

# IFT 603-712 : TP1

## Travail par équipe de 2 ou de 3

Remettez votre solution aux numéros 1 à 4 en format PDF ou manuscrit (et scanné) via turnin. Remettez aussi votre solution du numéro 5 via turnin.

→ DATE LIMITE DE REMISE : 6 octobre 2023, 23h59.

1. [1 point] Démontrez la propriété de l'entropie suivante :

$$H[x,y] = H[y|x] + H[x].$$

Vous pouvez faire la démonstration pour le cas discret ou continu.

2. [1.5 points] Démontrez la propriété de l'information mutuelle suivante :

$$I[x,y] = H[x] - H[x|y].$$

Vous pouvez faire la démonstration pour le cas discret ou continu.

3. [1 point] Démontrez la propriété suivante :

$$\text{Cov}[x,y] = E_{xy}[xy] - E[x]E[y],$$

où Cov est la fonction de covariance et E est l'espérance mathématique. Vous pouvez faire la démonstration pour le cas discret ou continu.

4. [0.5 point] Soit une variable aléatoire binaire  $X$  ayant produit la séquence d'observations suivante:  $\{0,0,0,1,0,1,0,0,1\}$ . Partant de ces observations, donnez : (a) la distribution  $P(X)$  également appelée « l'évidence »; (b) l'espérance mathématique de  $X$ ; (c) la variance de  $X$ ; et (d) l'entropie de  $X$ . Précisez vos calculs.

5. [6 points] Programmez les algorithmes de régression linéaire et non linéaire polynomial ainsi que la recherche d'hyperparamètre « k-fold cross-validation ». Pour ce faire, vous devez télécharger le fichier `ift603_tp1_prog.zip` du site web du cours.

L'algorithme doit être implémenté via la classe **Regression**. Votre implémentation de cette classe doit être placée dans le fichier `solution_regression.py` qui contient déjà une ébauche. Veuillez vous référer aux commentaires sous la signature de chaque méthode de la classe **Regression** afin de savoir quoi implémenter. Votre implémentation doit être efficace et utiliser les fonctionnalités de la librairie *Numpy* (e.g. **vous devez donc éviter les boucles for inutiles**). Cependant, le but est aussi d'apprendre à coder vous-mêmes les algorithmes de *machine learning*, donc vous ne devez pas utiliser la librairie *scikit-learn* si cela ne vous ait pas explicitement demandé dans le code à remplir. De plus, vous ne devez pas importer des librairies autres que celles qui sont déjà importées dans le code. Les fonctions qu'il vous faut compléter sont identifiées par le commentaire « AJOUTER CODE ICI ».

**Note 1 :** bien que vide, le code de la classe **Regression** s'exécute déjà. Pour vous en convaincre, vous n'avez qu'à taper la commande suivante dans un terminal linux :

```
python3 regression.py 1 sin 20 20 0.3 10 0.001
```

En tapant :

```
python3 regression.py
```

vous verrez à quoi correspondent les arguments de ce programme.

**Note 2 :** le code des TPs a été testé avec de versions de python 3.8. Pour faire fonctionner votre code sur les ordinateurs du département, vous devez :

1. démarrer une session linux (ubuntu)
2. démarrer un terminal
3. normalement, si vous démarrez une session ipython (tapez *ipython* dans le terminal) vous verrez « Python 3.8.x » (ou plus récent).

**Note 3 :** il est recommandé de rédiger son code dans un *IDE* comme Spyder ou PyCharm.

**Note 4 :** pour exécuter le code des notebooks disponibles sur le site web du cours, vous devez taper la commande « jupyter notebook » dans un terminal.