

# PdP - cahier des charges

---

## Matériaux

3 séries de relevés expérimentaux au format texte concernant les paramètres de neurones ainsi que leur classe (**ech1.txt**, **ech2.txt** et **ech3.txt**)

Des éléments permettant de comprendre l'origine des paramètres utilisés (**info\_parameters.pdf** et **parameters.pdf**)

Des références à Scikit-learn et Python permettant de sélectionner une méthode de classification et de l'utiliser (**Mastering Machine Learning with scikit-learn.pdf**, **Learning scikit-learn\_ Machine Learning in Python.pdf** et **Python Data Science Handbook.pdf**)

## Objectif principal

Fournir un logiciel capable de prédire avec le plus d'efficacité possible la classe d'un neurone en utilisant tout ou partie d'un jeu de valeurs pour les paramètres de classification.

## Objectif secondaire

Fournir une interface fenêtrée (Tkinter ?) permettant de d'utiliser plus simplement le logiciel sans recours à la ligne de commande.

## Démarche

Il s'agit de mettre au point une méthode de classification reposant sur un algorithme dont le choix est laissé à l'appréciation des étudiants. Cela nécessitera à chaque fois 4 étapes :

1. compréhension de la méthode et des éléments de la bibliothèque à utiliser
2. développement d'un script à des fins de test
3. lancement d'ensembles de simulations pour explorer l'effet des paramètres de la méthode et des paramètres expérimentaux employés
4. présentation des résultats

Quand cette démarche aura été effectuée sur les principales méthodes de classification proposées, la meilleure sera retenue et le développement d'une interface d'utilisation sera envisagée.

## Pour commencer ...

Il est suggéré de commencer par une étude de l'effet de la classe sur chacun des paramètres pour se faire une idée de leur influence éventuelle. Cela peut se faire à l'aide de `scipy.stats` et de la bibliothèque `matplotlib` pour la visualisation de résultats. Il est peu probable de trouver qu'un seul ou deux paramètres suffisent à la classification mais cela donne une idée de leur contribution éventuelle.

Ensuite il serait bien de commencer par l'utilisation des **SVM** (car c'est la méthode qui a donné pour le moment les meilleurs résultats, plus de 70%) est ensuite conseillée comme point de départ. Si vous connaissez déjà une méthode pouvant servir à la classification vous pouvez bien entendu commencer avec.

## Références

1. [https://en.wikipedia.org/wiki/Statistical\\_classification](https://en.wikipedia.org/wiki/Statistical_classification)
2. [https://en.wikipedia.org/wiki/Statistical\\_classification#Algorithms](https://en.wikipedia.org/wiki/Statistical_classification#Algorithms)
3. [https://en.wikipedia.org/wiki/Support\\_vector\\_machine](https://en.wikipedia.org/wiki/Support_vector_machine)
4. [http://scikit-learn.org/stable/supervised\\_learning.html](http://scikit-learn.org/stable/supervised_learning.html)
5. <http://scikit-learn.org/stable/modules/tree.html#classification>
6. [http://scikit-learn.org/stable/modules/lda\\_qda.html#dimensionality-reduction-using-linear-discriminant-analysis](http://scikit-learn.org/stable/modules/lda_qda.html#dimensionality-reduction-using-linear-discriminant-analysis)
7. <http://scikit-learn.org/stable/modules/svm.html#classification>
8. <http://scikit-learn.org/stable/modules/sgd.html#classification>
9. <http://scikit-learn.org/stable/modules/neighbors.html#nearest-neighbors-classification>
10. [http://scikit-learn.org/stable/modules/gaussian\\_process.html#gaussian-process-classification-gpc](http://scikit-learn.org/stable/modules/gaussian_process.html#gaussian-process-classification-gpc)
11. [http://scikit-learn.org/stable/modules/neural\\_networks\\_supervised.html#classification](http://scikit-learn.org/stable/modules/neural_networks_supervised.html#classification)