TP 2 - Panorama des méthodes de fouille de données

Introduction

L'objectif de cette séance est de passer en revue les différents types de méthodes utilisées en fouille de données, que ce soit en analyse exploratoire ou en analyse prédictive.

1 Analyse exploratoire

1.1 Classification / Clustering

Dans cette première partie, vous avez à étudier les données « covid19 » établies à partir des fichiers téléchargées depuis le site Web du Centre européen de prévention et de contrôle des maladies [1], une agence de l'Union européenne. Ces données fournissent des indicateurs de santé liées à la pandémie de COVID-19 sur 30 pays de l'espace économique européen. Vous trouverez un descriptif des variables étudiées dans le fichier `covid19.v2.txt`.

1. Ouvrez le fichier `covid19.v2.csv` à l'aide de la bibliothèque pandas. Combien y a-t-il d'attributs ? De quels types sont-ils ?
2. Y a-t-il des valeurs manquantes dans les données ? Comment ces valeurs sont-elles représentées dans le ficher ?

   Les nombres de doses injectées dans un pays sont liés à sa population. De façon à avoir des variables qui donnent une idée plus précise de la couverture vaccinale du pays, calculez plutôt le nombre de doses pour 1 000 habitants. Une fois que vous aurez réalisé ce calcul, supprimez la variable correspondant au nombre d'habitants.
3. Il est souvent utile d'appliquer un filtre de normalisation sur tous les attributs avant d'utiliser des méthodes de *clustering*. Quel est l'effet du filtre StandardScaler sur les données ? A-t-on besoin ici d'employer ce filtre sur les données étudiées ?
4. Effectuez une analyse en composantes principales (ACP) sur les données à l'aide de la bibliothèque scikit-learn.

   Joignez au rapport l'affichage des instances étiquetées par le code du pays suivant les facteurs 1 et 2, puis suivant les facteurs 1 et 3 de l'ACP.
5. Que représentent les 3 premiers facteurs de l'ACP ? Vous pourrez répondre à cette question en affichant la variance expliquée en fonction du rang du facteur et en traçant le cercle de corrélation entre les facteurs et les variables originales.

   Commentez les deux graphiques obtenus à la question précédente (affichage dans les deux premiers plans de l'ACP) en discutant la proximité entre les pays suivant les 3 facteurs.
6. Reprenez les données telles qu'elles étaient avant leur transformation par l'ACP. En utilisant KMeans de la bibliothèque scikit-learn, réalisez un *clustering* avec la méthode des k-moyennes, où k représente le nombre souhaité de *clusters*. Tracez la variation du R^2 pour k variant de 2 à 20. D'après cette courbe, quelle valeur de k retiendriez-vous ?

1. `https://www.ecdc.europa.eu/en/covid-19/data`

7. En fixant k à 8, commentez les profils des groupes contenant la France, le Danemark et la Bulgarie.

1.2 Recherche de règles d'association

Cette partie aborde la recherche de règles d'association, en illustrant son principe par l'algorithme *Apriori*, disponible dans la bibliothèque mlxtend.

Apriori opère sur des données dont tous les attributs sont nominaux, les attributs numériques devant être discrétisés auparavant. L'algorithme génère des règles, sous la forme *prémisses* (appelées *antecedents* dans la bibliothèque utilisée) ⇒ *conclusion* (appelée *consequents*). Ces règles sont ordonnées par défaut selon leur support, c'est-à-dire le nombre d'instances qui satisfont la règle.

La découverte de règles d'association est susceptible de produire un nombre très important de règles, dont la plupart sont peu pertinentes pour le problème. Afin de limiter le nombre de résultats obtenus, l'algorithme est basé sur un niveau de couverture minimal, c.-à-d. la proportion des instances couvertes par chaque règle. Pour des raisons d'efficacité algorithmique, la méthode *Apriori* commence par chercher des règles à un item (restreint à la classe à prédire), puis étend celles qui ont une couverture suffisante avec une valeur d'attribut pour obtenir des règles à deux items. Elle effectue ensuite de nouvelles itérations en ajoutant à chaque fois un nouvel item.

Dans cet exercice, il est demandé d'étudier le fichier `supermarket.csv`, qui fournit des informations sur les achats dans un supermarché de Nouvelle-Zélande. L'objectif est ici de découvrir des règles s'appliquant sur le panier de consommation.

1. Ouvrez les données à l'aide de la bibliothèque pandas. Quels sont les types des attributs ? Que représentent les valeurs associées aux ' ?' ? Comment agit les commandes du fichier squelette, permettant la discrétisation ?
2. Le niveau de couverture est très dépendant des données. Établissez l'ensemble des itemsets fréquents à l'aide de la fonction apriori() de mlxtend en fixant un support minimal à 0,1. Expliquez ce que signifie cette valeur de 0,1. Combien d'itemsets obtenez-vous ? Donnez leur fréquence en fonction du nombre d'items qu'ils contiennent.
3. Construisez les règles à partir de l'ensemble des itemsets précédemment obtenus en recourant à la fonction association_rules() de mlxtend. Pour ne pas avoir un nombre trop important de règles générées, fixez un seuil minimal de 0,7 pour la valeur de confiance.

   Combien de règles obtenez-vous ? Commentez la première règle obtenue.
4. *Apriori* possède plusieurs critères pour évaluer la pertinence d'une règle, parmi lesquelles *confidence* et *lift*[2]. Quelle est la meilleure règle pour l'ensemble « supermarket » pour chacune des métriques *confidence* et *lift* ? Commentez les scores et règles obtenus.
5. Un dirigeant du supermarché souhaite améliorer ses ventes en réorganisant le positionnement des produits dans ses rayons. D'après votre analyse du panier de consommation fourni, quels produits sont souvent achetés conjointement avec des biscuits et des aliments pour animaux domestiques ?[3]

2. Vous pouvez vous référer au cours de l'UCE ou bien aux informations disponibles sur la page `https://rasbt.github.io/mlxtend/user_guide/frequent_patterns/association_rules/` .

3. Vous aurez peut-être besoin de revoir la valeur auparavant fixée de support minimal.

2 Analyse prédictive

2.1 Classement

Dans cette partie du TP, vous allez étudier et comparer le comportement de différents classifieurs sur les données « Adult earnings ». Ces données collectées aux États-Unis en 1994 à partir d'une base de donnée du bureau de recensement concernent les informations personnelles et financières de divers habitants de ce pays. Les données sont disponibles dans l'archive `adult.zip`. Pour obtenir la signification des valeurs des attributs, vous pouvez vous référer au fichier `.info` de l'archive.

Les méthodes de classement que nous testerons sur ces données seront :

— le classifieur élémentaire associant chaque instance à la classe majoritaire (DummyClassifier avec la stratégie most_frequent),
— la méthode bayésienne naïve (GaussianNB),
— l'arbre de décision (DecisionTreeClassifier),
— la forêt d'arbres décisionnels (RandomForestClassifier)
— la méthode de régression logistique (LogisticRegression),
— le modèle SVM (SVC).

1. Ouvrez les données d'apprentissage et de test à l'aide de la bibliothèque pandas. Combien y a-t-il d'attributs dans ces fichers ? Quels sont leurs types ? Quelle est la variable à prédire ? Les classes sont-elles équilibrées dans les fichiers ? Y a-t-il beaucoup de valeurs manquantes dans les données ? Comment ces valeurs sont-elles représentées dans le ficher ?
2. Expliquez en une phrase en quoi consiste la méthode élémentaire de classement DummyClassifier utilisant la stratégie most_frequent. Dans quel cas ce classifieur pourrait donner les meilleurs résultats parmi ceux testés ?
3. Dans un premier temps, seules les données numériques sont considérées pour prédire la classe. Remplacez chaque valeur manquante par la moyenne obtenue par chaque attribut en utilisant SimpleImputer. Normalisez ensuite les valeurs numériques à l'aide de StandardScaler.

   Calculez les performances des cinq méthodes de classement étudiées sur les données « adult earnings » en réalisant une validation croisée en 5 blocs sur les données d'apprentissage. Parmi les méthodes testées, quelle est la plus performante par rapport au taux de classification ?
4. Dorénavant, on se propose de n'employer que les attributs catégoriels. Remplacez chaque valeur manquante par le mode (valeur la plus fréquente) de chaque attribut en recourant à SimpleImputer. Réalisez une disjonction des variables catégorielles à l'aide de la classe OneHotEncoder de la bibliothèque scikit-learn.

   Calculez les performances de classement des cinq méthodes étudiants, en réalisant à nouveau une validation croisée en 5 bocs. Observez-vous des performances similaires à ce que vous aviez à la question précédente ?
5. Utilisez enfin l'ensemble des colonnes, c-à-d en prenant en compte les attributs numériques et catégoriels. Calculez à nouveau les performances de classement en validation croisée à 5 blocs et commentez les résultats obtenus.
6. Sélectionnez le meilleur modèle testé lors des configurations précédentes et apprenez-le sur l'ensemble des données d'apprentissage. Quelle est la performance mesurée avec ce modèle sur le corpus de test ? Quelle est la classe pour laquelle la prédiction fait le plus d'erreurs ?