Réalisation d'une interface d'analyse de données par apprentissage supervisé

Machine Learning sous Python

Master 2 SISE
Université Lumière Lyon 2

DARANKOUM Davy, PERENON Clément, LE GALEZE Pierre
clement.perenon@gmail.com
pierre.legaleze@gmail.com
davydarankoum6@gmail.com
GitHub

5 décembre 2021

Résumé

Réalisation d'une interface d'analyse de données sous Python par apprentissage supervisé avec une sélection des hyper-paramètres automatisés.

Table des matières

1 Introduction

1.1 Objectifs du projet

Ce projet s'intègre dans le cadre de notre cours de Machine Learning sous Python. Il doit permettre de réaliser une interface graphique ayant pour objectif d'appliquer des algorithmes de prédictions sur un jeu de données. L'outil doit pouvoir être aisément utilisé par des personnes non initiées à la programmation Python.

1.2 Cahier des charges

- L'outil prendra en entrée un jeu de données annoté au format csv (avec une virgule pour séparateur). Le jeu de données comportera des noms de colonnes à la première ligne.
- L'application permettra à l'utilisateur de définir les variables cibles et les variables explicatives.
- En fonction du type de la variable cible (catégorielle ou numérique), l'application proposera au moins 3 algorithmes de classification, ou 3 algorithmes de régression que l'utilisateur peut appliquer.
- L'utilisateur pourra choisir un ou plusieurs de ces algorithmes à appliquer aux données. Pour chaque algorithme à appliquer, les valeurs optimales des principaux hyper-paramètres pourront être soit définies par l'utilisateur, soit identifiées automatiquement.
- Pour chacun des modèles à appliquer, on utilisera une validation croisée et on fournira en output les métriques d'évaluation, le temps de calcul, et surtout une/des figure(s) permettant de percevoir de façon synthétique la différence entre les prédictions et la réalité.
- Vous devrez utiliser l'une des deux bibliothèques Dash ou Bokeh pour la conception de l'interface graphique et y intégrer les visualisations à l'aide de Plotly (pour les rendre interactives)
- Livrable et calendrier : Rapport et code source commenté, pour le dimanche 5 décembre et d'une soutenance la semaine suivante.

1.3 Package utilisé

En ce qui concerne la manipulation de données en règle générale nous utiliserons Pandas et parfois Numpy. L'affichage des graphiques sera réalisé avec PlotLy car nous souhaitons que ces derniers soient interactifs. De plus, la bibliothèque PlotLy fonctionne très bien avec Dash que nous avons sélectionné pour réaliser l'interface graphique.

Pour l'interface graphique, nous avons opté pour Dash. Dash semblait, après une étude des deux technologies, plus simple à prendre en main et parfaitement adapté à nos besoins.

Nous avons fait le choix pour les algorithmes de machine learning d'utiliser le package sklearn. Le but de ce projet n'étant pas de développer nous-mêmes les algorithmes, la fiabilité et la performance des algorithmes de sklearn s'adaptaient parfaitement à notre projet.

1.4 Principe du code Python

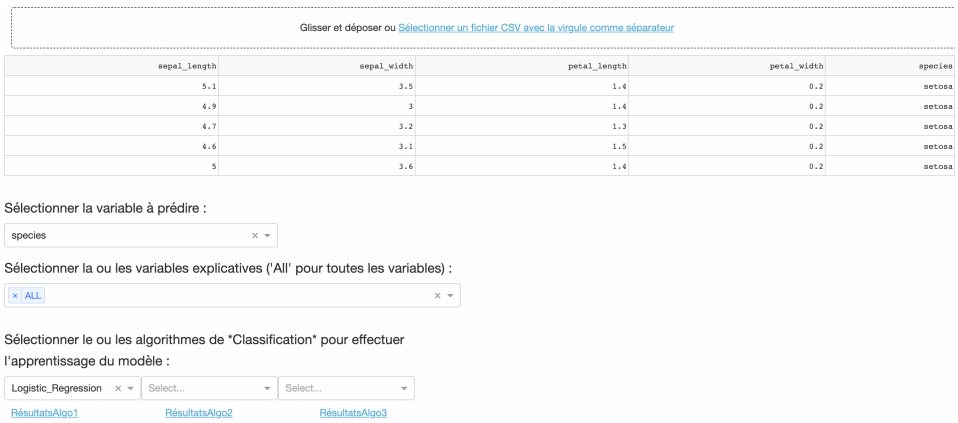
La fonction permettant de faire les analyses du jeu de données est divisée en 4 parties : l'initialisation des paramètres du modèle sélectionné par l'utilisateur, le réglage des hyper-paramètres avec GridSearch, l'entrainement/test du modèle par cross-validation et enfin, l'affichage des résultats.

2 Guide d'utilisation

2.1 Page d'accueil

Une fois sur la page d'accueil, l'utilisateur va dans un premier temps sélectionner son jeu de donnée (avec une "," comme séparateur) en cliquant sur le bouton "Drag and Drop or Select files". Une fois le jeu de donnée chargé, l'utilisateur aura la possibilité de choisir via la liste des colonnes, la variable à prédire ainsi que les variables explicatives.

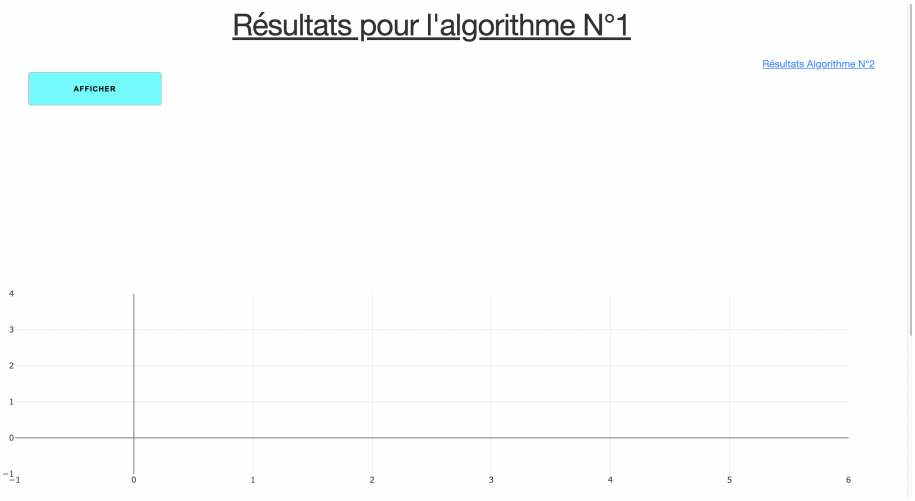
Ensuite, en fonction du type de la variable à prédire, les modèles que l'utilisateur pourra choisir seront soit des modèles de régression ou de classification. L'utilisateur a le choix entre 3 modèles de chaque type, si celui-ci souhaite n'obtenir les résultats que d'un seul, il lui suffit de le sélectionner dans la liste puis de cliquer sur le bouton "ResultatsAlgo1" (pour la première liste) située en dessous, une nouvelle fenêtre s'ouvre alors avec les résultats du modèle sélectionné. Par contre, si ce dernier aimerait avoir les résultats de chaque algorithme, il faut qu'il sélectionne un modèle par liste et qu'il appuie sur les 3 boutons "ResultatsAlgo1", "ResultatsAlgo2", "ResultatsAlgo3".



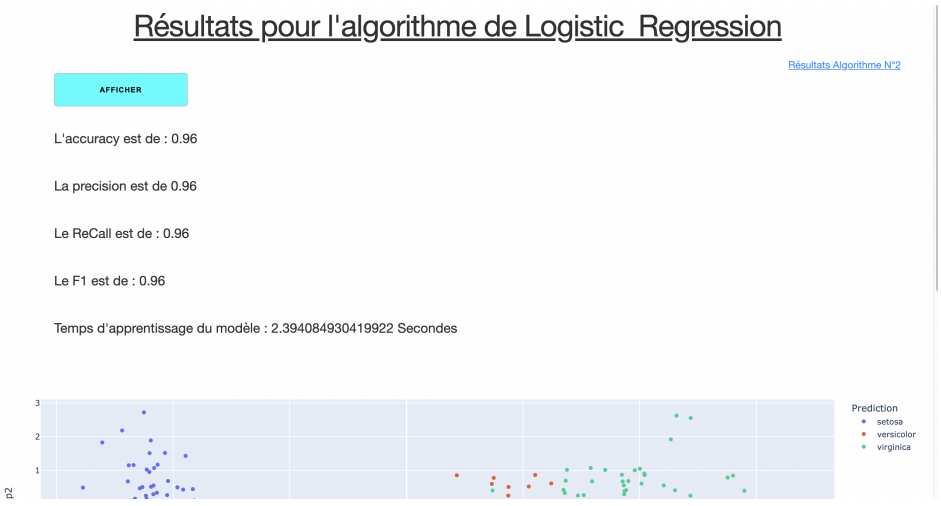
2.2 Page de résultat

Une fois sur la page de résultat, il suffit d'appuyer sur afficher pour obtenir les différentes métriques d'évaluation ainsi que le ou les graphiques correspondants. Si l'utilisateur a sélectionné plusieurs algorithmes, il peut ouvrir leurs pages en appuyant sur le bouton "Résultats Algorithme N°.." situer en haut à droite.

Sans avoir appuyé sur Afficher :



Après avoir appuyé sur Afficher :



3 Présentation de l'architecture de l'application

3.1 Affichage et utilisation des outils "layout"

Les layouts sont de petits objets avec deux fonctions principales : façonner le visuel de l'application et décrire l'interactivité de l'application.

Nous avons un premier layout qui permet d'importer le jeu de données puis un second qui va permettre d'afficher les 5 premières lignes du jeu de données pour vérifier si l'importation est effectuée avec succès.

Les layouts restant sont destinés à sélectionner la variable cible, la ou les variables explicatives, ainsi que choisir l'algorithme de machine learning désiré pour l'analyse. Ce seront des listes déroulantes remplie dynamiquement par les algorithmes et les callback que nous expliquerons dans la partie suivante 3.2 page 7.

3.2 Callback et fonctions

Les callbacks sont les éléments clés qui permettent l'interactivité de l'application. Ce sont des fonctions Python qui sont automatiquement appelées à chaque fois que la propriété d'un composant d'entrée change. Ils sont composés d'entrée, de sortie et parfois si besoin d'état.

L'application est composée de 4 couples de callback/fonctions :

- update_output qui va veiller à importer le jeu de données, afficher quelques lignes du dataframe, alimenter le nom de colonnes et afficher la première liste déroulante pour choisir la variable cible.
- update_dropdown va mettre à jour la liste déroulante suivante qui permet de choisir les variables explicatives. Elle prend en entrée le nom des colonnes privée de la variable cible.
- set_good_model va permettre d'interpréter la variable cible sélectionnée et, en fonction de son type (catégorielle ou numérique), de mettre à jour la liste des algorithmes adaptés à son traitement prédictif.

- get_model est la plus grosse fonction de l'application : c'est cette fonction qui va permettre l'exécution des algorithmes, la gestion de Gridsearch, l'affichage des graphiques, de la matrice de confusion, etc ...

4 Modélisation graphique et interprétation des données en sortie

4.1 Classification

Dans le cas où le type de la variable cible implique une classification, il sera affiché :

- Deux graphiques basés sur les deux premiers composants d'une Analyse en Composantes Principales (ACP), l'un utilisant les prédictions du modèle et l'autre les vraies valeurs, ainsi, l'utilisateur pourra comparer visuellement les performances de l'algorithme.
- La précision
- L'exactitude (accuracy)
- Le rappel
- Le F1 score

4.2 Régression

Si cela implique une régression :

- Un scatter plot montrant la proximité entre les vraies variables et celles prédites
- Le carré moyen des erreurs (MSE)
- L'erreur absolue moyenne (MAE)
- L'erreur quadratique moyenne (RMSE)

5 Pistes d'améliorations et projets d'évolutions futures

Il serait intéressant de permettre un mode "avancé" comprenant :

- Permettre à l'utilisateur de choisir lui-même les hyper-paramètres
- Implémenter des fonctionnalités pour améliorer la précision de certains algorithmes (Boosting, RandomForest ...)
- La prise en charge de plus de type de fichier et pourquoi pas un outil permettant de pré-traiter le jeu de données (nettoyer les variables, modifier le typage, recoder des variables...)
- Se connecter à une base de données, un site, un serveur pour alimenter l'application