

**Mini Projet 2 : Devoir à rendre**

Livrables :

Partie du mini projet	Date limite	Document à fournir sur google classroom
Partie 1 et 2	12/11/2023	<ul style="list-style-type: none"><li>- Fichier pdf pour partie 1</li><li>- Deux fichiers ipynb pour partie 2 avec description du code avec des commentaires</li><li>- description du code et résultat obtenu dans un fichier pdf</li></ul>
Partie 3	17/11/2023	<ul style="list-style-type: none"><li>- Fichier ipynb pour partie 3 avec description du code avec des commentaires</li><li>- description du code et résultat obtenu dans un fichier pdf</li><li>- comparaison entre les deux modèle</li></ul>

L'objectif de ce mini – projet est de Construire deux modèles de machine learning pour prédire si un client va abandonner la banque ou non.

Pour répondre à cet objectif, nous allons :

- Faire appel aux concepts de MLOps
- Utiliser le dataset suivant « Churn\_Modelling.csv », voir googleclassroom
- Préparer le dataset
- Créer le modèle, entraîner le modèle, évaluer le modèle
- Essayer ensuite d'augmenter le score du modèle obtenu

**Partie 1 : ce qu'il faut connaître avant l'écriture du code source**

1. Quelles sont les étapes à suivre pour réaliser un projet Machine Learning
2. Quelles sont les pratiques à suivre pour préparer les données tout en se référant au dataset en question. Justifier toute opération à effectuer.
3. Quelle est l'architecture du réseau de neurones à adopter tout en se référant au dataset en question. Dessiner le réseau.
4. Quelles sont les fonctions d'activation à choisir ? justifier les choix
5. Ce que c'est la généralisation d'un modèle ? quelles sont les pratiques à suivre pour assurer ce point ?
6. Ce que c'est l'overfitting et l'underfitting ?
7. Ce que c'est Bias variance tradeoff ?
8. Ce que c'est le k-fold cross validation ?
9. Comment le k-fold cross validation peut résoudre le problème d'overfitting ? et comment peut assurer un Bias variance tradeoff
10. Quelles sont les pratiques à suivre pour améliorer la qualité de chaque modèle

## Partie 2 : création du modèle ANN, compilation, training, évaluation, tuning, prediction

### Dans un fichier App1\_ANN\_Churn\_Modelling, répondre aux questions suivantes :

1. Préparer le dataset
2. Créer le réseau de neurones adéquat
3. Compiler le réseau de neurones en suivant les pratiques qu'il faut
  - Quel optimizer choisir ? justifier le choix
  - Quelle est la fonction de loss à choisir ? justifier la réponse
4. Entraîner le modèle (utiliser 100 epochs)
5. Calculer la matrice de confusion relative au modèle obtenu
6. Mesurer l'accuracy du modèle (score inférieur à 0.84 n'est pas accepté).
7. En utilisant votre modèle, prédire si le client suivant va abandonner ou non la banque :
  - Pays : France
  - Score de crédit : 600
  - Genre : Masculin
  - Âge : 40 ans
  - Durée depuis entrée dans la banque : 3 ans
  - Balance : 60000 €
  - Nombre de produits : 2
  - Carte de crédit ? Oui
  - Membre actif ? : Oui
  - Salaire estimé : 50000 €

### Dans un fichier App2\_ANN\_Churn\_Modelling, répondre aux questions suivantes :

8. Créer / Entraîner/ évaluer en utilisant la technique cross entropy:
  - La création d'une méthode qui retourne le modèle compilé est nécessaire (voir annexe-program1)
  - Utiliser `KerasClassifier` de `scikeras.wrappers` pour créer le modèle  
`model = KerasClassifier(create_model)` #create\_model est la fonction en annexe
  - Utiliser `cross_val_score` de `sklearn.model_selection`
  - Afficher la moyenne et déviation standard des accuracy obtenus
  - Interpréter les résultats obtenus :
    - a) Moyenne
    - b) Variance faible
    - c) Variance élevée

### Dans un fichier App3\_ANN\_Churn\_Modelling, répondre aux questions suivantes :

9. Améliorer la qualité du modèle en trouvant des hyperparamètres meilleurs en utilisant grid search.  
Voir annexe-program2

### Partie 3 : création du modèle Decision Trees from scratch

1. Préparer le dataset
2. En se basant sur les références fournies pour créer un arbre de décision from scratch, créer le modèle adéquat pour connaître si un client va abandonner la banque ou non.
3. Décrire l'algorithme choisi
4. Entraîner le modèle
5. Dessiner l'arbre de décision obtenu
6. Calculer la matrice de confusion relative au modèle obtenu
7. Mesurer l'accuracy du modèle.
8. En utilisant votre modèle, prédire si le client suivant va abandonner ou non la banque :  
Pays : France  
Score de crédit : 600  
Genre : Masculin  
Âge : 40 ans  
Durée depuis entrée dans la banque : 3 ans  
Balance : 60000 €  
Nombre de produits : 2  
Carte de crédit ? Oui  
Membre actif ? : Oui  
Salaire estimé : 50000 €
9. Comparer les scores obtenus avec ce qui a été obtenu avec le réseau de neurones
10. Quelles sont les pratiques à suivre pour améliorer la qualité du modèle

### **Annexe-program1**

```
def create_model():  
    model=Sequential()  
    h_layer1=Dense(...)  
    h_layer2=Dense(...)  
    output_layer=Dense(...)  
    model.add(h_layer1)  
    model.add(h_layer2)  
    model.add(output_layer)  
    model.compile(...)  
    return model
```

### **Annexe – prgram2**

```
model = KerasClassifier(create_model)  
from sklearn.model_selection import GridSearchCV  
parameters={"batch_size":[25,30],"epochs":[100,500],"optimizer":["adam","sgd"]}  
grid_search=GridSearchCV(estimator=model,param_grid=parameters,scoring="accuracy",cv=10)  
best_params= grid_search.best_params_  
best_score= grid_search.best_score_
```