Supervised machine learning

BOISSIN Thibaut

table of contents

I. Problem description

1. Fashion MNIST

This data-set contains 28*28 pictures of various clothes sold on the internet. The objective is to classify the type of clothes in the picture.
This data set has some interesting characteristics such as :

- well balanced multi-class target
- bounded numerical features
- fairly decent amount of data
- very clean data (no missing values, no outliers)

On the other hand this problem doesn't have any trivial solution. This data-set has also high dimensions as it has 784 features.
The is 60000 rows in the training set, assuming the data is homogeneously distributed, we have : $60000/784 * 9 = 688$ rows per dimension per category which seems to be enough to avoid Bellman curse. The presence of e 10000 rows test set permitted us to avoid the usage of k-folding (which may be computationally quite expensive).

2. Wine quality scoring

This data-set contains various information about different wines such as the name of the winery, the price of the bottle, the country of origin... The objective is to estimate the quality score of a wine (regression task). This data-set is quite interesting as is has more realistic data :

- many text features with many different values
- presence of potentially unbounded values (we hope a 3000$ bottle is not common !)

This data-set is interesting as it contains more realistic data, which are more difficult to handle.

2.1. Pre-processing

The data of the wine data-set were processed as follow :

- normalization of the price
- keeping the categorical values that have at least 20 instance in the data-set
- apply the one-hot encoding on these categories

The idea of keeping only values that appears 20 times in the data-set comes from a simple reason : as these text values are not ordered, the one-hot encoder seems to fit the case. But using the one-hot encoder can lead to a very high amount of columns, which may be difficult to process.
A feature was initially present in the data-set, containing a list of adjectives qualifying the wine. With more time it would have been great to include this feature with a tf-idf representation.

II. Tree Classifier

1. On fashion MNIST data-set

On one hand this data-set the tree got fairly good results, as the data-set appears to be fairly linearly separable. But on the other hand we expected the algorithm to

quickly over-fit as the trees make it's decision on single pixels. The following figure 1 shows that the algorithm tends to over-fit with trees where $max\_depth \geq 10$. Note that tou can that the accuracy has been chosen as metric, but a custom defined metric would have been great to define. For instance have a confusion between "Sneaker" category and "sandal" categories is less problematic than a confusion between "Sneaker" and "Dress".

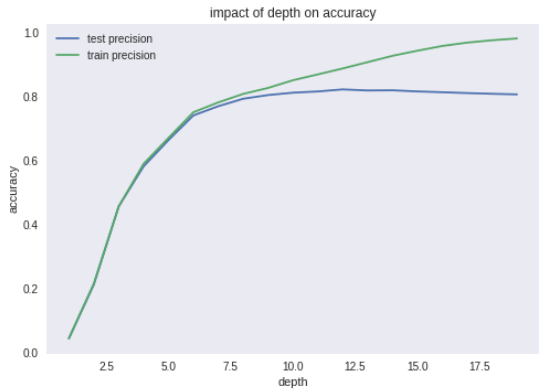


FIGURE 1 – learning curve of the tree on the Fashion MNIST data-set

For the time efficiency of the algorithm, the results shown in the figure 2 seems linear. The training duration was first thought to be related to the number of splits in the trees, which would be 2^{depth}. But as we make partitions, the amount of data processed at a given depth remain equals to the data-set size. This is why the time complexity is $m * O(n)$ where m is the data-set size and n the depth of the tree. For the testing time the complexity of the algorithm is $O(n)$ (number of nodes traversal before reaching a leaf).

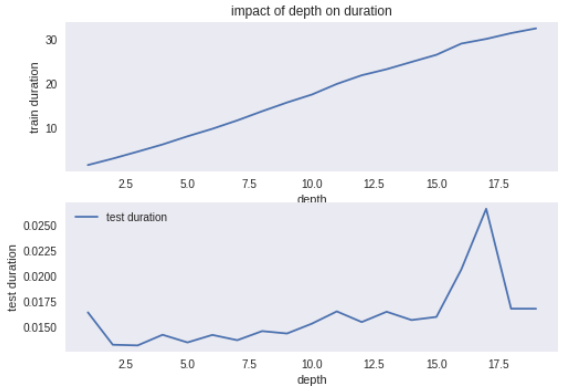


FIGURE 2 – Training and Prediction duration of the tree algorithm on the Fashion MNIST data-set

2. On Wine quality Scoring data-set

On this data set the trees were expected to work poorly as the chosen task is a regression task. The figure 5 shows sign of over-fitting from $depth \geq 10$. This leads to a maximum mean average error of 2.0 points. This results would have certainly

been improved by applying a log transformation on the price feature (see appendix 1)

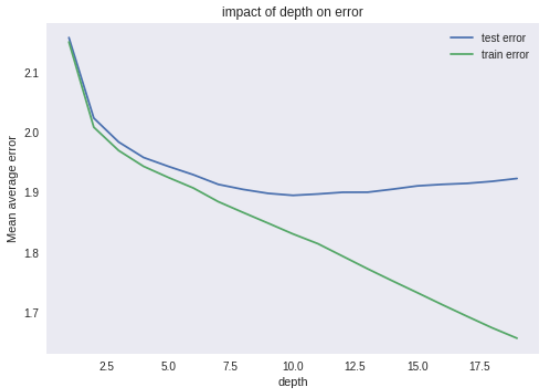
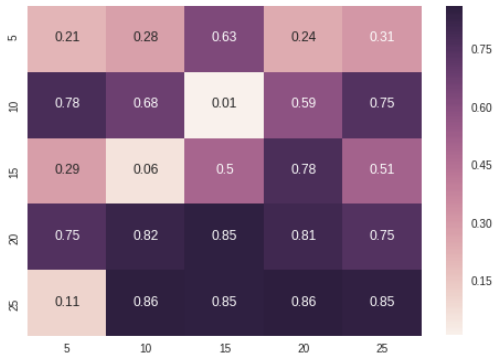


FIGURE 3 – learning curve of the tree on the Wine scoring data-set

III. Neural Networks

1. Fashion MNIST

Usually, two hidden layers are enough to approximate any function, but in this particular case (imahe recognition), it can be a good strategy to have narrower and deeper networks. This is why deep networks have been explored, also the small size of images allow us to still work with fully connected networks. The fig 4 shows the accuracy results for the tested values. As we can see better results may be expected with larger networks but we couldn't afford an other stage of grid search (computationally expensive).



Vertically : width of the network, Horizontally : depth of the network

FIGURE 4 – grid search results on Fashion MNIST data-set.

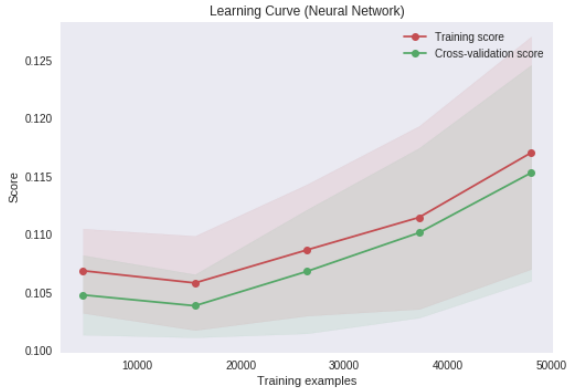


FIGURE 5 – learning curve of Neural Networks on fashion MNIST

IV. Appendix

1. price repartition in the wine scoring data-set