Projet Machine Learning

Consignes & Évaluation

Objectif & Données

Objectif

Résoudre un problème de **Régression** ou de **Classification** en adoptant une démarche scientifique rigoureuse.

Attentes :

- Performances du modèle.
- **Justification** des choix techniques.
- Analyse critique des résultats.
- Code de qualité professionnelle.

Données : Choix libre (ex : Kaggle).

1. Dépôt GitHub

- Hébergé sur un dépôt public ou privé (m'inviter).
- **Travail d'équipe :** L'historique des commits doit refléter la participation de **tous**.
- **Structure du Code :**
  - **EDA :** Notebooks (`.ipynb`) autorisés *uniquement* pour l'exploration.
  - **Pipeline ML :** Scripts Python (`.py`) modulaires obligatoires pour l'entraînement.
  - *Interdit :* Entraînement final dans un notebook.
- **Qualité :** PEP8, commentaires, `requirements.txt` ou `conda`.

2. Rapport

- Format PDF, maximum 20 pages.
- **Contenu :**
  - Analyse des données.
  - Choix des méthodes.
  - Résultats et interprétations.

Comparaison d'Approches

Vous devez comparer au moins **deux** approches distinctes :

1. **Approche "Classique" :**
   - Scikit-Learn, XGBoost, LightGBM...
2. **Approche "Deep Learning" :**
   - Réseau de neurones (PyTorch, Keras/TensorFlow, JAX).

\+ **Composante Personnalisée ("Custom Implementation") :**

- Une partie personnalisée (Loss spécifique, métrique custom, couche réseau, algo optimiseur...).

Rigueur Scientifique

- **Prétraitement :** Nettoyage, gestion NaN, encodage justifiés.
- **Validation :** Stratégie robuste impérative (Cross-Validation, Stratified K-Fold).
- **Optimisation :** Recherche d'hyperparamètres (GridSearch, RandomSearch, Optuna) obligatoire.
- **Intégrité :** Attention absolue au **Data Leakage**. Séparation stricte Train / Validation / Test.

Plagiat et IA

Attention

- **Code Internet :** Ne pas recopier aveuglément. Les outils de détection seront utilisés. Plagiat = **0**.
- **IA Générative :** Utilisation aveugle interdite. Code non maîtrisé ou copié tel quel = Plagiat.

La pire approche serait d'appliquer "à l'aveugle" les fonctions de sklearn. On attend une analyse critique.

Barème (sur 20 points)

Critère	Pts	Détails
Rapport	4	Clarté, synthèse, analyse critique.
Code & Git	4	Modulaire (`.py`), propre, commits réguliers.
Méthode Classique	2	Implémentation correcte, justifiée.
Deep Learning	2	Architecture pertinente, analyse apprentissage.
"Custom impl."	2	Complexité et pertinence.
Prétraitement	2	Nettoyage et Feature Engineering.
Démarche Scient.	4	Validation croisée, hyperparamètres, rigueur.

Pénalités (Cumulatives)

Sanctions

- **Plagiat / Copie :** Note finale **0/20**.
- **Pas de Git (ou upload unique) : -10 points**.
- **Rendu Notebook (hors EDA) : -10 points**.
- **Manque de participation : -5 points** (individuel).