Mini-projet Data Mining : US Adult Income
Classification supervisée

1. Objectif

L'objectif de ce mini-projet est la création de trois modèles de classification supervisée, en appliquant les principes de l'apprentissage automatique sur un ensemble de données (Data Set).

La création d'un modèle de classification est un processus impliquant un ensemble d'étapes, à savoir : compréhension des données (data understanding), préparation de données, création et validation de modèles et finalement l'utilisation du modèle.

En réalisant ce mini-projet, vous allez ainsi mettre en pratique l'ensemble des principes théoriques de ce processus en utilisant le langage **Python** et son écosystème (ensemble de packages) dédié à la data science et l'apprentissage automatique (machine learning).

2. Organisation et déroulement

- Le mini-projet doit être réalisé sous forme d'un **Notebook** sous **Jupyter Notebook**.
- La remise des projets aura lieu au plus tard dimanche **31 Janvier 2021**.
- Le **Notebook** (le fichier portant l'extension **.ipynb**) est le seul fichier à remettre.
- Vous n'êtes pas demandés d'élaborer un rapport pour ce mini-projet. En fait, le Notebook fait office du rapport.

3. Ensemble des données (Data Set)

L'ensemble des données que vous allez utiliser au cours de ce mini-projet est nommé « **US Adult Income** ». C'est un ensemble d'informations à propos de 48842 citoyens américains récupérées dans le cadre d'un recensement de la population américaines en 1994.

L'objectif est de créer un modèle de classification supervisée permettant de prédire est ce que le **revenu annuel** d'un adulte américain dépasse les 50 000$ ou non.

L'ensemble des données est divisé en deux parties : ensemble d'apprentissage (**train.csv**) et ensemble de test (**test.csv**).

Voici le **dictionnaire** de l'ensemble des données :

Variable	Définition	Valeurs
income	Revenu annuel (classe)	>50000$, <=50000$
age	Age	
workclass	Statut professionnel	
fnlwgt	Final weight : c'est un entier attribué par l'agence de recensement	
education	Le plus haut niveau d'éducation atteint par un individu.	
education-num	Le plus haut niveau d'éducation atteint par un individu sous forme numérique	
marital-status	Statut familial	
occupation	Profession	
relationship	Relation d'un individu avec un autre	
race	Race	
Sex	Sexe	
capital-gain	Les gains annuels	
capital-loss	Les pertes annuelles	
hours-per-week	Nombre d'heures de travail par semaine	
native-country	Pays d'origine	

4. Spécifications techniques

Le mini-projet doit être réalisé obligatoirement en respectant les spécifications techniques suivantes :

- Langage de programmation : **Python**
- Environnement de développement **: Jupyter Notebooks**
- Packages du calcul scientifique et manipulation des données **: numpy, scipy, pandas**
- Package de visualisation des données : **matplotlib et/ou seaborn**
- Package d'apprentissage automatique (machine learning) : **scikit-learn**

5. Spécifications fonctionnelles

Au cours de ce mini-projet vous allez aborder l'ensemble des étapes du processus d'extraction de connaissances à partir de données (ECD).

Alors, voici l'ensemble des exigences que vous devez satisfaire en réalisant ce mini-projet :

5.1. Data understanding :

- **Importation** des **Packages** du langage Python dédiés à la data science (voir spécifications techniques)
- **Chargement** des ensembles de données : l'ensemble d'apprentissage et l'ensemble de test
- **Affichage** des données : afficher un aperçu des 10 premières instances de chaque ensemble de données.
- **Description** et **analyse** des données de l'ensemble d'apprentissage : Afficher le volume (nombre total d'instances) et la dimension des données (nombre total des

attributs), les types et le codage des données et quelques statistiques descriptives (moyenne, écart-type, quartiles, valeur minimale, valeur maximale, etc.). Analyser les différentes valeurs.

- **Visualisation** des données : afin d'approfondir votre compréhension des données et chercher d'éventuelles **corrélations** entre la variable cible (la classe) les autres attributs de l'ensemble d'apprentissage, vous êtes demandés de réaliser plusieurs types de graphiques (histogrammes, nuages de points, boîtes à moustaches, etc.). La variable fondamentale dans tous les graphiques doit être l'attribut classe (la variable cible).

5.2. Nettoyage des données

- Détection et traitement des **valeurs manquantes** des deux ensembles de données (apprentissage et test) : afficher dans un tableau, le nombre de valeurs manquantes pour chaque attribut des deux ensembles de données. Sur la base de votre compréhension des données, proposer puis appliquer une technique de traitement des valeurs manquantes. Afficher un aperçu de chacun des deux ensembles de données pour mettre en évidence la disparition des valeurs manquantes.
- Détection et traitement des **valeurs aberrantes** des deux ensembles de données (apprentissage et test) : proposer puis appliquer une technique de détection des valeurs aberrantes. Afficher dans un tableau, le nombre de valeurs aberrantes pour chaque attribut des deux ensembles de données. Sur la base de votre compréhension des données, proposer puis appliquer une technique de traitement des valeurs aberrantes. Pour chaque attribut faisant l'objet d'une ou de plusieurs valeurs aberrantes, proposer puis afficher un graphique mettant en évidence la disparition de ces valeurs.

5.3. Transformation des données

- Sur la base de votre compréhension des données, vous pouvez proposer, avec justification, de supprimer un ou plusieurs **attributs** qui ne sont **pas discriminants** (pertinents) par rapport à la création des modèles de classification.
- Dans le but de faciliter l'application de certains algorithmes de machine learning, en l'occurrence l'algorithme des KNN, vous devez **transformer** toutes les données de type catégorielles en données numériques. Dans les deux ensembles de données (apprentissage et test), repérer les attributs catégoriels et transformer les en données qualitatives numériques.
- **Normaliser** si besoin (avec justification) les valeurs des attributs des deux ensembles de données. Afficher un aperçu des deux ensembles de données après l'application de la normalisation.

5.4. Création et optimisation des modèles

Après les étapes de compréhension et de préparation de données, il est temps de créer des modèles de classification supervisée.

L'objectif est de créer **trois modèles** de classification : les **KNN** (les plus proches voisins), les **Arbres de Décision** et un autre modèle de votre choix. Ensuite, nous devons comparer leurs performances et choisir bien évidemment celui qui donne les meilleurs résultats de classification. Dans ce mini-projet nous allons considérer l'**exactitude** (**Accuracy**) comme étant la mesure de **performance** des modèles.

Pour atteindre cet objectif, vous devez :

- Diviser **l'ensemble d'apprentissage** en deux sous-ensembles : **80%** pour l'apprentissage et **20%** pour l'**évaluation**.

- Utiliser la technique de **validation croisée** (en utilisant la classe **GridSearchCV** de scikit-learn) pour **ajuster** (optimiser) les **paramètres** de chaque modèle de classification. Ce traitement doit s'appliquer sur le sous-ensemble d'apprentissage (**80%**).
- Sur la base du sous-ensemble d'apprentissage (80%), créer les trois modèles de classification en utilisant les valeurs de paramétrage (les plus convenables) déduites de la phase précédente (validation croisée)
- Appliquer les trois modèles que vous avez créé, sur le sous-ensemble d'évaluation (20%) et calculer leur mesure de performance (accuracy).
- Comparer les performances des trois modèles et en choisir le meilleur.

5.5. Test du modèle

Selon les résultats obtenus dans la phase précédente, vous devez appliquer le modèle de classification qui a donné la meilleure valeur d'exactitude (accuracy) sur l'ensemble de test, afficher les résultats de classification ainsi que les mesures de performances suivantes :

- Exactitude (Accuracy)
- Matrice de confusion
- Précision
- Rappel
- F-score