

# IFT603 – 712 Techniques d'apprentissage

## Recommandations pour les travaux pratiques

- `numpy` possède un mécanisme très puissant qui s'appelle le *broadcasting*: <https://numpy.org/doc/stable/user/basics.broadcasting.html>  
Celui-ci permet de traiter les vecteurs ou les matrices comme des scalaires, et vous permet d'éliminer des boucles inutiles et inefficaces.
- N'incluez pas de code superflu, et surtout pas deux solutions au même problème. Choisissez en une. En entreprise, on appelle ça du "*dead code*", c'est une très mauvaise pratique.
- Évitez de laisser de laisser du code en commentaire.
- SVP, n'ajoutez pas d'imports. Les travaux sont conçu pour vous forcer à utiliser les outils qu'on vous fournis, sans plus. Le principe des travaux est de vous faire implémenter des algorithmes d'apprentissage automatique. Si vous utilisez une bibliothèque qui fait le travail à votre place, vous vous privez du processus d'apprentissage.
- Ne supprimez pas les commentaires existants et ne vous gênez pas pour ajouter des commentaires dans votre code. En programmation, les commentaires sont un de vos meilleurs outils, servez vous-en.
- Utilisez un "*linter*" ! La plupart des IDE en possède un, celui-ci permet de formater du code afin de le rendre plus propre et répondant aux standards PEP8. Dans pycharm/intellij, c'est **Ctrl-Alt-L**, tout comme dans Spyder (je crois). J'en ai même un dans vim !
- Le code doit être développé 100% dans gitLab. Pas de code transmis par courriel. Aussi, évitez de pousser du code dans le main car chaque membre de l'équipe doit avoir ses branches et faire de nombreux commits. Attention, un travail avec trop peu de *commits*, trop peu de branches et/ou trop peu de *merges* peut mener à une importante perte de points.
- Pour chaque TP, n'oubliez pas de remettre un fichier txt « **gitLab.txt** » dans lequel vous donnez le lien vers votre repo gitLab. Vous devez également remettre **l'agenda d'équipe** ainsi que le **formulaire d'intégrité** (voir site web du cours).