TELECOM PARIS

SI221

TP2 : k-Nearest Neighbors

Auteur :
Paul Fayard, Pablo Pevsner

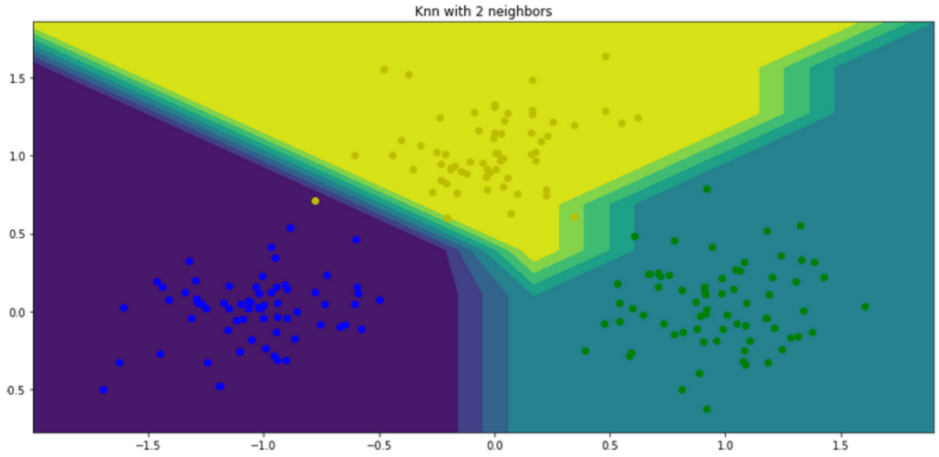
Professeur encadrant :
Aslan Tchamkerten

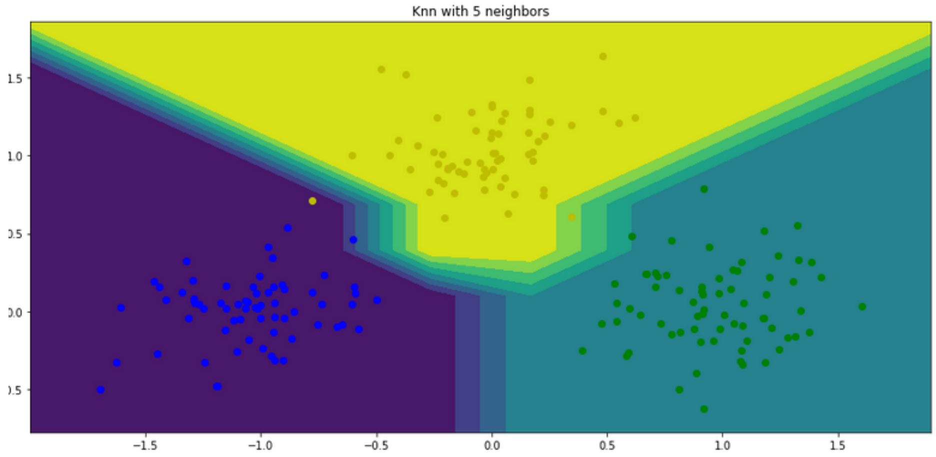
23 mars 2020

1 k-NN classification : Synthetic dataset

Question 1

We look at what happens with the decision boundary when K increases.

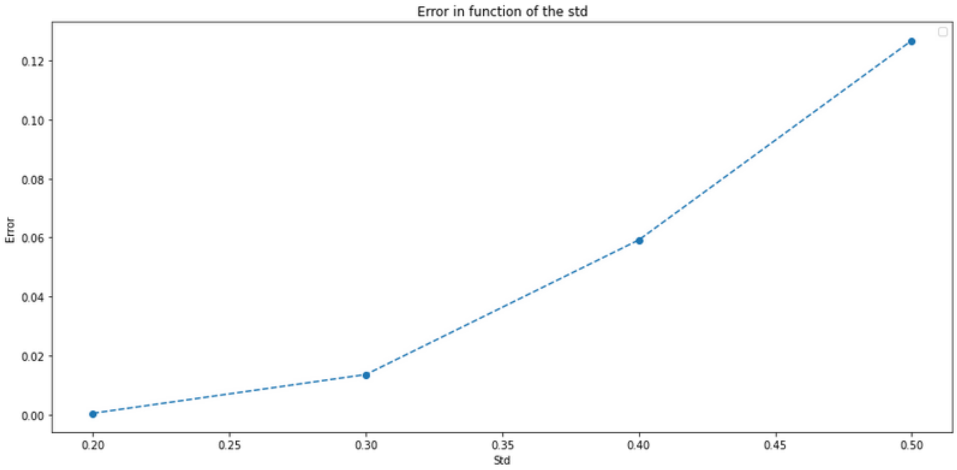

Knn with 2 neighbors

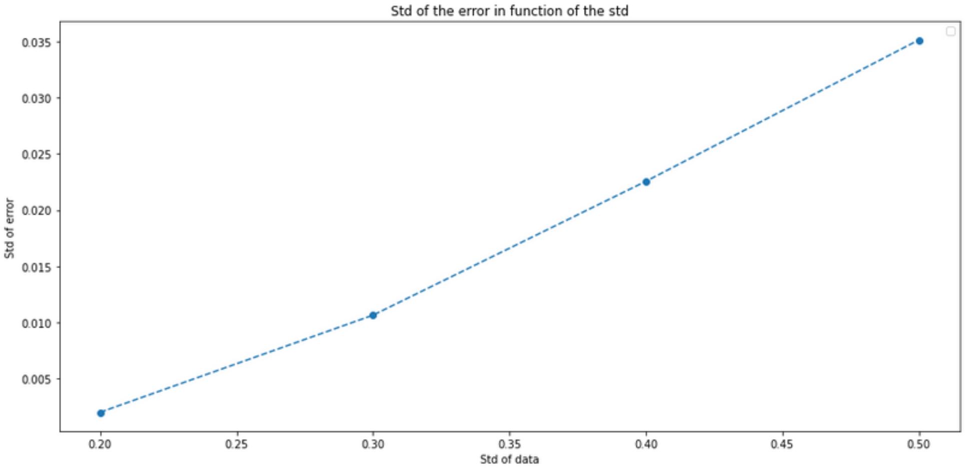

Knn with 5 neighbors

When K increases, we take into account more neighbors, and that make the boundaries more linear. Because we realize a kind of "mean" over several points. The result does not only depend on one point, which could be an outlier.

Question 2

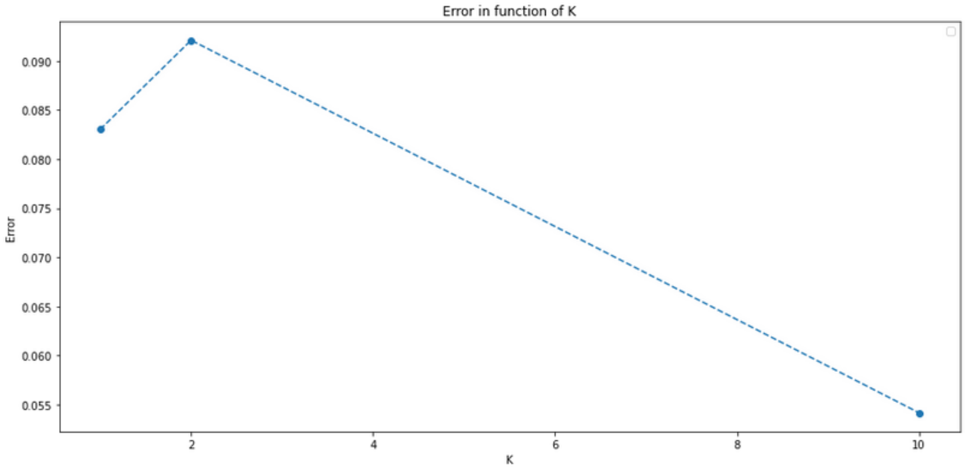
A) K is fixed to one, the standard deviation of our dataset varies.

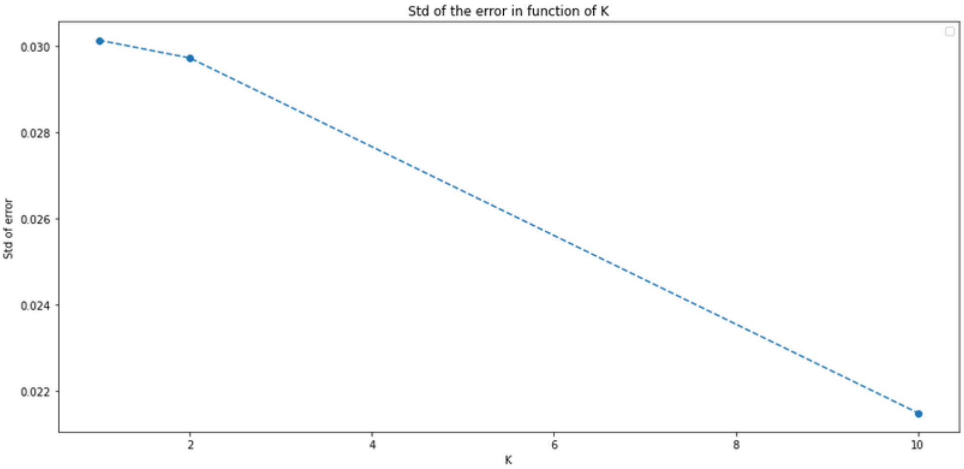




When the std increases, the points are less separable so the error increases also. The std of the error also increases becauses the cases are more diverse

B) Std is fixed, K varies.

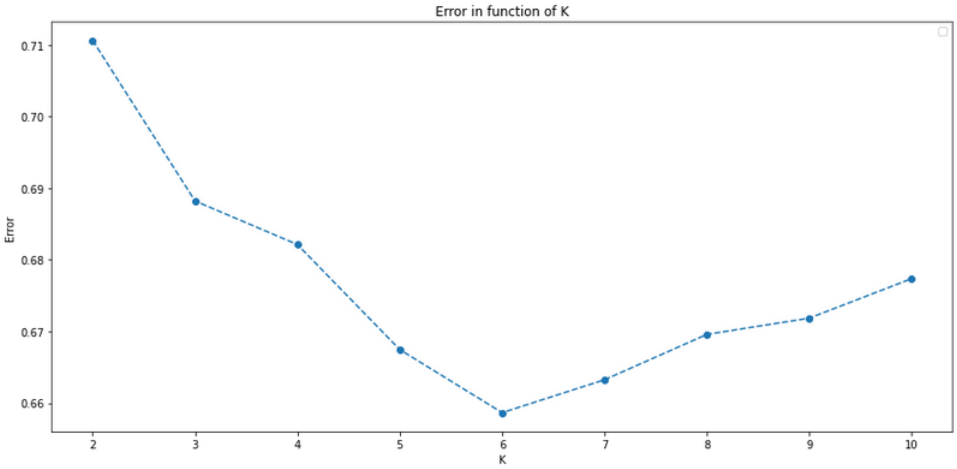




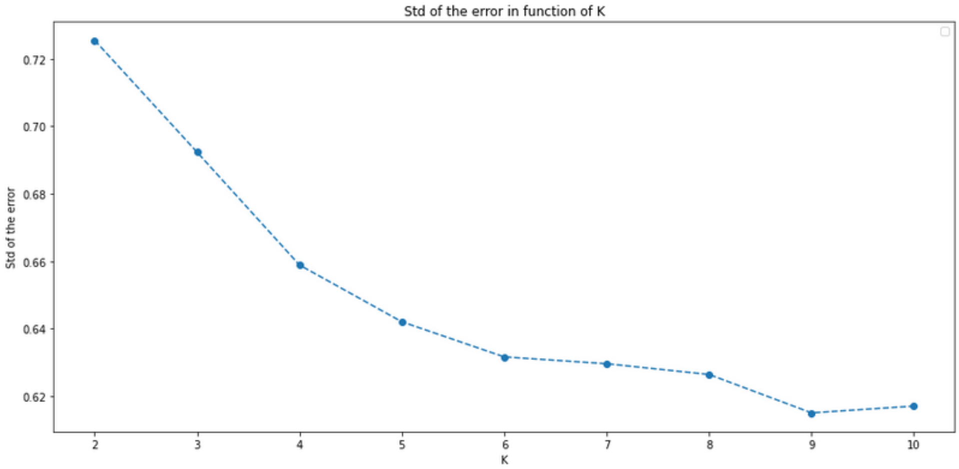
When K increases, we take the mean over more points so the result is more precise and the error decreases. But when K becomes too important, we are going to consider only the majority class, so the error becomes to increase. Conclusion : There is certainly a best K according to our dataset and we must practice cross validation in order too find it. Also, the standard deviation decreases with K in so far as we average over more points.

2 k-NN regression : Szeged-weather dataset

In this exercice, we practice regression and not classification. Therefore, after having found the K-Nearest Neighbors, we do not take the majority class, but we take the mean of the apparent temperatures of those neigbors. Therefore, we have adapted our function « predicted_label » into a function « predicted_label_regression ».



When we look at the error in function of K (K between 2 and 10), we notice that it doesn't vary a lot. It's always between 0.65C and 0.72C. The algorithm takes a very long time to run so we were only able to do it once. We can observe that the error decreases when K goes from 2 to 6 and increases afterwards. So the best K here is 6.

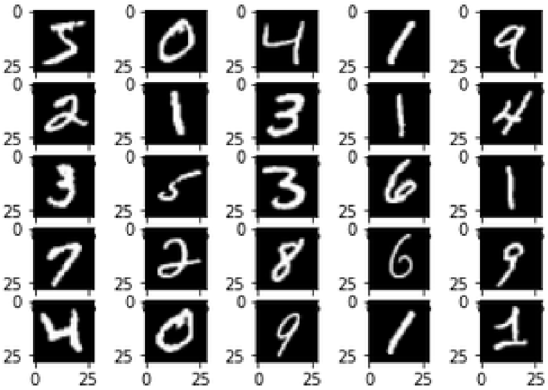


As in the first exercice, the standard deviation of the error decreases with K as we average over more points, which are probably the same if we take a huge number of neigbors.

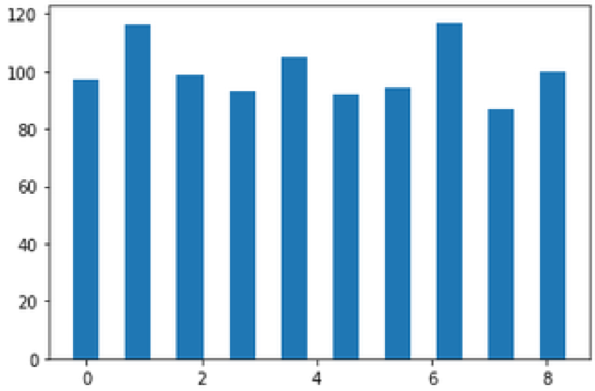
3 k-NN classification : MNIST dataset

Question 1

As in exercice 1, we practice KNN for multiclass classification. We have now 10 classes, that corresponds to the 10 digits. Here is some pieces of data :



Here is an histogram of our labels (for the training set) :

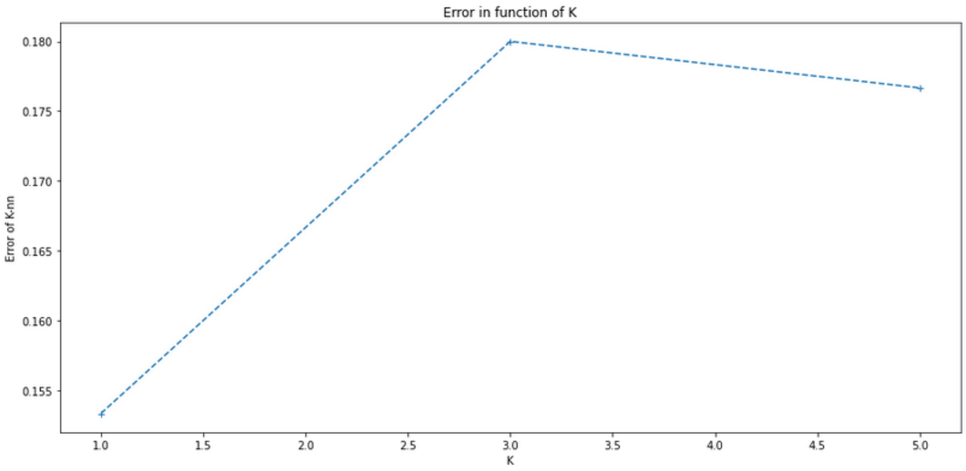


The most represented labels are 1 and 7 in the training set, 1 and 4 in the test set, but the data is distributed almost with a uniform law over (0,9).

Question 2

As for exercice two, the algorithm was very long to run, because the features are simply the pixels and it's very long to compare each pixels from each images, so we were only able to do it once.

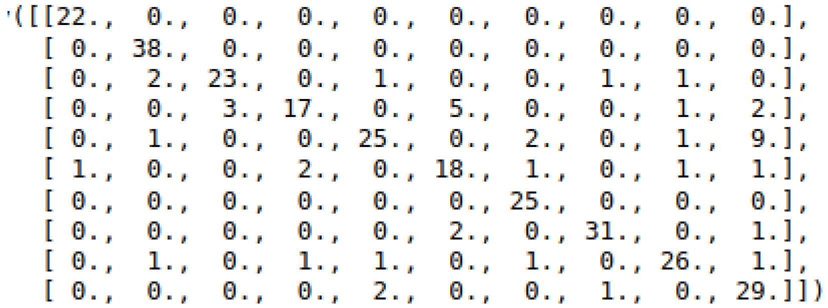
Here is our result :



The error is the best when we consider only one neighbor.

Question 3

Below is an example of a confusion matrix, obtained with K=1. We can notice that when we compare the pixels, the algorithm has a tendency to confuses the 9 and the 4, and also the 5 and the 3.



```
'([[22.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.],
   [ 0., 38.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.],
   [ 0.,  2., 23.,  0.,  1.,  0.,  0.,  1.,  1.,  0.],
   [ 0.,  0.,  3., 17.,  0.,  5.,  0.,  0.,  1.,  2.],
   [ 0.,  1.,  0.,  0., 25.,  0.,  2.,  0.,  1.,  9.],
   [ 1.,  0.,  0.,  2.,  0., 18.,  1.,  0.,  1.,  1.],
   [ 0.,  0.,  0.,  0.,  0.,  0., 25.,  0.,  0.,  0.],
   [ 0.,  0.,  0.,  0.,  0.,  2.,  0., 31.,  0.,  1.],
   [ 0.,  1.,  0.,  1.,  1.,  0.,  1.,  0., 26.,  1.],
   [ 0.,  0.,  0.,  0.,  2.,  0.,  0.,  1.,  0., 29.]])
```

Conclusion

KNN is one of the most simple algorithm of machine learning. It has two parameters : the number K, which musn't be too small and nor too big, and the way we calculate the distances between our data points. We have to test different K in order to find the best one. When K increases, we average over more points so the boundaries between our classes become more linear. When we have a big dataset or a lot of features the algorithm is very long. It's a good algorithm for simple problems, but when the problems get more complex, it doesn't appear to be very efficient.