HW1 Report
Supervised Learning

Subject :
CS 7641 Spring 2017 - OMS

Author :
Melisande Zonta Roudes
GT account name : mzr3

February 6, 2017

The goal of this project is to test five different algorithms : decision tree classifier, boosting, k nearest neighbors, support vector machine and neural network on two interesting datasets.

1 Dataset Choices

In order to choose well the two datasets, many criteria are considered.

1. The first one is about the number of instances and attributes indeed the inability of running the datasets is eliminatory however the datasets must sufficiently large to abide by the rule of the 2/3 training size vs 1/3 testing size.

2. The second one is the type of object contained in the classes (numeric, boolean or character) and the output is either categorical or binary.

3. The third criterion is about the associated task (classification or regression), as our algorithms are classification ones.

4. Finally, it is important to analyse the histograms of each parameter relatively to the others (scatter matrix) to detect if those datasets are linearly sharable and thus if they can be considered as "easy" datasets or not.

These were my tests for selecting a set of data nevertheless the list produced was not the final one because the experiments made on these datasets created another factor of selection which are the results over the different algorithms. Indeed, the "breast cancer" dataset presents so much good results for every algorithms, it was difficult to put forward the properties of each one. My two datasets were picked up from the UCI repository in Machine Learning : `https://archive.ics.uci.edu/ml/datasets.html`.

1.1 First Dataset

The first dataset called "Pima Indians Diabetes Data Set" is about the diagnostic of diabete among Pima Indians. The instances were determined from female patients aged of at least 21 years old. The attribute characteristics are integer and the associated task is classification. Moreover, there are 768 instances and 8 attributes which makes mandatory the use of cross validation. The attributes are some biological properties as the glucose concentration or the blood pressure and the other are the age or the number of pregnancy. The output is binary. Although the main disadvantage of this dataset is the presence of missing values which were replaced by zeros (biologically impossible for some features). The interesting results obtained allowing to compare the algorithms performances convinced me of the advantage of this dataset.

1.2 Second Dataset

The second dataset called "Letter Recognition" is the identification of black-and-white rectangular pixel displays as one of the 26 alphabet's letters. 20000 instances are exploited and 16 attributes were chosen to produce the output. The attribute characteristics is again integer and the associated task is classification. However the interest of this dataset stands in the categorical type of the output (the 26 different letters) and of course the large number of instances.

1.3 How to handle those two datasets ?

When coming to analyse the performances of our algorithms on our datasets, the first step is to compute the training score and the testing score. To do so, we have first splitted our dataset in training set which represent 2/3 of the dataset and testing set (the remaining datas). In order to visualise if those scores are iterable, we compute the Monte Carlo curves on 1000 iterations as we can see on 1 *a.* and *b.*. If the standard deviation is greater than 2%, we would rather use cross-validation on our datas ie we divide the datas into k_{folds} and then select one of the k_{folds} as our testing set whereas the others are the training ones. The cross validation used is a stratified one indeed when looking at the repartition of the instances in the classes of the first dataset we can see that on 768 instances, 500 are standing in the class of value 0 (Not diagnosed) and 268 in the other class, so we must split our data fairly (the same percentage in each class). On figure 1 c. the standard deviation after cross validation is still greater than 2% so the cross validation is not sufficiently efficient whereas figure 1 d. shows 1.4% standard deviation which guarantees the accuracy of the stratified cross validation. The second dataset is really fairly splitted for the 26 categories (700 instances each). In order to improve the first dataset accuracy score, we use another type of cross validation which assures stratified randomized folds.

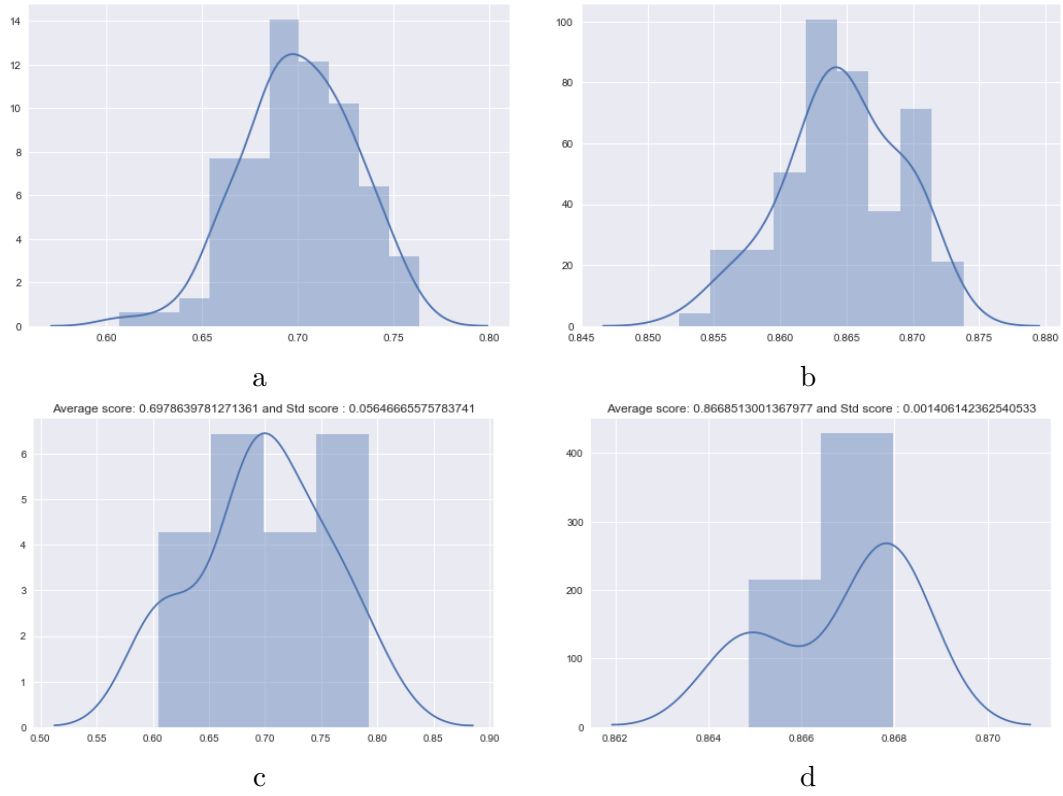
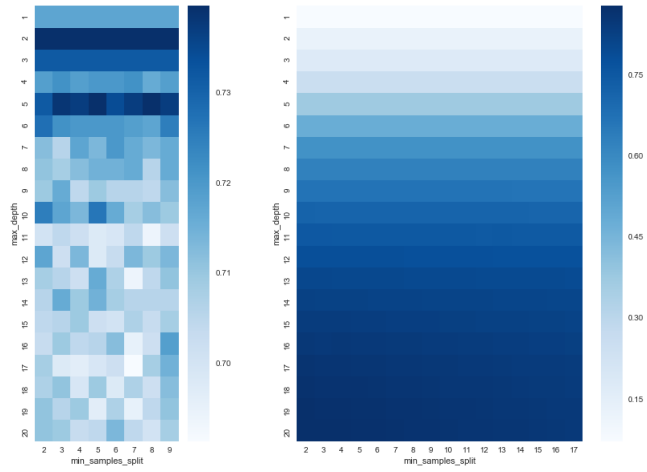


Figure 1: Histograms
a. Dataset 1 : Before cross Validation. *b.* Dataset 2 : Before cross Validation.
c. Dataset 1 : After cross Validation. *d.* Dataset 2 : After cross Validation.

2 Decision Tree Classifier

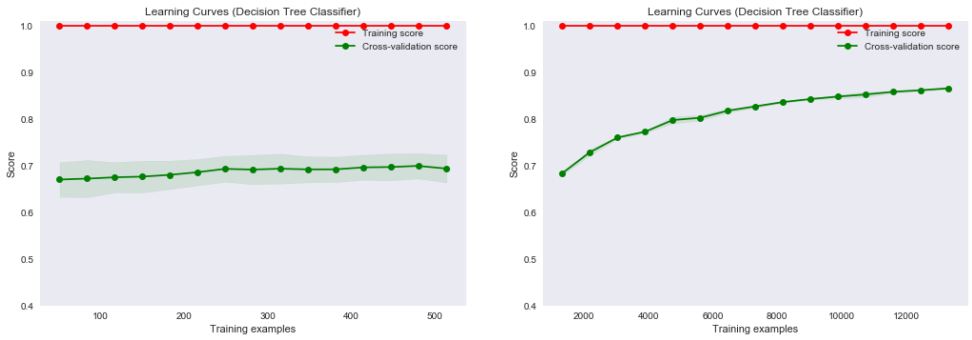
The classification tree allows to predict to which class will belong the instance based on the different features of our instances. The one we use belongs to Classification and Regression Tree, it is relative to the C4-5 algorithm which converts the trained trees into sets of if-then rules but CART is different from C4-5 because it handles numerical target variables. The decision is taken by computing the information gain (from the entropy value) and allows to determine the best attribute. CART builds binary trees by using feature and attribute that will lead to the largest information gain. After having followed the steps mentioned previously, we tune our parameters in order to find the ones that will provide the best accuracy score. The most interesting features are maximum depth of the tree and the minimum number of samples that will lead to the division of the node. These two parameters will prevent from over fitting if well determined. Our approach corresponds to stop the growing of the tree before it becomes too large whereas the most usual one which is called pruning consists in testing the original tree against some pruned versions of it in order to delete the unnecessary nodes.



a b

Figure 2: Cross Validation Grid
a. Dataset1. *b.* Dataset2.

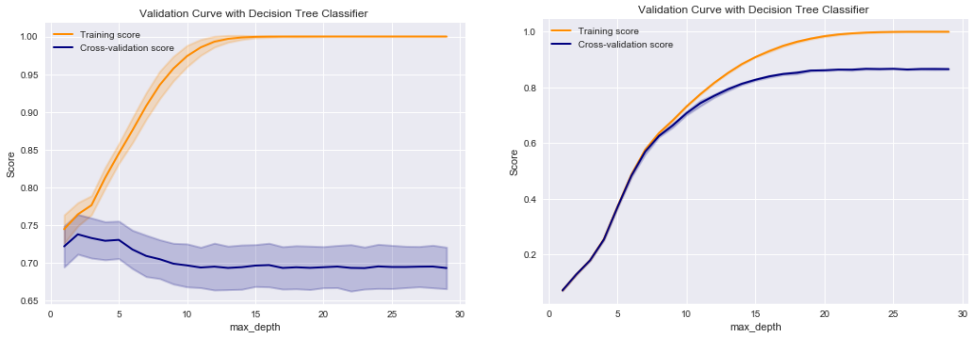
The figure 2 a. shows the presence of overfitting for the first dataset indeed when a depth of 5 is reached the accuracy score begins to decrease. The second dataset presented on figure 2 b. has no overfitting, we can just conclude that a depth of 9 is sufficient to provide the maximal accuracy score. The second parameter tuned can be used for pruning but is not really relevant on those curves especially for the second dataset which is not at all sensitive to this feature.



a b

Figure 3: Learning Curves
a. Dataset1. *b.* Dataset2.

The figures 3 a. and b. represent the learning curve of our decision tree classifier for our two datasets. These learning curves show the improvement of the performances in function of the size of training set to demonstrate the learning process. Whereas on the right figure, the cross-validation score is converging towards the training score when the number of training examples is increasing, it is not the case for the first dataset which is slightly better for an important number of training examples but cannot be relevant.



a b

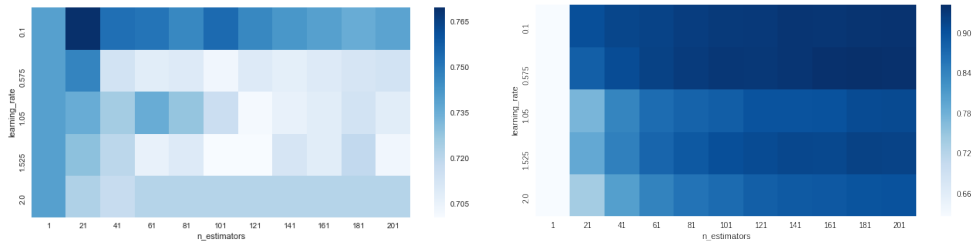
Figure 4: Validation Curves
a. Dataset1. *b.* Dataset2.

The figures 4 a. and b. represent the validation curve for the maximum depth parameter of our decision tree classifier. These curves show, as mentioned in the analyses of the above grid, the presence of overfitting for the first dataset (the curve is increasing until it reaches its maximum and then is decreasing) whereas the other show that when the plateau is attained we can stop the growing of the maximum depth moreover the underfitting can be noticed before the plateau.

3 Boosting

The principle of the boosting which belongs to the ensemble methods is to combine multiple weak learners to obtain a high accuracy score. A weak learner is a learner whose has a

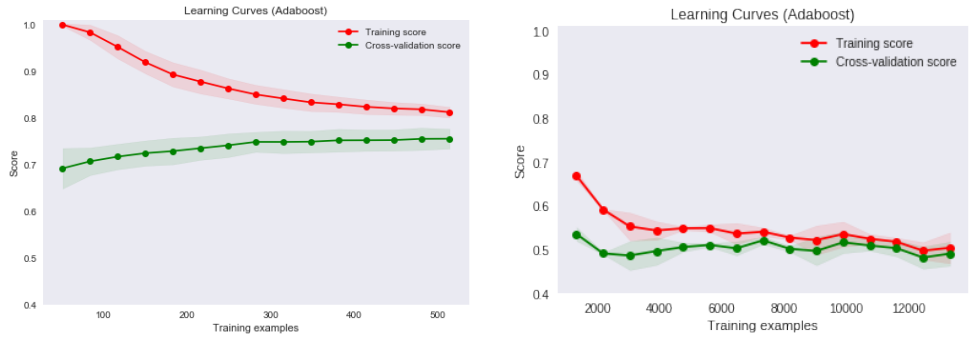
performance score a little bit better than the score by random guessing. The algorithm used in our case is the AdaBoost Algorithm (Adaptative Boosting). During the training, the set of weak learners fit the training data and after they are linearly combined (with weights) to build a strong classifier. The weak learner used here is a decision tree classifier which is one of the popular algorithms used for boosting because of its lower generalisation accuracy compared to the four other learning methods. In order to ensure the low accuracy score, by observing one more time the figure 2 we choose for the first dataset a maximal depth of 2 and for the second one a maximal depth of 8.



a b

Figure 5: Cross Validation Grid
a. Dataset1. *b.* Dataset2.

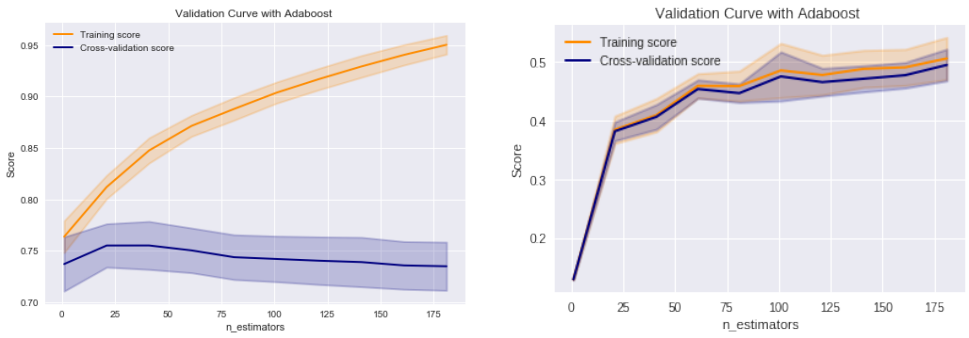
To identify the best parameters for the AdaBoost algorithm, we build our map in function of the number of estimators and of the learning rate which handles the contribution of each classifier. The figure 5 a. show that the best parameters for the first dataset are 21 estimators and a learning rate of 0.1. We can see the underfitting and the overfitting respectively before and after 21 estimators. The 5 b. provides the best results for 181 estimators and a learning rate at 0.55. We notice underfitting but still no overfitting.



a b

Figure 6: Learning Curves
a. Dataset1. *b.* Dataset2.

The figures 6 a. and b. show the convergence of the training score and of the cross-validation score for both datasets. The accuracy score is not great since the training score is far from 1 and they are decreasing while the number of training examples is growing.



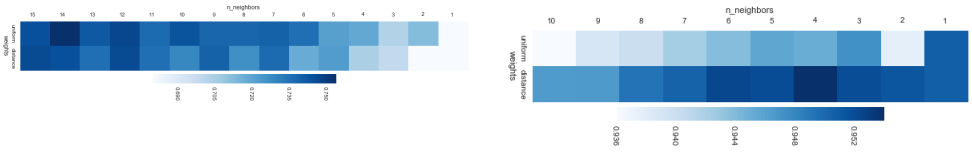
a b

Figure 7: Validation Curves
a. Dataset1. *b.* Dataset2.

The figures 7 a. and b. show the training score and cross validation score in function of the number of estimators, we recover the same remarks on the overfitting of the first dataset after 21 estimators and the underfitting of the second dataset but the absence of overfitting. On both curves, the training score is increasing towards 1 which shows that the more the number of estimators is important, the better the algorithm learns.

4 K Nearest Neighbors

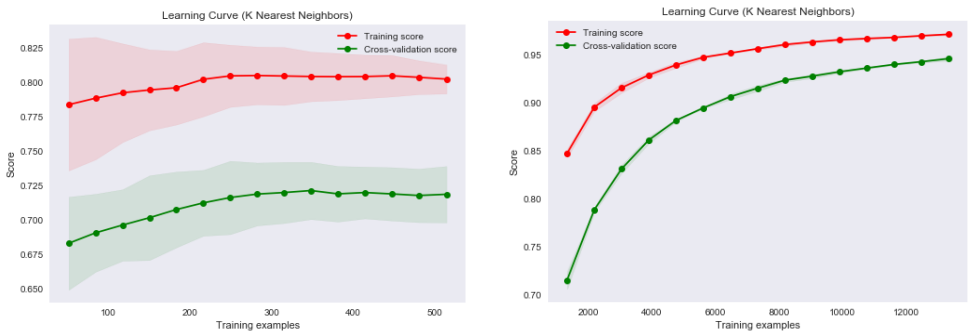
The K Nearest Neighbors algorithms is the only example of lazy learner among the five algorithms. A lazy learner does not have the training step before data are received, it computes a function to fit to the training data only when a new query is sent. This algorithm is related to the distance between data points indeed two close points are expected to have a same value. This leads to the induction of the parameters that have to be tuned to obtain great performances. We will make vary the number of neighbors to consider around the position of our new query and the weights used to handle the neighbors' impact.



a b

Figure 8: Cross Validation Grid
a. Dataset1. *b.* Dataset2.

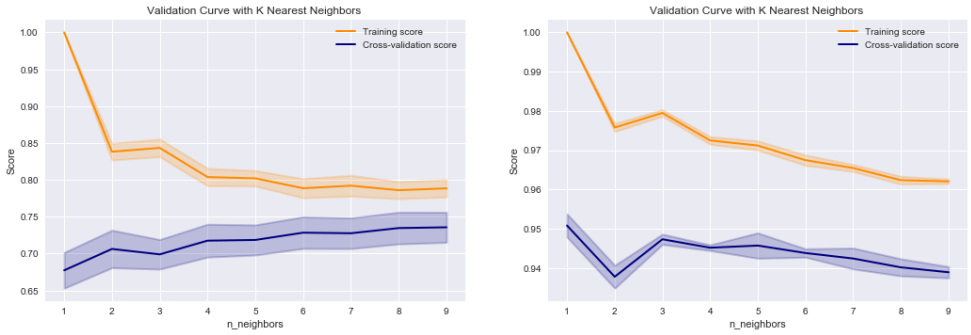
The figure 8 a. show that the best parameters are for 14 neighbors and uniform weights. We can well notice the underfitting and the beginning of an overfitting. We can notice on the figure 8 b. the presence of overfitting after the best parameter is reached that is to say 4 neighbors and the inverse distance is considered for the weights. We detect overfitting once this value is reached.



a b

Figure 9: Learning Curves
a. Dataset1. *b.* Dataset2.

The figure 9 a. and b. show that the learning score and cross validation score are improving while the number of training examples is increasing. The learning score of the second dataset is really close 1 and as before does not suffer of overfitting. The first dataset has not good results which could be explained by the missing values which make false the research among the neighbors.



a b

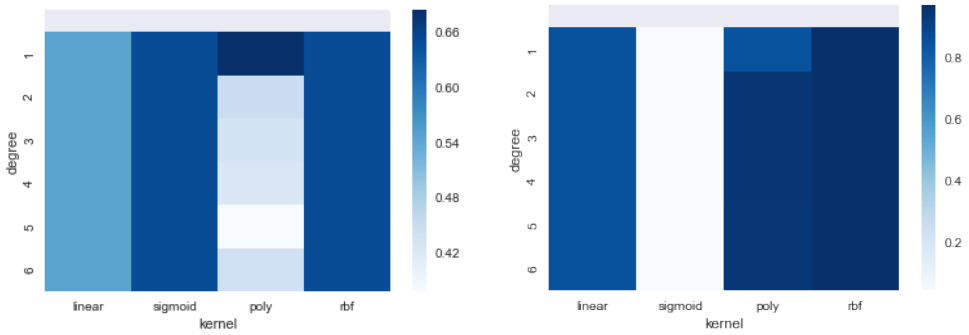
Figure 10: Validation Curves
a. Dataset1. *b.* Dataset2.

The figures 10 a. and b. plot the accuracy score in function of the number neighbors, besides showing the value of the parameters that fit the best our training data as mentioned before, we can notice that the training score is far from 1 for the dataset 1 and that some overfitting is present for dataset 2 which is not the case for the other algorithms.

5 Support vector Machine

The Support vector Machine algorithm is about separate datas in order to cluster them. The datas need to be linearly separable and the goal of the algorithm is to maximize the margins between the groups. Different kernels can be used according to the type of division we can do between groups : hyperplanes, circles ... That leads us to the parameters we can tune which

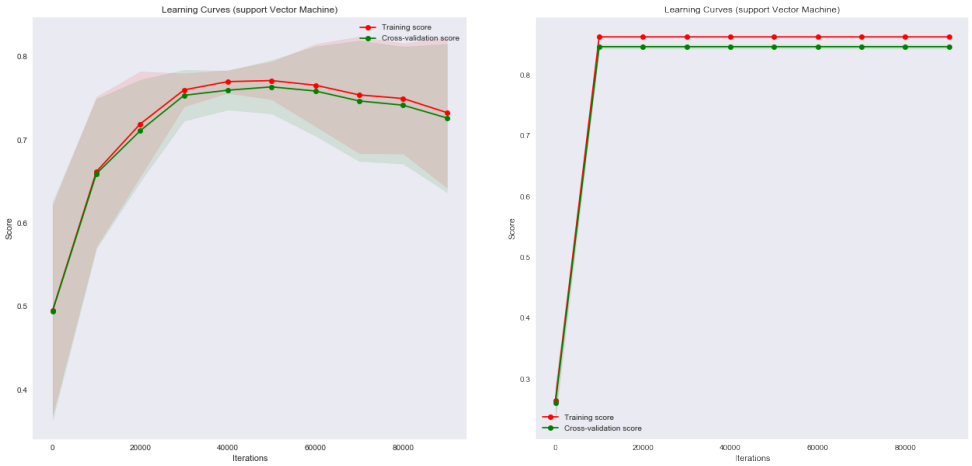
concerns the function to compute to constitute the groups, that is to say the kernel type and the degree associated.



a b

Figure 11: Cross Validation Grid
a. Dataset1. *b.* Dataset2.

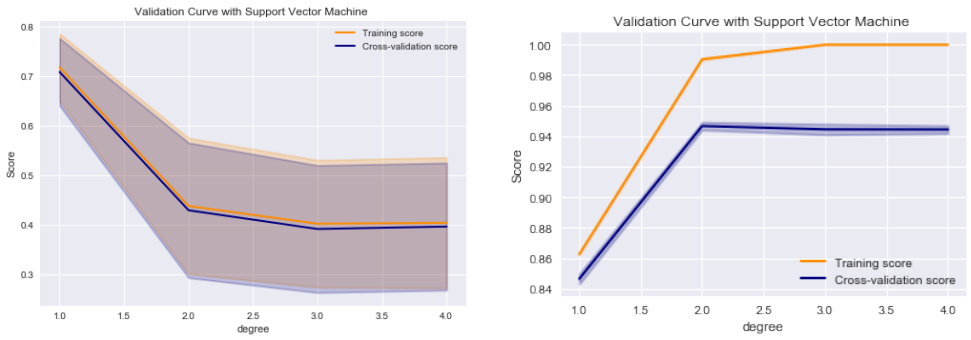
The figure 11 present interesting results since the first dataset reach its best parameters for a polynomial kernel with a degree of 1 where as the second dataset has the best results with a radial basis function which means that the first one is linearly separable and that the second one is clustered in circles.



a b

Figure 12: Learning Curves
a. Dataset1. *b.* Dataset2.

This time we compute our learning curves in function of the number of iterations and we will do the same for neural networks because they are both iterative algorithms The figure 12 b. shows that we don't need to compute so much iterations, we could have stop to 20000 iterations. We can notice on the figure 12 a. that the scores are decreasing when 50000 iterations are reached so we would better stop there.



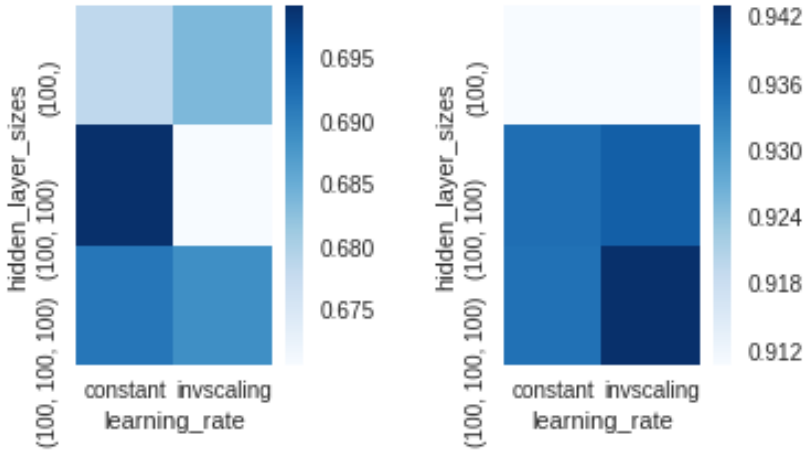
a b

Figure 13: Validation Curves
a. Dataset1. *b.* Dataset2.

The figure 13 a. and b. represent the training and cross validation score in function of the scores. We recover the same observations as before concerning the better performance of the degree 1 for the polynomial kernel and we see better results for degree greater than 1 concerning the kernel polynomial for the second dataset.

6 Artificial Neural Network

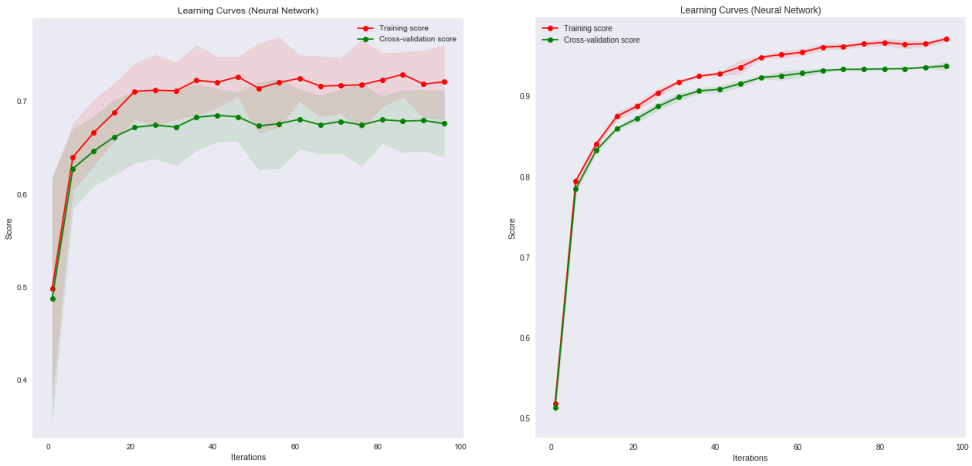
Finally, we computed the algorithm with the most complex structure among the five. We chose a multi layer perceptron which is the basic neural network. Designed to look like the human neurons, the most important parameters are the number of layers and the number of neurons in each one. Once the structure determined, we have to consider the activation function usually the sigmoid one used in the gradient descent as a non linear thresholding function. Finally, the learning rate can be adjusted in the gradient descent rule. As this algorithm is computationally really expensive, the activation function was set to the logistic one (sigmoid) which seemed to provide the best results and the hidden layers were tested on few values.



a b

Figure 14: Cross Validation Grid
a. Dataset1. *b.* Dataset2.

The figure 14 a. demonstrates that the best results are found for a neural network with two hidden layers (4 layers with the input and the output) of 100 neurons each and for a constant learning rate whereas the figure 14 b. has the best accuracy score for a 5 layers neural network (3 hidden layers) with 100 neurons each with a decreasing inverse scaling exponent of the time.



a b

Figure 15: Learning Curves
a. Dataset1. *b.* Dataset2.

The figure 15 a. show the iterations could be stop at 20 because it's always instable and not good whereas concerning the second dataset the figure 15 b. show an increasing score when the number of iterations is higher.

7 Comparison between the five algorithms

In order to conclude on these five algorithms, all scores are showed. The first one indicates the score after cross-validation and the second one concerning the testing score represent the one reached with the best parameter which fitted the training data. The training score could have been extracted from the validation curves but the graduation being not sufficiently precise, it has been preferable not to put them.

Table 1: Comparison table

		Decision Tree Classifier	Boosting	K Nearest Neighbors	SVM	Artificial Neural Network
Dataset 1	Training score	1.0	0.97	1.0	0.30	0.73
	Testing score	0.69-0.87	0.72-0.77	0.72-0.76	0.36-0.68	0.63-0.69
Dataset 2	Training score	1.0	0.94	1.0	0.87	0.94
	Testing score	0.86-0.88	0.87-0.95	0.94-0.96	0.95-0.97	0.92-0.94

We can observe on this table that the gridsearch parameters improve almost all the time the testing scores, this is striking for the decision tree classifier and the support vector machine algorithms.
However concerning the first dataset on diabetes, the results stay average probably because of the missing data which can be handle by the decision tree but not as well by KNN or SVM for example. The second dataset presents really good results and recovers a slight improvement from the tuning. In conclusion, by judging on the repartition by class of the first dataset, it seemed imperative to do randomised stratified cross validation whereas on the other dataset simple cross validation would have been sufficient.
These datasets demonstrate how the decision tree classifier, the boosting method, the K Nearest Neighbors, the Support Vector Machine and the artificial neural network handle missing values and binary ones (dataset1) and categorical, non separable values (dataset2).
Finally, we can say a word on the computation time of these algorithms. Concerning the dataset 1, the decision tree classifier was the fastest, followed by KNN, then SVM, boosting and finally ANN was the slowest. However, for dataset 2 the KNN was the fastest followed by decision tree, SVM, boosting and ANN.
To conclude, we tested different types of algorithms, tree classifier, ensemble methods, neural networks and SVM which are eager leaners and KNN the lazy learner.