TP3 - Classification multiclasses : légumes secs

Remise (code et rapport) le lundi 18 avril sur Moodle pour tous les groupes.

Consignes

- Le devoir doit être fait par groupe de 2 au maximum. Il est fortement recommandé d'être 2.
- Inscrivez vous à la plate-forme Kaggle suivante [1].
- Vous devrez remettre le fichier `my_bean_tester.py` ainsi que votre rapport au format pdf (matricule1_matricule2_TP3.pdf) à la racine d'un dossier compressé (matricule1_matricule2_TP3.zip).
- Indiquez vos nom et matricules dans le fichier PDF et en commentaires en haut du fichier de code soumis.
- Toutes les consignes générales du cours (interdiction de plagiat, etc.) s'appliquent pour ce devoir.

1 Introduction

Les légumes secs sont des graines qui composent une base de l'alimentation humaine dans de nombreuses régions du monde. Généralement associés à une alimentation saine, ils représentent une source d'apport en glucides, protéines, fibres et minéraux essentiels comme le fer et le calcium. Ils sont de fait facilement recommandés pour leur qualités nutritionnelles, notamment par le 'Guide alimentaire canadien' officiel, en plus d'être associés à un impact écologique réduit comparé à d'autres alternatives. L'industrie des légumineuses est en croissance et les producteurs s'intéressent au problème de différentiation des nombreux types de haricots qui existent dans une optique de qualité de production, certains haricots s'adaptant plus ou moins bien à différents types de sols, de climats, etc.

L'objectif de cet exercice est de déduire la variété d'un haricot sec sur la base de données extraites de photos. Comme vous êtes dans un cours d'intelligence artificielle et non d'agriculture, vous préférerez déléguer cette tâche à un agent intelligent plutôt qu'à vos propres yeux.

2 Énoncé

Pour ce devoir, vous participerez à un défi *Kaggle* dont le but est de développer une approche d'apprentissage automatique (ML) pour déterminer l'espèce d'un haricot (Seker, Barbunya, Bombay, Cali, Dermosan, Horoz ou Sira). Vous trouverez sur la page de la compétition Kaggle (voir la note en bas de page) les fichiers à télécharger, la procédure pour faire une soumission, et un tableau *leaderboard* pour voir votre score et votre rang parmi les autres groupes du cours. Vous êtes libre d'utiliser les méthodes et les bibliothèques de ML de votre choix.

L'évaluation se fera sur un ensemble test privé différent de celui sur Kaggle mais avec les mêmes propriétés statistiques. Ainsi, le *leaderboard* vous indique *grossièrement* votre performance sans être parfaitement exacte. Par ailleurs, les scores privés seront mesurés après avoir entrainé votre modèle sur le fichier d'entrainement fournis dans Kaggle. C'est pourquoi il est important de d'assurer que votre solution soit reproductible.

Le concours est disponible à l'adresse suivante [2].

1. https://www.kaggle.com/t/3703bd10c9e147e7b652c9f1abaf0e23
2. https://www.kaggle.com/c/tp3-inf8215-h22

⚠**Vous devez implémenter votre solution en python 3 et vos résultats doivent être reproductibles (plus de détails à ce sujet dans la Section 5.2.**

3 Rapport

En plus d'implémenter votre méthode, vous devez rédiger un rapport **succinct** qui détaille votre méthodologie pour résoudre ce problème et fournir les résultats de votre méthode. Précisément, votre rapport doit contenir au *minimum* les informations suivantes :

1. Titre du projet
2. Nom de l'équipe sur Kaggle, ainsi que la liste des membres de l'équipe (nom complet et matricule)
3. Prétraitement des attributs (*Feature design*) : décrivez et justifiez vos étapes de prétraitement des attributs et indiquez ceux que vous avez sélectionnés dans votre modèle.
4. Méthodologie : Décrivez et justifiez toutes les décisions concernant la répartition des données en ensemble d'entraînement et de validation, ainsi que les techniques utilisées pour gérer le déséquilibre entre les classes (*Unbalanced data*), la stratégie de régularisation, le réglage des hyperparamètres, etc. Ajoutez également toutes les informations que vous jugez nécessaires pour la compréhension de votre modèle (type de réseau, nombre de couches, etc.).
5. Résultats : Présentez une analyse détaillée de vos résultats à l'aide de tableaux ou de graphiques.
6. Discussion : Commentez vos résultats et indiquez quels sont les avantages et les inconvénients de votre approche et de votre méthodologie.
7. Références (si applicable).

Le rapport ne doit pas dépasser 3 pages et doit être rédigé sur une ou deux colonnes à simple interligne, avec une police de caractères de 10 points ou plus (des pages supplémentaires pour les références et le contenu bibliographique sont autorisées). Vous êtes libre de structurer le rapport comme vous le souhaitez tant que vous incluez les éléments mentionnés précédemment. Les sections d'introduction, de description du problème, et de conclusion ne sont pas obligatoires.

4 Ressources fournies

Vous avez accès à 3 fichiers :

— `beans_train.csv` : les données que vous devez utiliser pour entraîner votre modèle.
— `test_public.csv` : les données pour lesquelles vous devez prédire l'espèce de haricot sec. Ce fichier test est aussi utilisé par Kaggle pour votre classement public dans l'onglet *leaderboard*.
— `sample_submission.csv` : un example de fichier à soumettre pour obtenir votre classement.

Pour cette classification à plusieurs classes, vous avez accès à 16 attributs pour chaque haricot dont son périmètre, son aire et différents facteurs de forme extraits automatiquement de photos. Vous trouverez les fichiers code suivant dans Moodle :

— `main.py` **[Ne pas modifier]** : Utilisez ce fichier pour lancer une expérience avec votre modèle.
— `bean_testers.py` **[Ne pas modifier]** : fichier contenant la classe abstraite d'un modèle, ainsi qu'un modèle aléatoire. Votre modèle devra être au minimum supérieur à celui-ci.
— `my_bean_tester.py` : fichier contenant votre modèle de prédiction. Vous devez modifier les méthodes `__init__()` `train()` et `predict()` de la classe `MyBeanTester`.

5 Format

5.1 Compétition Kaggle

Pour la compétition dans Kaggle, vous devez soumettre vos prédictions en utilisant le même format que celui qui se trouve dans `sample_submission.csv`, c'est-à-dire un fichier csv avec deux colonnes et une ligne d'entête : `ID` et `class`. La commande `python main.py` expliquée ci-dessous produira un fichier de prédiction avec le bon format. Il vous suffira d'uploader le fichier csv automatiquement généré dans Kaggle.

5.2 Code et rapport dans Moodle

Lors de votre soumission sur Moodle, donnez votre code (`my_bean_tester.py`) ainsi que votre rapport (matricule1_matricule2_TP3.pdf) à la racine d'un dossier compressé (matricule1_matricule2_TP3.zip).

En utilisant le fichier `main.py` fournis dans Moodle votre solution doit pouvoir être utilisée comme ceci :

```
python main.py (--train_file chemin/vers/train.csv)
               (--test_file chemin/vers/test_public.csv)
               (--prediction_file chemin/vers/submission.csv)
```

Les paramètres sont optionnels et sont par defaut assigné à `./data/beans_train.csv`, `./data/test_public.csv`, et `./data/predictions.csv` respectivement. Cette commande va :

1. appeler la fonction `train()` sur votre `MyBeanTester` pour entrainer votre modèle sur le fichier indiqué par `--train_file`.
2. appeler la fonction `predict()` sur votre `MyBeanTester` pour prédire des résultats sur le fichier indiqué par `--test_file`.
3. sauvegarder les prédictions dans un fichier indiqué par `--prediction_file`. Ce fichier peut être uploadé dans Kaggle tel quel.

6 Critères d'évaluation

Les points seront attribué selon la répartition suivante : 10 points pour la performance de votre modèle sur l'ensemble de test privé, et 10 points pour le rapport écrit.

L'équation suivante sera utilisée pour votre performance au concours Kaggle :

$$
\begin{aligned}
\texttt{points} &= 0 && \text{if } P \leq \texttt{RandomBaseline} \\
\texttt{points} &= 7 \times \frac{P - \texttt{RandomBaseline}}{\texttt{WeakBaseline} - \texttt{RandomBaseline}} && \text{if } \texttt{RandomBaseline} < P \leq \texttt{WeakBaseline} \\
\texttt{points} &= 7 + 3 \times \frac{P - \texttt{WeakBaseline}}{\texttt{StrongBaseline} - \texttt{WeakBaseline}} && \text{if } \texttt{WeakBaseline} < P \leq \texttt{StrongBaseline} \\
\texttt{points} &= 10 && \text{if } P \geq \texttt{StrongBaseline}
\end{aligned}
$$

où P, `RandomBaseline`, `WeakBaseline` et `StrongBaseline` sont les performances de votre modèle, d'une prédiction aléatoire (0.16 pour ces données), d'un modèle peu précis (0.6 pour ces données), et d'un modèle plus performant (0.9 pour ces données) respectivement. En particulier, être moins qu'une méthode

aléatoire ne vous rapporte aucun point, battre le modèle peu précis vous assure d'avoir au moins 7/10, et battre le modèle plus performant vous donne la totalité des points pour le code. **De plus, un bonus de +1 sur la note finale du cours (sur 20) sera attribué au meilleur groupe**.

⚠**Malgré l'aspect *concours*, votre classement n'a aucune influence sur vos points pour ce TP** (sauf pour le bonus).

Pour le rapport écrit, les critères d'évaluation comprennent :

- La cohérence de votre méthodologie (pré-traitement, sélection des caractéristiques, validation, algorithmes utilisés, etc.).
- La rigueur technique de la description de vos algorithmes.
- La clarté générale du rapport et de ses sections.
- La qualité des descriptions, des graphiques, des figures et des tableaux. Par exemple, n'oubliez pas d'indiquer à quoi correspondent les axes et de mettre les légendes des figures. Expliquez également les résultats des figures dans le test principal.
- La qualité de l'analyse des résultats finaux et intermédiaires.
- L'organisation générale et la qualité rédactionnelle.

7 Conseils

- Utilisez des librairies déjà faites et facile à utiliser tel que scikit-learn et keras. Tensorflow et Pytorch ne sont pas recommandé si vous êtes débutant en machine learning.
- Vérifiez si les classes sont équilibrées dans l'ensemble d'entrainement.
- Vérifiez si certains attributs ne sont pas redondant ou n'apportent aucune information.
- Explorez différentes valeurs pour les hyperparamètres avec *random search* ou *grid search*.
- Réservez une partie de votre ensemble d'entrainement pour un ensemble de validation.
- Vérifiez votre performance avec la technique du *Cross Validation* et/ou *Early Stopping*.
- Énormément de ressources sont présentes sur le web. N'hésitez pas à vous en inspirer. Etre capable de trouver par vous même des informations est également une compétence visée dans ce TP.
- **Ne passez pas un temps démesuré à améliorer votre modèle.** Passer 10 heures pour augmenter votre prédiction de 1% ne vaut peu être pas le coup, en considérant que vous avez d'autres cours à travailler :-)