarily pure, therefore the number of trees in the forest, as well as a limitation of the trees depth can be explored.

In this study, the *Gini impurity* measure is selected, as an implementation with the default parameters proposed in Scikit-learn reveals an *accuracy of* 81.7% against 76.7% reached with the entropy criterion on the testing dataset.

Figure 7 shows the evolution of the accuracy for an increasing maximal number of leaves and an increasing maximal depth of each tree. It first reveals the equivalence between the two criteria. A possible explanation is the choice of the Gini impurity measure leading to well balanced trees, as shown above in Figure 3. The depth limitation by the number of leaves is therefore more detailed and it appears the best performing model limits its trees to 8 leaf nodes.

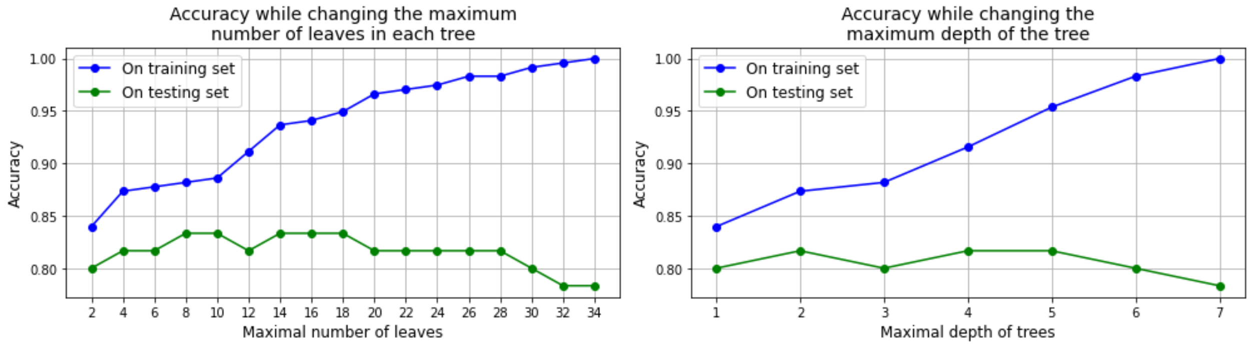
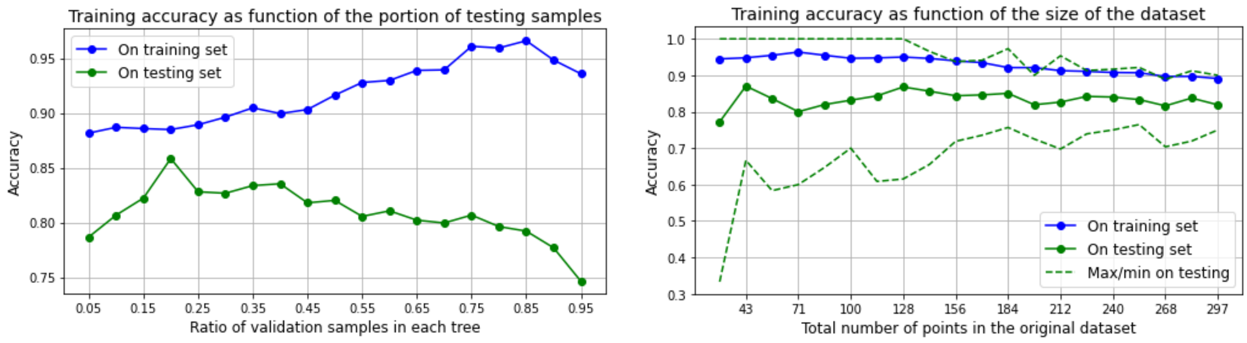


Figure 7: Evolution of the accuracy with the maximal authorized number of leaves per tree (left) and maximal authorized depth (right) per tree.



(a) Effect of the testing ratio on the accuracy of the random forest

(b) Accuracy of the model when only a portion of the initial dataset is considered

Figure 8: Trends for the evolution of accuracy

The strength of a random forest lays in the consideration of lots of individuals, therefore the performances are relatively stable. The number of random selected samples at each tree (Figure 27a in appendix) and the maximum number of trees in the forest (Figure 27b in appendix) have no particular effect on the performances. Therefore the modeling choice is made on non-significant variations during the process. For the first criteria, 40% of the available samples is considered while a number of 87 trees per forest is chosen for the second.

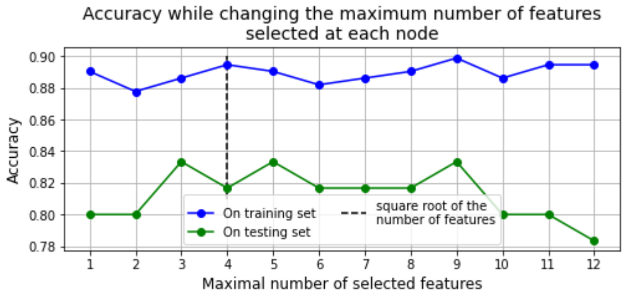


Figure 9: Accuracy for increasing number of features randomly selected to candidate for the best split at each node of each tree in the forest.

Regarding the number of features randomly selected at each node, a very light trend can be observed (Figure 9, with a decrease for very small or large choices. A shortlisting of 5 features leads to better results.

Figure 8 is directly related to the data used to train the model. The optimal split of data (left) is 80% of training samples for 20% of testing samples.

Ran 30 different times with all the above exposed parameters, the use of different portions of the initial dataset reveals interesting trends (right of Figure 8). First the uncertainty of the accuracy (on the testing set) slightly decreases when more data are used. However, a very light curving in the average result shows a lightly better accuracy when around one third of the dataset is used. This observation might however be biased: as a famous president recently said, the less we test and the fewer we get wrong answers.

n estimator	87
max features	5
max samples	40%
max leaf nodes	8
criterion	Gini

Table 1: Parameters of the best achieved random forest for the *heart decease* dataset

The model seems to perform very well with only few data, and is robust in its results. The random state is set to zero , to enable reproducibility. The best investigated model takes the parameters exposed in the Table 1 and achieves an accuracy of 83.3% on the testing set (89.8% on the training set) with 13.3% of false positive (predicted positive while actually negative) and only 3.3% of false negative.

1.2.3 Neural Network

The neural network is a complex model constituted of multiple perceptrons connected together. In this subsection, the training data are standardized. A simple test ran with the default parameters proposed by *Scikit-learn* reveals an accuracy around 80% when the *heat-decease* dataset is standardized against around 60% with the original one.

To explore the parameters in terms of order of magnitude, a first *grid search* is ran with various parameters (Table 2). To simplify, only models with one unique hidden layer will be investigated. As the grid search is based on the *cross-validation* method (realized here with 5 folds), the test is run with the whole dataset.

This technique reveals a best accuracy of 85.1%. However, it is first interesting to notice that the best 5% of the tested models (28 in total) manage an accuracy greater than 84.1%, and thus with all possible values tested for each parameter.

Tested parameters
activation function
L2 regularization term
number of neurons on the hidden layer
learning rate
method to decrease the learning rate
solver

Table 2: Parameters tested on the neural network model

The results obtained regarding the categorical parameters tested is exposed in Figure 10. In terms of solver, the best results are achieved by the *adam*, what is an improved version of the stochastic gradient descent (sgd). This first solver computes significantly faster than the *sgd* algorithm and also seems to bring stronger guarantees of a good accuracy.

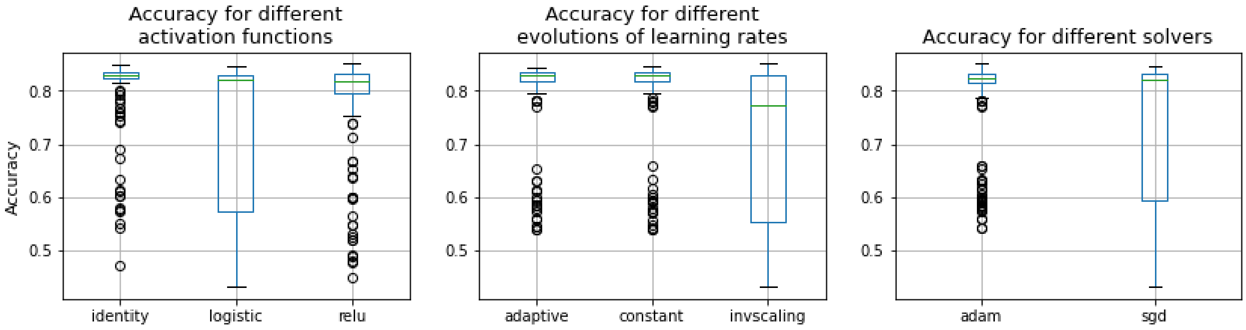


Figure 10: Distribution of the grid search results for the categorical parameters

This guarantee is also observed in the way of making the learning rate evolve, with the *adaptive* and the *constant* methods. However, the four best models issued from the grid search are using the *invscaling* method. The constant method results in a constant learning rate, while the adaptive divides the learning rate by 5 if two consecutive epochs failed at decreasing the learning loss by a certain value. The *invscaling* method decreases regularly the learning rate. An example of learning curve is exposed in Figure 11.

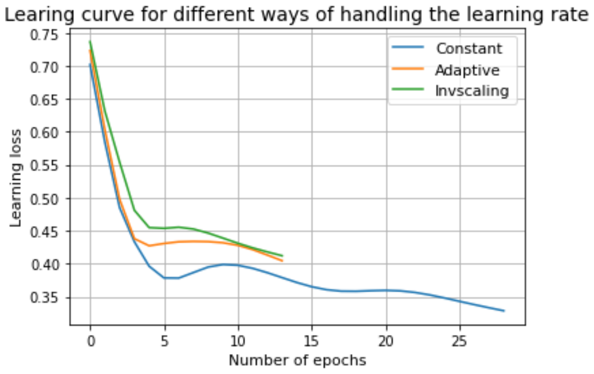
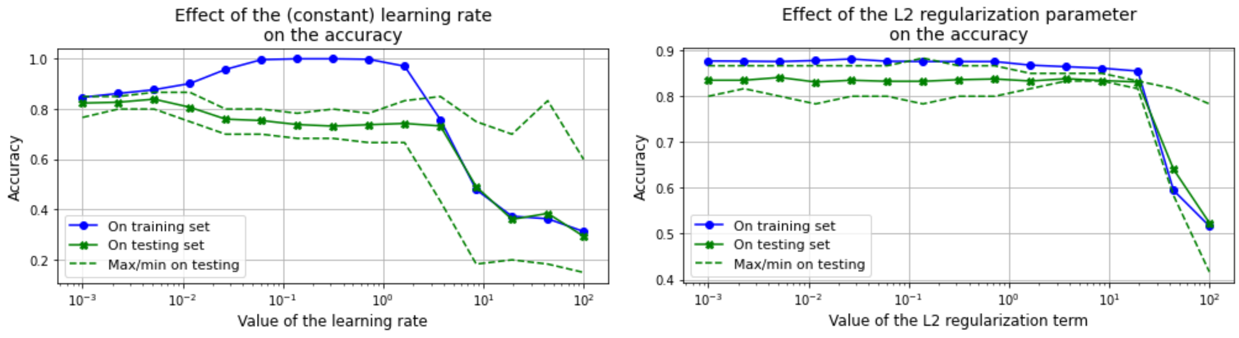


Figure 11: Learning curve of tree networks using different handling methods of the learning rate

A (fully connected) neural network can be adjusted in various different ways as it also comprises various numerical parameters. The three most important are the number of hidden layers with the number on neurons on each, the learning rate and the L2 regularization parameter. Figure 12a shows for 30 trials that testing accuracy decreases as the learning rate increases, but also that the variability of the model's result increases. It also reveals a breaking point at around 5 in the performances, meaning that the optimization steps are too large for the algorithm to find an optimum. Such a point also exists for the L2 regularization parameter, as shown in Figure 12b. This term seems otherwise not to have any effect on the performances.

The results are similar while studying the number of neurons on the unique hidden layer (Figure 13 left). First, the testing results are quite uncertain (vary between 79% and 93% for a same set of parameters). The experiment surprisingly highlights a testing accuracy greater than a training accuracy. Finally, similar accuracy are achieved for 20 neurons and above.



(a) Learning rate

(b) L2 regularization term

Figure 12: Average evolution of the accuracy of a neural network with its learning rate (left) and its L2 regularization term (right). Computed with 30 independent trials.

The convergence of the model seems a priori to be required, however forcing it leads to a clear overfitting and poor testing performances for all numbers of neurons (Figure 13 right).

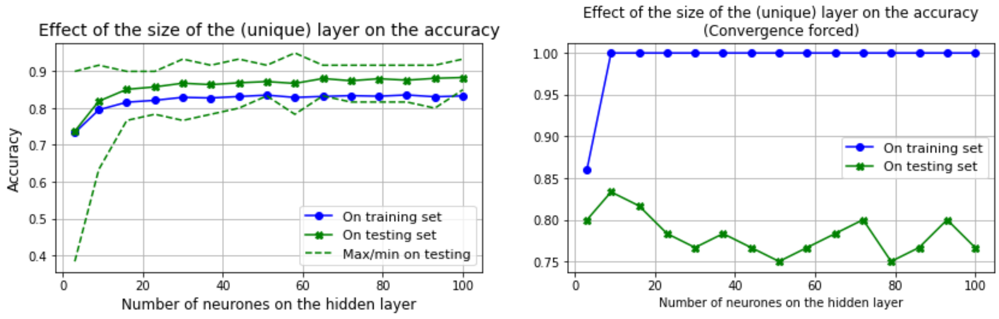
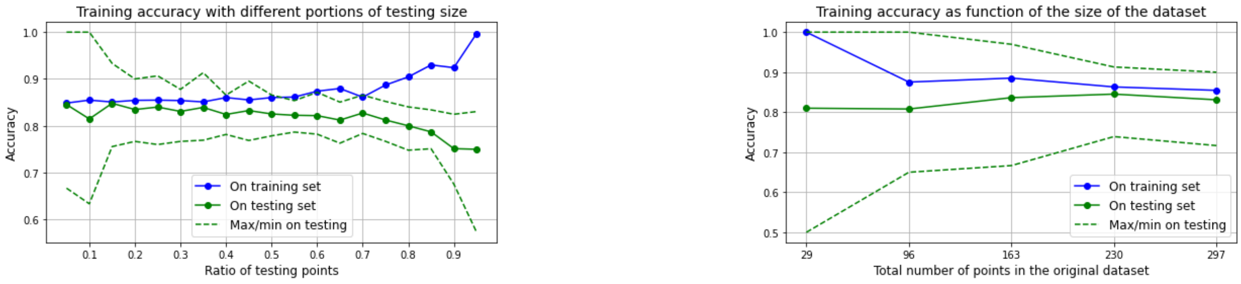


Figure 13: Effect on the accuracy of the number of hidden neurons when the convergence is achieved (right, 2000 epochs), and when no convergence is achieved (left, 500 epochs). The experiment without convergence was computed 30 times for each point.

The quantity of data in the *heart decease* is limited, however two final parameters related directly to the dataset are to be investigated. Using the best parameters defined so far, Figure 14a reveals an accuracy slightly but constantly decreasing when more data are used for the testing (i.e less for the training). It also highlights a large and constant variability of the performances for testing sets above 15% of the total dataset. The model does not necessarily performs better with more data, as also highlighted in Figure 14b, however the more data, the less volatile the results are. Note that this last experiment reveals that are still very largely spread (between 70% and 90% of accuracy).



(a) Testing ratio

(b) Quantity of data

Figure 14: Average evolution of the accuracy of a neural network with its learning rate (left) and its L2 regularization term (right). Computed with 30 independent trials for each point.

On the *heart decease* dataset, the best score achieved was an accuracy of 88.1% in average over 25 trials. The parameters used are: 20 hidden neurons, a logistic activation, a L2-regularization term of 10^{-2}, a learning rate starting at of 0.1 and evolving with the *adaptive* method. However the results are very volatile.

1.2.4 Best model for the heart decease dataset

For this first dataset, the best performing model is the neural network, that achieved a performance of 88.1% on average on 30 the binary classification task, changing only by the randomness of initial weights attribution and random separation of training and testing set. Under the same conditions, the random forest model reached 83.3% of accuracy and the decision tree 76.6%.

However if a final choice is to be formulated, the *random forest* would be chosen. Indeed, its results are always the same, while the accuracy of the neural network oscillates randomly between 70% and 90%. In that sens, the random forest brings guarantee of a good accuracy while the neural network does not, even if it performs better in average.

1.3 Part 2: Digits dataset from Scikit-learn

The second data set explored is the *digits* set (Figure 15) available in *scikit-learn 0.23.1* (equivalent to the famous MNIST dataset) containing 1797 individuals of 64 features (8x8 images). To turn the original exercise related to this dataset into a binary classification, the current challenge is to classify the images based on whether the figure drawn is odd or even.

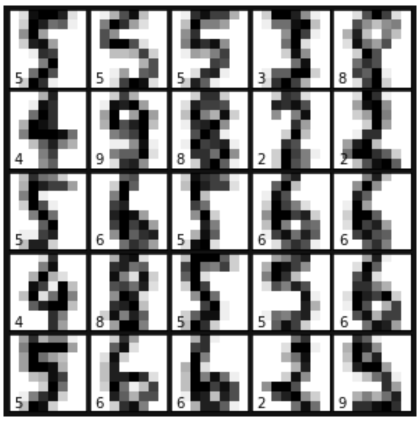


Figure 15: Some random examples of the *digits* dataset

The initial exercise is a famous problem of deep learning. The current dataset provided by Scikit-learn has apparently been modified to ease its use with simple algorithms, however the challenge (especially the related binary challenge proposed in this work) remains unusual and non-straightforward. This dataset is interesting, as the features are already meaningless and homogeneous digits (brightness of a pixel). The results obtained and exposed below reveal that high scores still can be obtained with all three methods.

A rapid analysis shows the decomposition of the images into lines, as well as the high values (ink to write the figures) centred in each row (Figure 16), meaning that the center vertical axis is black most of the time.

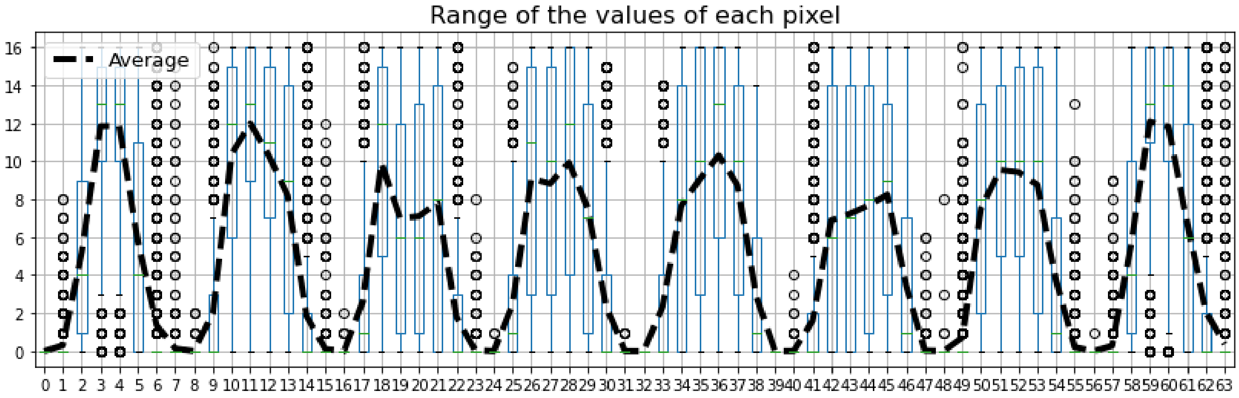


Figure 16: Distribution of the values of the *digits* dataset for each pixel

1.3.1 Decision Tree

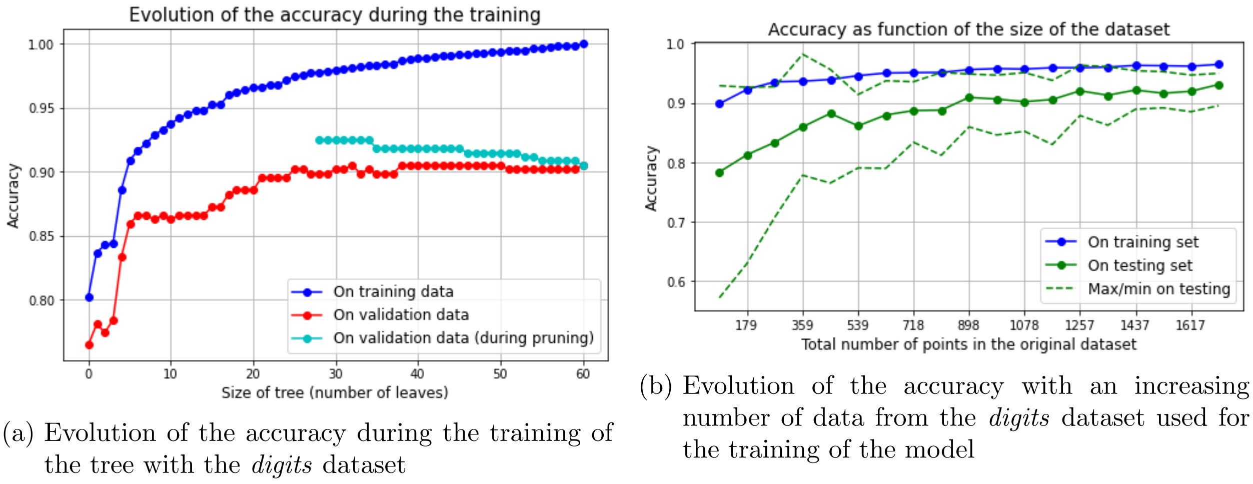
Due to the much larger number of samples and features in this second dataset, the decision trees are much larger. Therefore the different trends appear smoother, as the evolution of the accuracy during the learning process, like in Figure 17a, obtained with a training using the Gini information criterion.

However, based on the average result of 50 trees, a (non-significantly) better accuracy is reached with the entropy: 92.5% against 92.3% with the Gini criterion.

The more data the decision tree is trained with, the larger the tree (see in Appendix Figure 25 This characteristic should lead to systematic overfitting, however the evolution of the accuracy shown in Figure 17b highlights that the accuracy keeps increasing. In this exercise, having more data at disposal would then lead to even better results.

The accuracy is also subject to the ratio of training/testing data chosen. Figure 26 (see appendix) reveals a decreasing accuracy when the testing ratio increases, with a drop for more than 70%

of testing ratio. Therefore, the best accuracy for this model reached 92.5% with the entropy information criterion and a testing ratio of 20%.



(a) Evolution of the accuracy during the training of the tree with the *digits* dataset

(b) Evolution of the accuracy with an increasing number of data from the *digits* dataset used for the training of the model

Figure 17: Evolution of the accuracy during training of the tree (17a) and accuracy of the final model for different amount of draining data (17b)

1.3.2 Random Forest

For this model, a first implementation with the default parameters proposed by Scikit-learn reveal the same accuracy of 98.9% whether the *Gini impurity* or the *entropy information loss* is used. Arbitrary, the entropy information loss will be further used.

As for the previous dataset, the maximal depth and the maximal number of leaves are strongly related. The Figure 18 reveals that the random forest keeps increasing its accuracy up to a performances plateau starting at a *maximal depth of 8*.

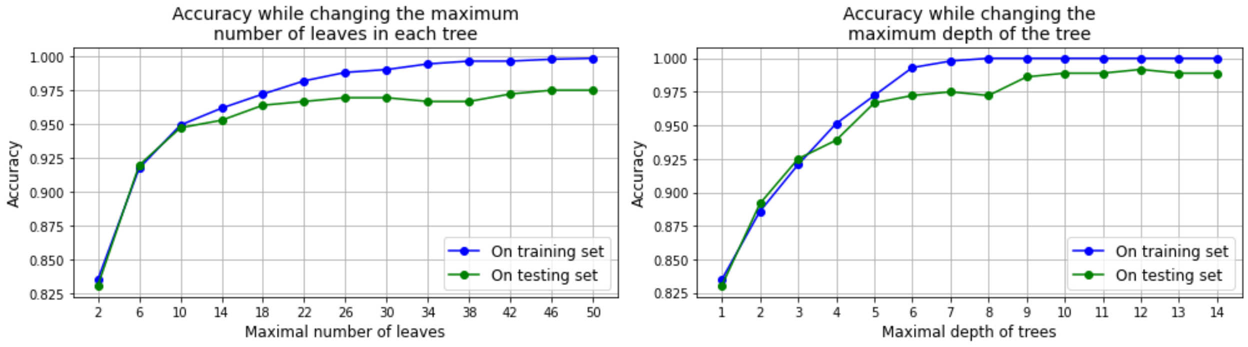


Figure 18: Effect of the maximal number of leaves (left) and maximal depth (right) of each tree on the performances of the random forest

Regarding the number of samples randomly selected for each tree, the accuracy reaches a plateau after 12% of the dataset and has almost no more influence. *This criteria will then be left to its default value: no limitation.*

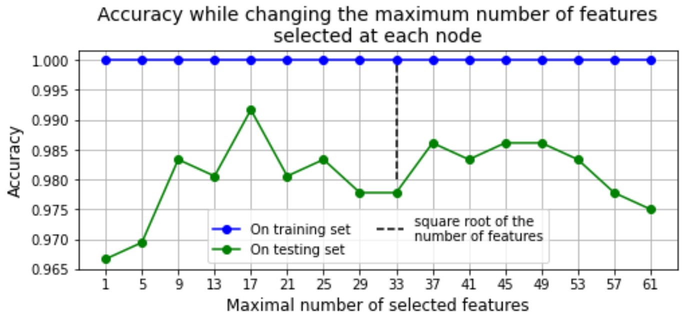


Figure 19: Effect of the maximal number of features on the accuracy of the random forest model.

n estimator	40
max features	17
max samples	None
max depth	None
criterion	Entropy

Table 3: Parameters of the best achieved random forest for the *digits* dataset

The maximum number of features to candidate for each best split plays a role in the accuracy, as seen in Figure 19. *The best number to chose for this parameter seems to be 17*, meaning that a choice made with only a fourth of the actual picture reveals to be the most efficient.

The number of trees in the random forest does not drastically influence the performances of the model: with 30 trees or more, the accuracy remains constant around 98%. A number of *40 trees* is chosen.

The training-testing ratio has no real influence on the score for the usual values (Figure 27c in appendix) and the number of data also plays a minor role (Figure 27d in appendix).

In definitive, with the parameters exposed in the Table 3, the best random forest model for this dataset reached 99.1% of accuracy, with only 3 wrong classifications (false negatives) out of 360 samples in the testing set. The importance of each feature in the model is presented in Figure 20, and confirms that edge pixels have no importance. One pixel (centered, at two third from the top) appears to be particularly determinant for the choice of this final model.

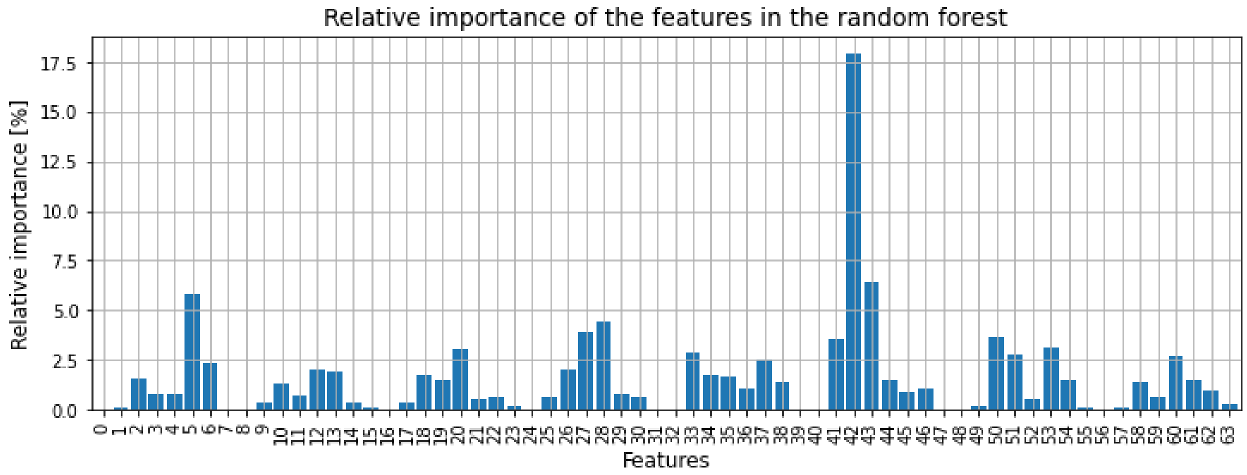


Figure 20: Relative importance of each feature in the random forest model. This importance is computed in *scikit-learn* based on the improvement of the Gini index caused by a split on the given feature

1.3.3 Neural Network

The classification of the *digits* dataset with a neural network is realized directly on the data. A first experiment ran for various parameters reveals that a modeling based on standardized data is 2% less accurate in the best cases. This can be explained by the fact that the important pixels are also the ones with high values, as expressed in Figure 20.

As for the first dataset, the parameters have been explored with a *grid search* method (Figure 21). Once again, the results show a great variety of possible parameters that lead to an accuracy above 97% (the maximal achieved score being 97.7%), with no clear trend whether a choice of parameter is better that another in absolute.

Starting from the optimal parameters of the best result of the above mentioned grid search, the effects of each parameter is visually examined based on a cross-validation with 5 folds. The following trends are observed in Figure 23: first, the *identity* activation function performs less good than the other proposed non-linearities. Then, the learning rate must initially be smaller than 2.10^{-2} to guarantee stable results and the L2 regularization term smaller than 1. Finally the number of neurons greater than 20. In the acceptable domains, there is no clear evolution of the accuracy.

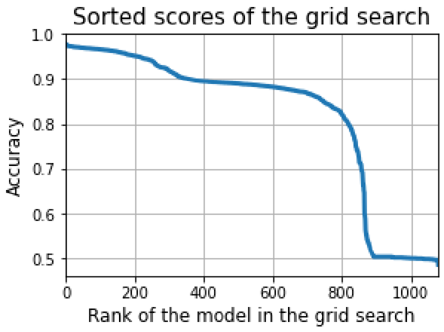


Figure 21: Sorted scores of the grid search

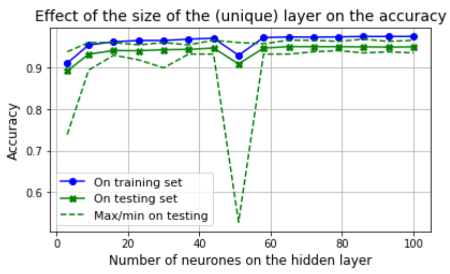


Figure 22: Accuracy for various number of hidden neurons

Due to the small value of the initial learning rate chosen (Table 4), the study of various method for handling the learning rate during the training lead to the exact same observations each time.

A further investigation of the *early stopping* and *warm start* parameters along with the *tolerance*, the *validation fraction* and the *maximal number of epochs* reveals slightly better results with an early stopping, and no significant effect of the four other characteristics (Figure 28 in appendix).

To compare with the results with the *heart decease* dataset, here the results obtained during the above experiment are much more stable. Figure 22 illustrates

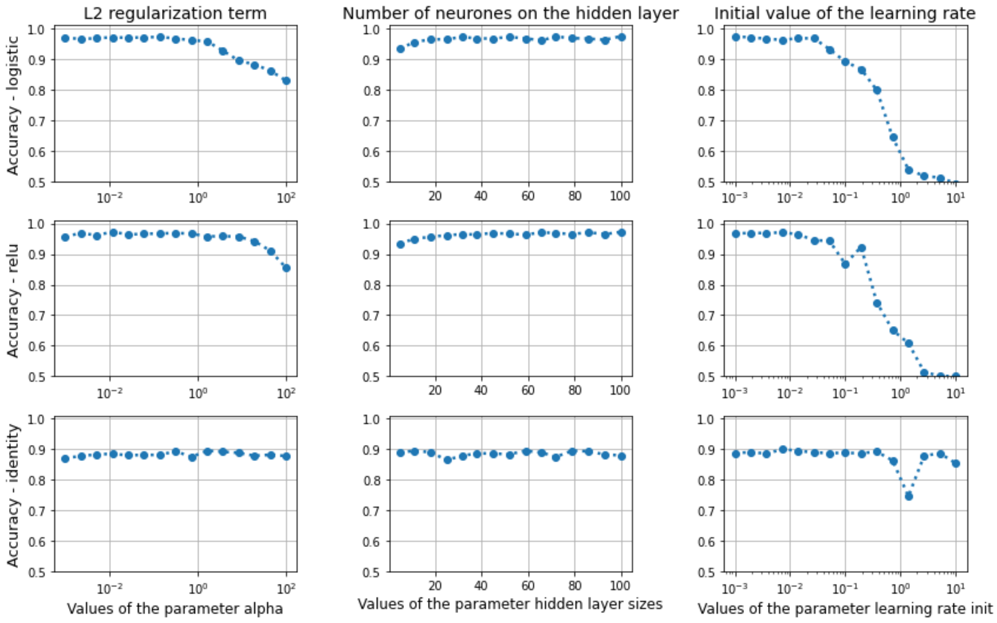


Figure 23: Effect on the accuracy of the neural network of the L2 regularization term, number of neurons on the hidden layer, initial value of the learning rate, for three different activation functions.

Parameter	Value
neurons	60
activation	logistic
L2 regularization	0.1
learning rate	7.10^{-3}
evolution method	adaptive
early stopping	True
maximum iteration	700
validation fraction	20%

Table 4: Chosen parameters for the neural network model

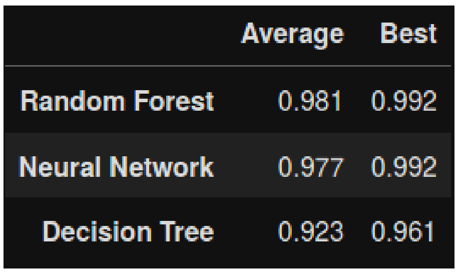
this stability on 30 estimations of the the accuracy per size of the hidden layer: with only one exception at 50 neurons, the difference between minimal and maximal achieved performance is small.

The model has also been challenged for various amount of available data and different ratio of training-testing sets for 50 trials. The results presented in appendix reveal a regular decrease of accuracy for an increasing proportion of the training set (Figure 29a) and an also regularly increasing accuracy when more data is available (Figure 29b). For the training-testing ratio, as no conclusion can be drawn, 20% of the data is reserved for the testing. Regarding the quality of data, the experiment only shows that the model is more reliable with larger amounts of data, and that with more data available, the neural network still could improve its performances.

Using more data brought stability to the results of the neural network. However after a few tests to adjust the default parameters for further experiments, it appears than varying individually each parameter shows no significant change in the performance. The best model achieved, with the finally chosen parameters presented in Table 4, a performance of 98.3% of accuracy on a testing set.

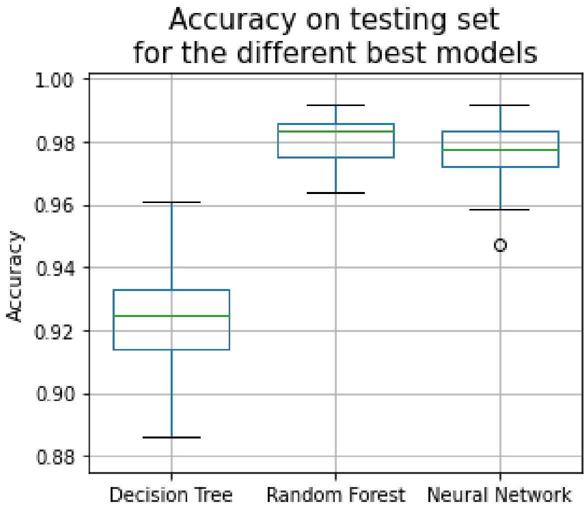
1.3.4 Best model for the digits dataset

All three models achieve high performances. The final chosen model is the random forest. Among 50 different experiments ran for each models with the same parameters, the random forest achieved the best testing accuracy, the better performances in average (Figure 24a) and also brings more guarantee on the result than the neural network (Figure 24b).



	Average	Best
Random Forest	0.981	0.992
Neural Network	0.977	0.992
Decision Tree	0.923	0.961

(a) Best results for the three models



(b) Boxplot of the comparison between models.

Figure 24: Best results achieved with the *digits* dataset. The experiment has been realized with 50 trials for each models, with identical parameters.

3 Appendix

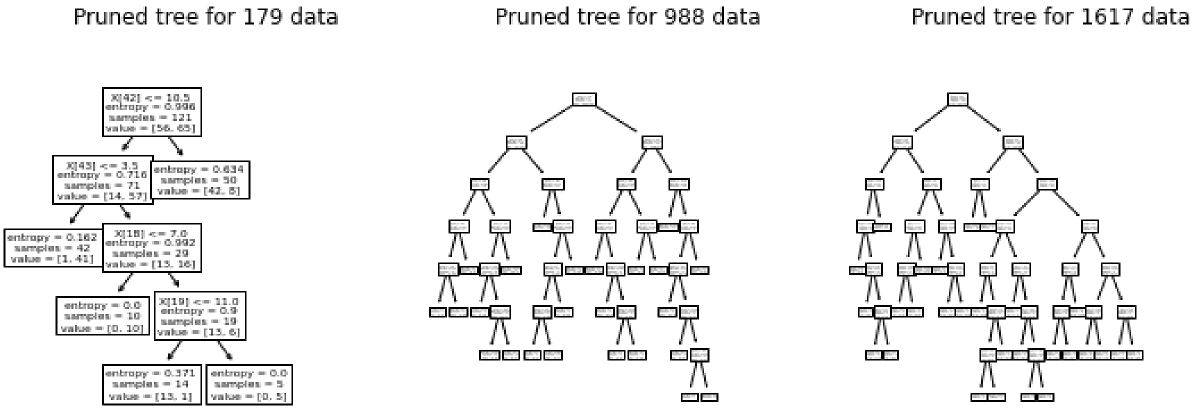


Figure 25: Decision tree for tree different sizes of sub-dataset considered. The more data is available, the bigger the tree. The model is then more likely to overfit.

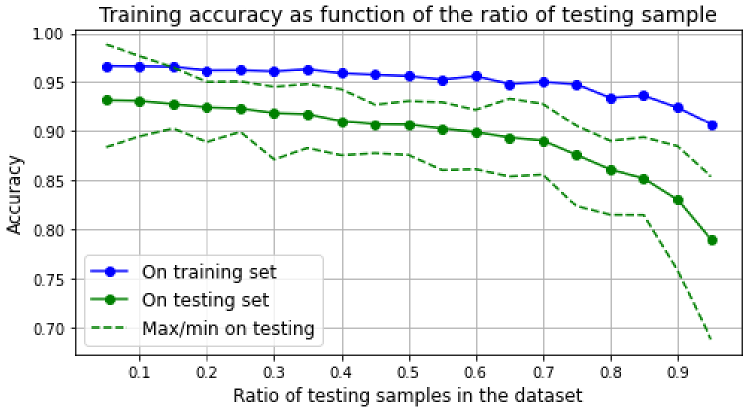
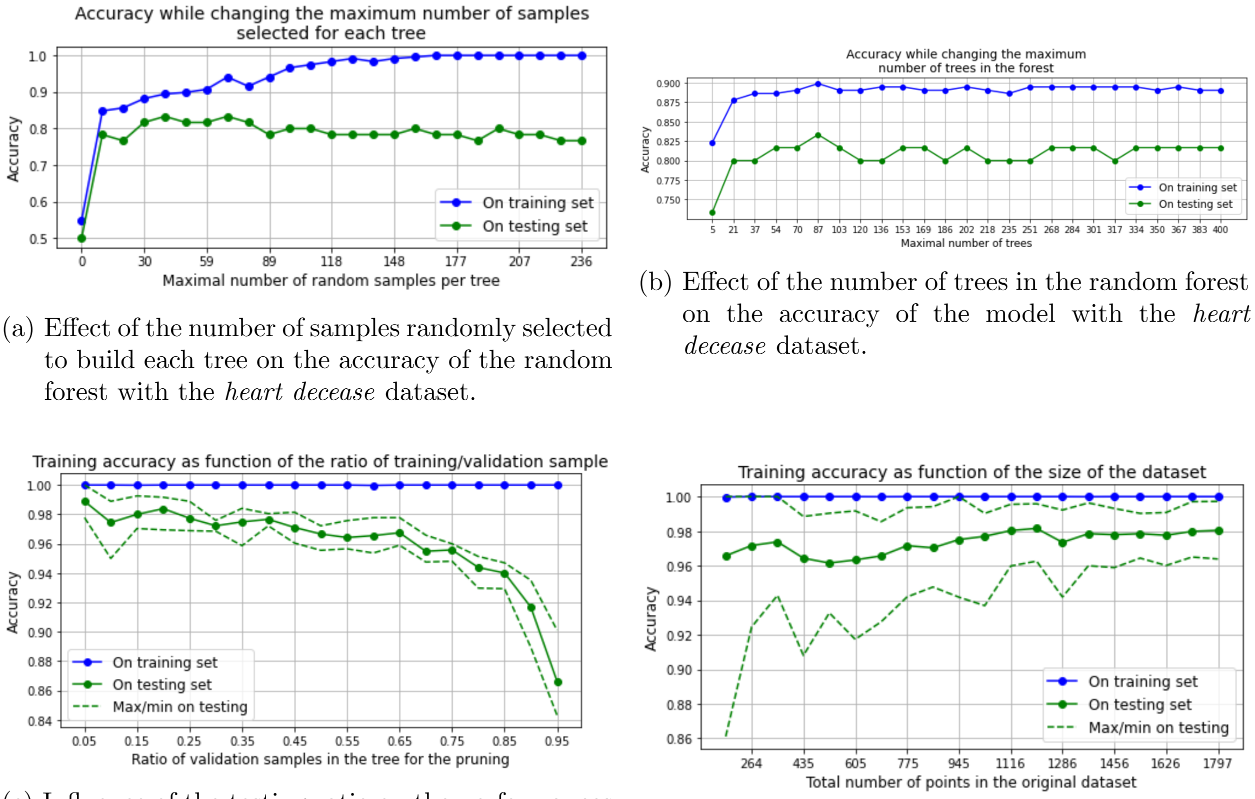


Figure 26: Evolution of the accuracy of the decision tree with increasing ratio of training samples (therefore decreasing number of training data).



(a) Effect of the number of samples randomly selected to build each tree on the accuracy of the random forest with the *heart decease* dataset.

(b) Effect of the number of trees in the random forest on the accuracy of the model with the *heart decease* dataset.

(c) Influence of the testing ratio on the performances of the random forest method applied to the *digits* dataset

(d) Accuracy of the random forest model on the *digits* dataset for various quantity of data available

Figure 27: Results of various experiments with a random forest binary classifier model

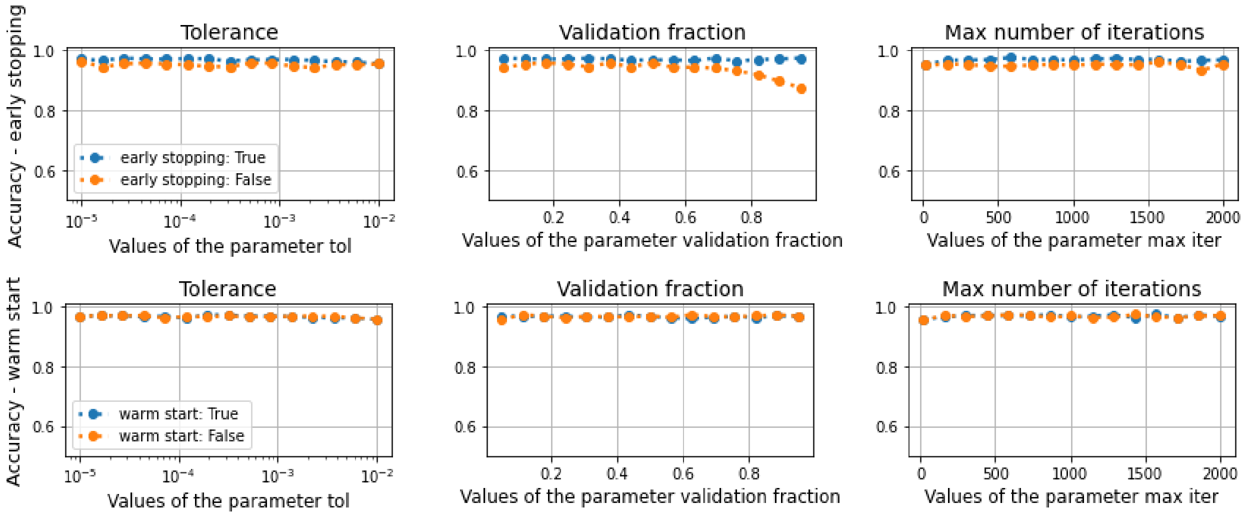
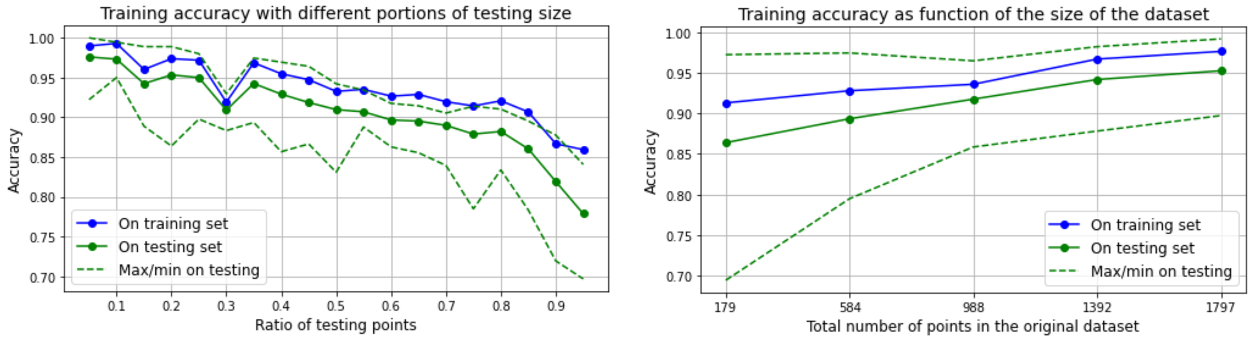


Figure 28: Effect of the warm start, early stopping, along with the tolerance, the validation fraction and the maximal number of epochs in the neural network model for the *digits* dataset.



(a) Training-testing ratio

(b) Quantity of available data

Figure 29: Effect of the quantity of data available (right) and reserved for the testing set (left) on the accuracy of the neural network with the *digits* dataset.