Analyse de données pour guider les choix d'affectation de l'aide humanitaire internationale

Projet - UV ODATA

2 novembre 2020

1 Objectifs du projet et livrables attendus

L'objectif du projet est d'effectuer une classification non supervisée des principaux pays du monde en matière de développement à partir de données socio-économiques et de santé sur chaque pays. Cette classification permettra de guider les choix d'affectation de l'aide humanitaire internationale en identifiant les pays dans lesquels il est nécessaire d'intervenir en priorité et éventuellement de préciser les domaines d'action.

Les méthodes de clustering à considérer / tester sont les suivantes :
- K-means
- Classification hiérarchique ascendante
- Modèle de mélange de Gaussiennes
- DBSCAN

D'abord, une étude préalable vous permettra de comparer ces 4 méthodes sur des données simulées. Vous étudierez le principe de la méthode DBSCAN par vous-même, vous analyserez les performances des différentes techniques sur plusieurs jeux de données et vous préciserez quelles sont les techniques les mieux adaptées à chaque type de données.

Ensuite, vous vous intéresserez au problème de clustering des principaux pays du monde en matière de développement à partir des données socio-économiques et de santé disponibles sur chacun de ces pays.

Vous analyserez les partitions obtenues par les différentes méthodes à l'aide des métriques de votre choix et vous justifierez les résultats obtenus. Enfin vous proposerez une liste des pays qui ont le plus besoin de l'aide humanitaire internationale (10 maximum). Si c'est possible, vous préciserez également les domaines d'action.

A la fin du projet, vous devrez fournir :

1. Un rapport (format pdf) comprenant les éléments suivants :
   — un rappel des objectifs du projet,
   — les réponses aux différentes questions posées,
   — une description du protocole expérimental mis en place : objectifs, métriques utilisées, ... Pour chaque méthode vous préciserez les paramètres choisis (initialisation, distance, ...) et vous justifierez vos choix,
   — une analyse et une interprétation des résultats obtenus.
2. Le code, clair et commenté.

2 Etude préalable : Comparaison des méthodes de clustering sur des données simulées

2.1 Etude de la méthode DBSCAN

Parmi les méthodes considérées, seule la méthode DBSCAN n'a pas été étudiée en cours, il faut donc faire une recherche sur cette méthode et comprendre son principe.

Expliquez le principe de la méthode et indiquez ses avantages.

2.2 Etude des classes et modules relatifs aux différentes méthodes

Dans le projet, vous allez considérer les classes ou modules suivants des librairies `scikit-learn` et `scipy` :

— Méthode K-means : `sklearn.cluster.KMeans`
— Classification hiérarchique ascendante : `scipy.cluster.hierarchy`
— Modèle de mélange de Gaussiennes : `sklearn.mixture.GaussianMixture`
— DBSCAN : `sklearn.cluster.DBSCAN`

Etudiez ces différentes classes et modules : paramètres d'appel, attributs, méthodes, fonctions.

2.3 Expérimentations

Il s'agit d'étudier les performances des 4 méthodes sur des données simulées correspondant à des clusters de formes différentes.

Importez les données contenues dans les fichiers suivants : `g2-2-20.txt`, `g2-2-100.txt`, `jain.txt`, `Aggregation.txt` et `pathbased.txt` à l'aide de la

fonction `read_csv()` de la librairie `pandas`. Visualisez les nuages de points correspondants.

Pour chaque méthode :
— Précisez les valeurs choisies pour les paramètres, par exemple pour l'initialisation, le type de distance, la méthode de linkage en classification hiérarchique....
— Proposez un partitionnement de chaque jeu de données. Pour le nombre de clusters K, vous choisirez le nombre réel de clusters.

Evaluez les performances des 4 méthodes :
— de façon qualitative / visuelle : représentez les clusters obtenus par des couleurs différentes. Comparez avec les "vrais" clusters quand l'information est disponible.
— de façon quantitative en comparant le partitionnement obtenu avec le vrai partitionnement (quand il est connu). Calculez l'indice de Rand ajusté (ARI) donné par la fonction `adjusted_rand_score()` du module `sklearn.metrics`.

Comparez les performances obtenues avec les 4 méthodes. Précisez quelles sont les méthodes les mieux adaptées aux différentes formes de clusters.

3 Classement des principaux pays du monde en fonction de leur développement

L'objectif est maintenant d'utiliser ces 4 méthodes pour classer automatiquement les principaux pays du monde en fonction de leur développement à partir de données socio-économiques et de santé sur chaque pays.

L'intérêt de cette classification est de guider les choix d'affectation de l'aide humanitaire internationale en identifiant les pays dans lesquels il est nécessaire d'intervenir en priorité et éventuellement de préciser les domaines d'action.

On dispose d'un fichier `data.csv`, disponible sur MLS, qui stocke pour 167 pays les données suivantes :
— Taux de mortalité infantile (nombre de décès d'enfants de moins de 5 ans pour mille naissances)
— Exportations de biens et de services par habitant (exprimé en pourcentage du PIB par habitant)
— Dépenses totales de santé par habitant (exprimé en pourcentage du PIB par habitant)
— Importations de biens et de services par habitant (exprimé en pourcentage du PIB par habitant)
— Revenu net par personne

— Inflation (taux de croissance annuel du PIB)
— Espérance de vie
— Taux de fécondité (nombre moyen d'enfants par femme)
— PIB (produit intérieur brut) par habitant

Visualisez le tableau de données, puis importez les données du fichier.

3.1 Examen des données

Après l'importation, il est important d'examiner plus en détail ces données, en particulier :
— la taille du jeu de données,
— le type des données (numérique : int, float ou qualitatif/catégoriel : object),
— la qualité des données (est-ce qu'il y a des données manquantes ou nulles ?),
— la distribution des données.

En utilisant les méthodes de la classe `DataFrame`, procédez à l'examen des données et notez les informations qui vous paraissent pertinentes.

En particulier, il est important d'identifier les données nulles et les données manquantes (représentées par le symbole 'NA' : Not Available) qui devront être pré-traitées avant le clustering (voir section suivante). En utilisant les méthodes `isnull()` et `isna()` de la classe `DataFrame`, vous pouvez obtenir respectivement le nombre de valeurs nulles et le nombre de valeurs manquantes pour chacune des variables.

Il est aussi intéressant de connaître les statistiques des données à traiter. Pour cela, vous pouvez utiliser la méthode `describe()` de la classe `DataFrame` et construire une visualisation de type histogramme pour chaque variable numérique avec la méthode `hist()` de la classe `DataFrame`.

Sélectionnez les variables qui seront utilisées pour le clustering.

3.2 Préparation des données

Pour faire fonctionner correctement les algorithmes de clustering, il est nécessaire d'avoir des données numériques de qualité (sans valeurs manquantes ou nulles).

Pour résoudre le problème des valeurs manquantes et des valeurs nulles, plusieurs solutions sont possibles :
— rechercher la vraie valeur manquante via d'autres sources,
— attribuer une valeur conforme à la distribution de la variable (moyenne, médiane, valeur la plus probable...) en utilisant la méthode `fillna()` de la classe `DataFrame`,

— supprimer la variable correspondante, si le nombre de valeurs manquantes est très important (plus d'un tiers des données environ).

Il est aussi nécessaire de recalibrer les données si les plages de valeurs sont différentes. Si nécessaire, centrez et réduisez les données en utilisant la classe `StandardScaler` de `scikit-learn`.

Après avoir réalisé toutes ces transformations, il est intéressant d'examiner à nouveau toutes les variables qui seront utilisées pour le clustering.

3.3 Recherche de corrélations

Pour aller plus loin dans l'analyse des données, il faut s'intéresser aux relations qui existent entre les variables. Pour cela, on peut calculer le coefficient de corrélation entre chaque couple de variables numériques en utilisant la méthode `corr()` de la classe `DataFrame` et la fonction `scatter_matrix()` de la librairie `pandas`.

Commentez les résultats obtenus.

3.4 Clustering des données

Effectuez le clustering proprement dit sur les données des différents pays avec les 4 méthodes proposées.

Pour chaque méthode :
— Précisez les valeurs choisies pour les paramètres.
— Evaluez la qualité des partitions obtenues pour différentes valeurs de K, le nombre de clusters, en utilisant les métriques de votre choix disponibles dans le module `sklearn.metrics`.
— Proposez une valeur de K, justifiez votre choix,
— Pour la valeur de K choisie, analysez les clusters obtenus au regard des données du fichier.
— Visualisez graphiquement les clusters en sélectionnant un (ou plusieurs) couple(s) de variables qui vous paraissent pertinents.

Comparez les résultats obtenus avec les 4 méthodes, interprétez et commentez les résultats.

Etudiez plus précisément le cluster correspondant aux pays les moins avancés en terme de développement. Proposez une liste de 10 pays maximum qui ont le plus besoin de l'aide humanitaire internationale. Si c'est possible, précisez également les domaines d'action.

3.5 Clustering des données après réduction de dimension

Avant d'appliquer les méthodes de clustering, effectuez d'abord une ACP pour réduire la dimension des données. Combien d'axes proposez-vous de conserver ? Donnez une interprétation de ces axes.

Appliquez les méthodes de clustering sur les nouvelles données obtenues après l'ACP et reprenez l'étude comme indiqué dans la section précédente.

Cette fois, représentez graphiquement les clusters obtenus sur le premier plan principal défini par les 2 premiers axes.

Comparez avec les partitions obtenues en appliquant les méthodes sur le jeu de données initial.

Donnez vos préconisations pour l'affectation de l'aide humanitaire internationale en identifiant les pays (10 maximum) dans lesquels il est nécessaire d'intervenir en priorité.