Projet d'Apprentissage Automatique

Joris LIMONIER

S1.2 - 2023/2024

(version du 31 janvier 2024)

1 Introduction

Je vous demande de lire **attentivement** les règles, puis de réaliser les tâches ci-dessous.

2 Règles & Conseils

2.1 Règles

— Date limite : **dimanche 03 mars à 20h00**.
— Mettez tous vos fichiers dans un dossier

`projet_<cursus>_<nom>_<prenom>`

(par exemple `projet_analyste_limonier_joris`). A l'intérieur de ce dossier, créez un dossier `src` contenant vos fichiers `.py` et un dossier `notebooks` contenant vos fichiers `.ipynb`.
Vous pouvez écrire vos interprétations dans des cellules Markdown du Notebook ou dans un rapport. Si vous choisissez d'écrire un rapport, créez un dossier `reports` contenant votre rapport au format `.pdf`.
Un exemple de structure du dossier `projet_analyste_limonier_joris` est donné ci-dessous :

```
src/
    predict.py
    train.py
notebooks/
    analyse.ipynb
    optimisation.ipynb
reports/
    rapport.pdf
```

— Vous devez travailler seul.
— Toutes les ressources sont autorisées (cours, ChatGPT, Copilot, etc.), ce qui implique qu'**une grande partie de la note portera sur l'explication et l'interprétation de vos résultats.** En particulier, tout copier-coller de code n'aura de valeur que votre explication et interprétation.
— Expliquez tout ce que vous faites, et interprétez vos résultats.
— Vous devez utiliser Python
— **Les questions sont plutôt des suggestions**. Si vous voulez explorer quelque chose de différent, faites-le. Toutes les questions ne s'appliquent pas à tous les jeux de données, soyez pragmatiques mais **justifiez vos choix**.
— Notation :
  — **Bonus** : jusqu'à 2 points au cas par cas, par exemple pour une analyse particulièrement intéressante, un travail particulièrement poussé ou encore une interprétation particulièrement pertinente.
  — **Malus** : jusqu'à 2 points au cas par cas pour tout ce qui fait perdre du temps inutilement ou n'est pas propre. Seront notamment pénalisés : du code ne respectant pas les conventions Python, pas commenté, pas exécuté, ou encore répété plutôt que d'utiliser des fonctions, des erreurs d'orthographes, des fichiers non nommés correctement, etc.
  — **Retard** Chaque tranche de 24h de retard retirera 5 points. Tout retard, même d'une minute, est un retard.
— Un travail plus approfondi est d'attendu des étudiants du cursus Scientifique.
— Chaque partie précise les étudiants qui doivent l'effectuer.
— Envoyez vos questions à joris.limonier.int@groupe-gema.com

2.2 Conseils

— Commencez tôt.
— Rendez le devoir à l'heure. Vous avez un mois pour faire ce devoir, essayez de ne pas le rendre en retard s'il vous plaît. Rendez le devoir plusieurs heures ou jours en avance pour éviter la panne de wifi à la denière minute, car il n'y aura aucune indulgence.
— Commencez simplement par faire une première analyse, puis procédez par itération pour améliorer votre résultat. Par exemple, commencez par un modèle basique, sans optimisation, puis ajoutez de l'optimisation, puis ajoutez de l'ingénierie des caractéristiques, etc. Faites ensuite un tableau comparatif de l'apport de chaque brique et interprétez les résultats.
— Ne vous limitez pas à ce qui a été vu en cours si vous le pouvez.
— Vous pouvez écrire vos fonctions et classes dans un ou plusieurs fichiers `.py`, puis les appeler dans un fichier `.ipynb` pour éviter d'avoir un notebook à rallonge.

3 Questions

3.1 Préparation (Cursus Analystes et cursus Scientifique)

1. **Chargement** - Trouver (par exemple sur Kaggle) et proposer un jeu de données de **classification** de 1k - 100k lignes, qui n'a été pris par aucun autre étudiant.
2. **Publication** - Publiez le sur la même feuille Google Sheets que le TD 6 (voir flux Classroom).
3. **Validation** - Attendre la validation du professeur. Si le jeu de données est refusé, vous devrez en trouver un autre. Si je ne réponds pas dans les 2 jours ouvrés, vous pouvez me relancer par courriel.
   Si vous décidez de changer de jeu de données, il faudra à nouveau attendre ma validation.
4. **Découverte** - Décrire le jeu de données
   — Que représente ce jeu de données ?
   — Que représentent les colonnes ?
5. **Analyse** - Analyser les colonnes présentes. Cela peut inclure (mais n'est pas limité à) :
   — Visualisation
   — Corrélation
   — Type
   — Gestion des valeurs manquantes
6. **Préparation.** Préparer les données pour l'entrainement. Par exemple, cela peut inclure :
   — Evaluation de l'utilité des colonnes
   — Encodage des variables catégorielles
   — Gestions des données textuelles
   — Normalisation des variables numériques

3.2 Groupement (Cursus Scientifique uniquement)

Dans cette section, nous nous plaçons dans le cas où nous n'avons pas accès à la variable cible. Nous allons donc chercher à regrouper les données en groupes homogènes.
Notez que la variable cible ne peut pas être utilisée pour le groupement (problème non supervisé), mais peut être utilisée pour l'analyse des résultats, notamment pour le calcul des métriques de performance (rappel, précision, etc.). Cela reproduit la situation d'un problème réel non supervisé, mais où des experts métiers peuvent annoter quelques données pour évaluer la qualité du groupement.
Dans la section 3.3, nous reproduirons les mêmes étapes mais en utilisant la variable cible pour entrainer un modèle de classification. Nous comparerons les résultats obtenus dans les deux cas pour mesurer l'avantage de disposer de la variable cible.
Dans la présente section, je vous invite à effectuer les étapes de la section 3.3, en les adaptant à un problème non supervisé.

3.3 Classification (Cursus Analystes et cursus Scientifique)

1. **Entrainement.** Entrainement d'un modèle de classification :
   — Séparation des données d'enrainement, de validation et de test
   — Entrainement du modèle
   — Evaluation du modèle
2. **Optimisation.** Optimisation du modèle :
   — Optimisation des hyperparamètres
   — Ingénierie des caractéristiques ("feature engineering")