Fouille de données
Travail Pratique N°2

UE Data analytics big data

- Enseignante : M. Salima Mdhaffar
- Etudiante : Bensafi Sarra
- Année : 2022-2023

1.1 Classification / Clustering

CODE: https://colab.research.google.com/drive/1TbnajnnqaWpTNUAS4r7i7KZQ71p07_2y?usp=sharing

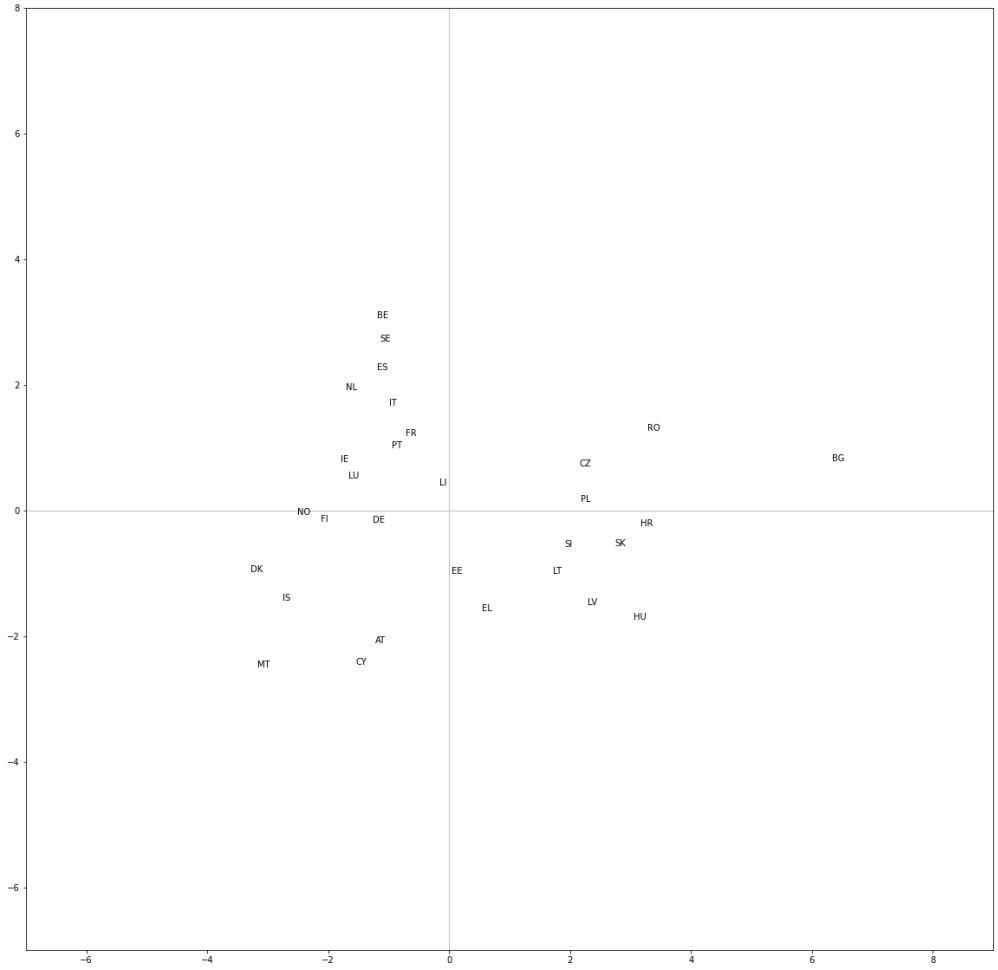
1. Lecture du fichier *covid19.v2.csv*
   Le nombre d'attributs est 18 et leur types sont

- **country** object
- **country_code** object
- **population** int64
- **vaccine_2021-winter** int64
- **vaccine_2021-spring** int64
- **vaccine_2021-summer** int64
- **vaccine_2021-fall** int64
- **vaccine_2022-winter** int64
- **testing_rate_2020** float64
- **testing_rate_2021** float64
- **deaths_rate-2020-spring** float64
- **deaths_rate-2020-summer** float64
- **deaths_rate-2020-fall** float64
- **deaths_rate-2021-winter** float64
- **deaths_rate-2021-spring** float64
- **deaths_rate-2021-summer** float64
- **deaths_rate-2021-fall** float64
- **deaths_rate-2022-winter** float644

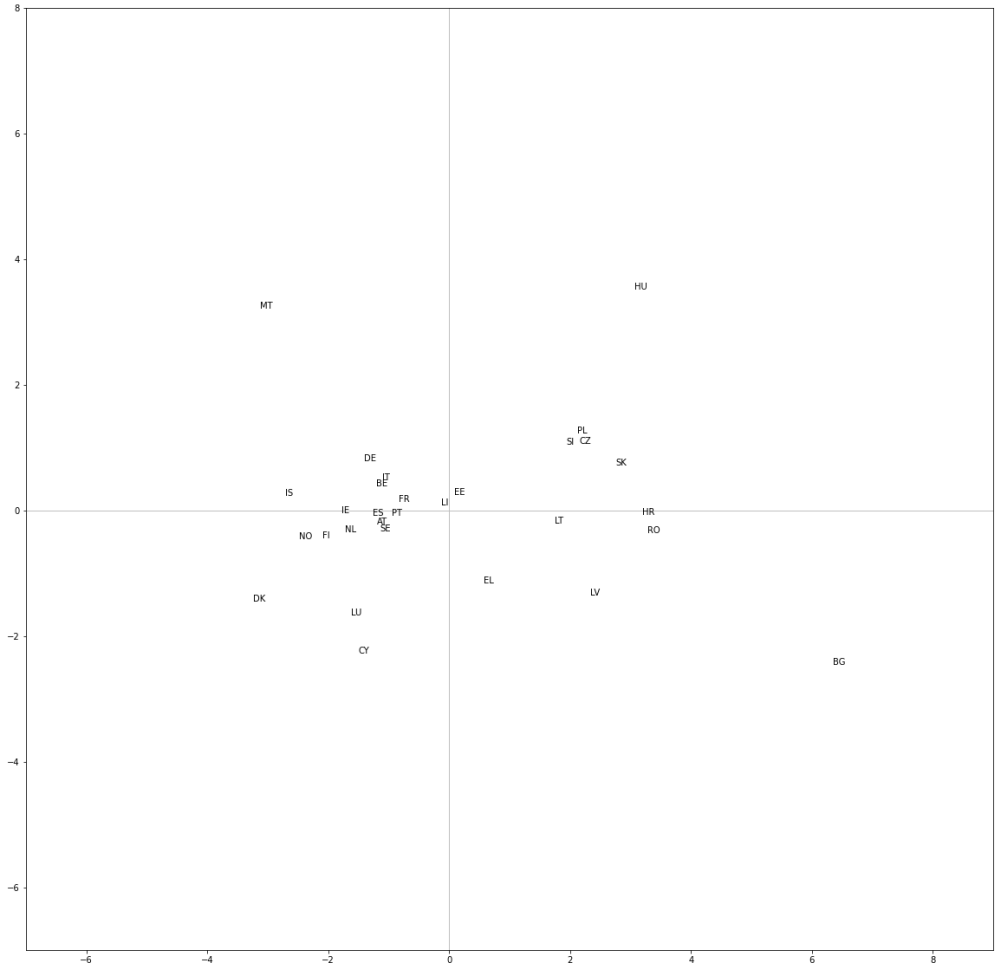
2. Il n'a pas de valeurs manquantes dans le fichier, dans le cas où elle existe ces valeurs sont parfois représentées par NA/None/?

3. On a besoin d'appliquer le filtre **StandardScaler** en général pour que nos valeurs soient sur la même échelle, comme par la suite ces données vont être l'entrée de nos modèles mathématiques, ces derniers n'apprécie pas les changements d'échelle, dans notre cas en particulier, nous avons certaines colonnes où il y a des outliers (valeurs aberrantes) alors, on a besoin de remettre les données sur la même échelle.

4. 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 :



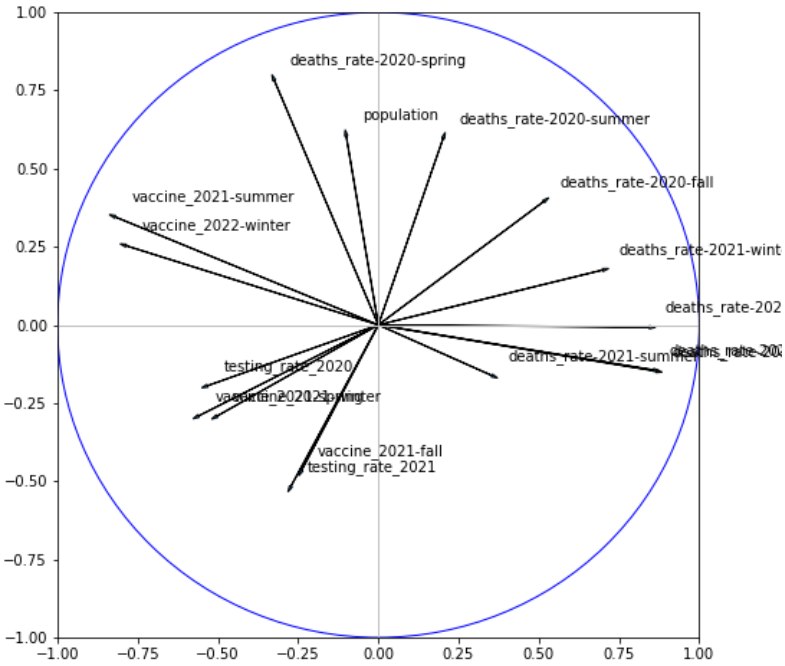
Acp instances axes 0 et 1



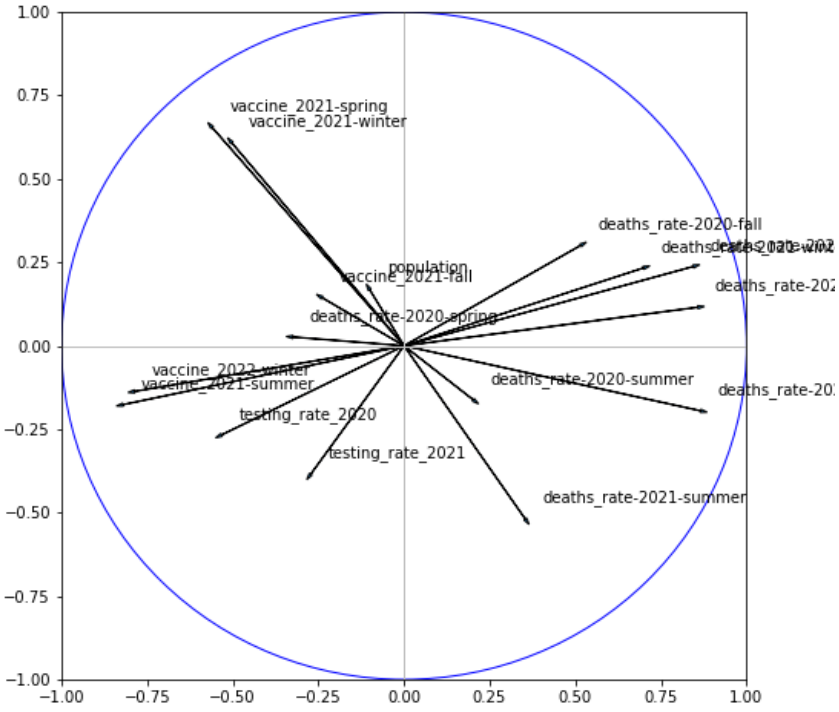
Acp instances axes 0 et 2

5. Que représentent les 3 premiers facteurs de l'ACP ?
    a. *la variance expliquée de chaque axe est expliqué par un ensemble de dimension*
    b. valeurs propres des 3 premières composantes
    0.35431 + 0.14367662 + 0.1061538 = **0.60414042**

    Plus de la moitié de l'information est contenu dans les 3 premiers axes.
    On peut voir que si on cumule l'ensemble de nos variances, on obtient 1
    c. les cercles de correlations



Cercle de corrélation entre l'axe 0 et l'axe 1



Cercle de corrélation entre l'axe 0 et l'axe 2

Analyse :

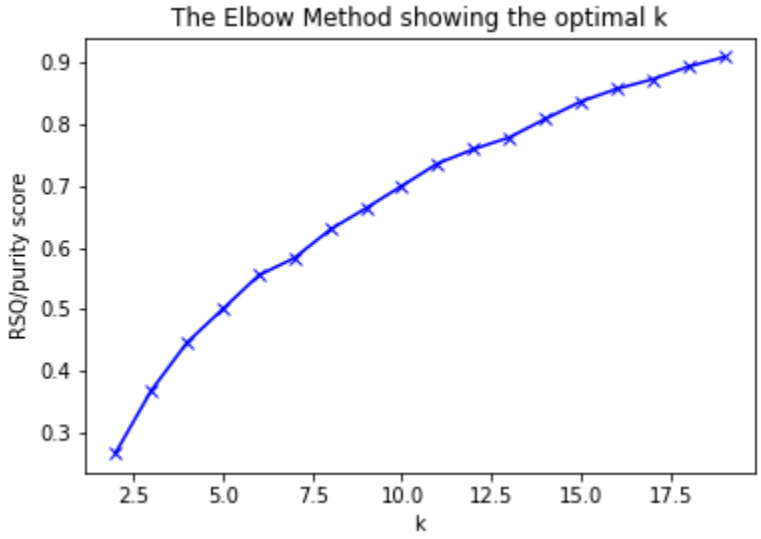
Exemple 1 : Si on regarde le cercle de corrélation n°1 on peut voir que le l'axe 1 est fortement corrélé avec les dimensions deaths_rate-2021-winter, deaths_rate-2021-spring et deaths_rate-2021-fall. Alors si on prend par exemple la Bulgarie (BG) qui est très proche de l'axe 1 dans le plot n°1 on voir dans les données que le nombre de death en Hiver 2021 , automne 2021 et été 2021 sont très élevé.

Exemple 2 :Si on prend l'exemple de la France FR on peut voir qu' elle est corrélé avec l'axe 2 qui renferme beaucoup d'information sur la dimension vaccine_2021-summer c'est à dire que beaucoup de gens ont été vaccinés en France dans l'été 2021 ce qui est exactement le cas.

Exemple 3 :Si on regarde le cercle de corrélation n°2 on peut voir que le l'axe 3 est fortement corrélé (positivement) avec les dimensions vaccine_2021-winter et vaccine_2021-spring. On déduit alors que Malte et Hongrie sont les pays qui ont le plus vaccinées en 2021.

6. Réalisez un clustering avec la méthode des k-moyennes
    a. La pureté n'est pas calculable dans notre cas car on a pas de cluster prédéfinis sur tel pays doit appartenir à telle classe car ce n'est pas des données labellisées déjà.

b. R carré
    i. Pour déterminer le nombre optimal de clusters, nous devons sélectionner la valeur de k au « coude », c'est-à-dire le point après lequel l'inertie commence à diminuer de manière linéaire. Ici j'ai choisi le K=9.



Courbe d'évolution d'homogénéité dans les clusters

7. En fixant k à 8, nous pouvons remarquer que
    a. Le cluster de
        i. France est composé de : Belgium, France, Germany, Italy, Spain, Sweden
        ii. Denmark est composé de : Denmark, Estonia, Finland, Iceland, Ireland, Netherlands, Norway
        iii. Bulgaria est composé de : Bulgaria
    b. on peut remarquer que les 3 clusters sont composé de pays proche géographiquement
    Le cluster France : la gestion du covid pour des pays comme Belgium, France, Germany, Italy, Portugal, Spain a été presque la même en terme de mort et de vaccinées, la France est l'un des trois pays d'Europe les plus touchés par la pandémie avec l'Italie et l'Espagne ce qui explique aussi cette proximité dans les cluster.
    Le cluster Denmark : représente cette approche scandinave sur la gestion du Covid-19 qui a laissé percevoir le choix de prendre des mesures très tôt dans l'apparition de la crise, afin de prévenir une brusque montée de l'épidémie sur son territoire, s'opposant à celle de la Suède, ayant une approche beaucoup moins restrictive

Bonus : par exemple le pays comme la Suède qui seul dans son cluster un des pays qui a eu une gestion très singulière du Covid-19 car le gouvernement n'a jamais imposé de confinement par exemple est cela ressort lors de l'analyse des données.
Le cluster Bulgaria : ce pays des Balkans a eu le taux de vaccination les plus bas de l'Union européenne et la situation épidémiologique était souvent hors de contrôle avec un nombre de morts très élevé.

Recherche de règles d'association

CODE **:** https://colab.research.google.com/drive/1LjpUYxe5oBJ1kIPNl3B63abG-pDfWXCu?usp=sharing

1. Je remarque que chaque ligne est une transaction effectuée par un client dans le supermarché et chaque colonne est un rayon(type de produit), alors on déduit que si il y'a **t** cela signifie que le client a acheter dans ce rayon et si **?** alors il n'a pas acheté. Les types sont alors catégorielles.
   La discrétisation a pour objectif de transformer le qualitatif **? t** vers binaire.
2. En fixant le support à **0.1** j'ai **10281** règles, le support permet de savoir si un itemset est souvent présent ou pas ici on veut que l'item soit présent dans au moins **10%** des transaction et veut dire qu'un élément ou tout ensemble doit se reproduire un certain nombre de fois pour être considéré comme suffisamment significatif , On peut par exemple décider qu'un item ou itemset est significatif lorsqu'il est présent à hauteur d'au moins **70%** dans les données ici on l'a fixé à **10%**.

fréquence des items Counter({4: 3950, 3: 2598, 5: 2470, 2: 634, 6: 558, 1: 52, 7: 20})

3. Nous avons 24570 règles
   Regle 1 :
   En achetant l'item su rayon **(department 1)** il est plus probable d'acheter l'item du rayon **(bread and cake)**
   nous avons 24570 règles
   Regle 4 :
   En achetant l'item **(tea)** il est plus probable d'acheter l'item **(bread and cake)**
   nous avons 24570 règles.
4. Le niveau de confiance va nous permettre de savoir combien de fois la règle est vraie. Si nous reprenons par exemple la règle {tea} => {cake} , l'objectif est de savoir dans combien de cas l'achat du tea conduit effectivement à acheter du cake.
   La règle avec la meilleure confiance de 0.937 est :

La finalité du lift est de mesurer le niveau d'interdépendance entre les éléments. Une valeur de lift de 1 indique qu'il n'y a pas de dépendance entre les éléments. Une valeur de lift supérieure à 1 indique au contraire qu'il existe une interdépendance importante entre les variables.
Le magasin pourra ainsi proposer des réductions sur les produits ** pour que les gens achètent des *** .
La règle avec le meilleur lift qui est de 2.346 est :

5. Les produits qui sont souvent achetés conjointement avec des biscuits et des aliments pour animaux domestiques sont :
    a. Avec seuil de confiance = 0.7 , nombre de produit sont 12
        i. {'vegetables', 'margarine', 'party snack foods', 'fruit', 'milk-cream', 'bread and cake', 'total_high', 'baking needs', 'sauces-gravy-pkle', 'tissues-paper prd', 'juice-sat-cord-ms', 'frozen foods'}
    b. Avec seuil de confiance = 0.5 , nombre de produit sont 19
        i. {'baking needs', 'beef', 'bread and cake', 'breakfast food', 'canned vegetables', 'cheese', 'dairy foods', 'department137', 'frozen foods', 'fruit', 'juice-sat-cord-ms', 'margarine', 'milk-cream', 'party snack foods', 'sauces-gravy-pkle', 'soft drinks', 'tissues-paper prd', 'total_high', 'vegetables'}
    c. Avec seuil de confiance = 0.1 , nombre de produit sont 24 :
        i. {'baking needs', 'beef', 'bread and cake', 'breakfast food', 'canned fruit', 'canned vegetables', 'cheese', 'confectionary', 'dairy foods', 'department137', 'frozen foods', 'fruit', 'juice-sat-cord-ms', 'laundry needs', 'margarine', 'milk-cream', 'party snack foods', 'prepared meals', 'sauces-gravy-pkle', 'soft drinks', 'tissues-paper prd', 'total_high', 'vegetables', 'wrapping'}

Ici nous pouvons observer que plus on augmente la confiance, plus le produit qu'on pourrait acheter diminue. Puisque le nombre de règles diminue en fonction de ce dernier.
Le principe des règles d'association est donc basé sur l'idée que la récurrence d'une séquence dans un ensemble de données peut nous amener à en faire une règle, Si nous nous plaçons dans le contexte du commerce la découverte de tels éléments nous permettra d'améliorer l'agencement du magasin, de mieux organiser les promotions et de parfaire le marketing des magazin.

Classement

Code : https://colab.research.google.com/drive/1vwG6fD3FpQg-F86MVhjag5M2pURsQw1x?usp=sharing

1. Lecture du fichier et compréhension des données
    a. Les attribut dans les fichier d'AdultEarnings sont 15

```
age            =>     int64
workclass      =>     object
fnlwgt         =>     int64
education      =>     object
education-num  =>  int64
marital-status =>   object
occupation     =>    object
relationship   =>    object
race           =>      object
```

sex => object
capital-gain => int64
capital-loss => int64
hours-per-week => int64
native-country => object
class => object

b. **La class a prédir est 'class'** qui représente le revenu d'un citoyens américains dans les années 1994
c. Il y a des valeurs manquantes dans la data car elles sont exprimées en ? , il y'a **4262** dans le train et **2203** dans le test
d. Train : Les classes ne sont pas équilibrés
<=50K 12435 et >50K 7841
Proportion
All = 32561
<=50K 24720/All = 0.75 => 75% sont inférieure a 50k
>50K 32561/All = 0.24 => 24% sont supérieure a 50k

2. **Dummy Classifier** : Il s'agit d'un modèle de classification qui fait des prédictions sans essayer de trouver des modèles pattern dans les données. Le modèle par défaut examine essentiellement l'étiquette la plus fréquente "most frequent" dans l'ensemble de données de formation et effectue des prédictions en fonction de cette étiquette tout simplement.
Généralement ce dernier est utilisé comme **modèle de références**, pour comparer les autres modèles. Le dummy certe peut donner dans notre cas une très bonne accuracy car nous avons des données déséquilibrés, mais une très bonne accuracy ne signifie pas que le model fait un bon travail.

3. On peut voir que les meilleurs classifieurs en considérant que les attributs numériques sont : **`Logistic regression et SVM`**
    a. Les taux de classifications sont :

- Pour Dummy classifier on cross-validation: 0.76
- Pour la méthode Naïve Bayes Classifier : 0.80
- Pour la méthode Decision tree : 0.77
- Pour la méthode Random Forest Classifier : 0.81
- Pour la méthode Logistic Regression Classifier : 0.82
- Pour la méthode SVM classifier : 0.82

    b. Matrice de confusion pour le dummy :

[[24720 0]
[7841 0]]

On peut remarquer qu'il a toujours classé toutes les instances dans la classe 0 c'est a dire income<=50k , on voit bien que 24720 classé correctement or 7841 qui était de la class 1 >50 on était classé comme class 0.

Les résultats **en considérant que les attributs numériques :**

```
CROSS VALIDATION With numerical attributes
Accuracy of Dummy classifier on cross-validation: 0.76 (+/- 0.00)
Accuracy of Naive Bayes classifier on cross-validation: 0.80 (+/- 0.01)
Accuracy of Decision tree classifier on cross-validation: 0.77 (+/- 0.01)
Accuracy of Random Forest classifier on cross-validation: 0.81 (+/- 0.01)
Accuracy of Logistic regression classifier on cross-validation: 0.82 (+/- 0.01)
Accuracy of SVM classifier on cross-validation: 0.82 (+/- 0.01)
Accuracy of Dummy classifier on cross-validation: 0.76
[[24720     0]
 [ 7841     0]]
Accuracy of Naive Bayes classifier on cross-validation: 0.80
[[23493  1227]
 [ 5411  2430]]
Accuracy of Decision tree classifier on cross-validation: 0.77
[[20839  3881]
 [ 3636  4205]]
Accuracy of Random Forest classifier on cross-validation: 0.81
[[22269  2451]
 [ 3847  3994]]
Accuracy of Logistic regression classifier on cross-validation: 0.82
[[23452  1268]
 [ 4750  3091]]
Accuracy of SVM classifier on cross-validation: 0.82
[[23895   825]
 [ 4912  2929]]
```

4. On peut voir que les meilleurs classifieurs en considérant que les attributs Catégorielles sont : **`Logistic regression, Random Forest classifier et SVM`**

    a. Les taux de classifications sont :

- Pour Dummy classifier on cross-validation: 0.76
- Pour la méthode Naïve Bayes Classifier : 0.51
- Pour la méthode Decision tree : 0.82
- Pour la méthode Random Forest Classifier : 0.83
- Pour la méthode Logistic Regression Classifier : 0.83
- Pour la méthode SVM classifier : 0.83

Les résultats en considérant que les attributs catégorielles:

```
CROSS VALIDATION With Categorical attributes
Accuracy of Dummy classifier on cross-validation: 0.76 (+/- 0.00)
Accuracy of Naive Bayes classifier on cross-validation: 0.51 (+/- 0.05)
Accuracy of Decision tree classifier on cross-validation: 0.82 (+/- 0.00)
Accuracy of Random Forest classifier on cross-validation: 0.83 (+/- 0.01)
Accuracy of Logistic regression classifier on cross-validation: 0.83 (+/- 0.01)
Accuracy of SVM classifier on cross-validation: 0.83 (+/- 0.01)
Accuracy of Dummy classifier on cross-validation: 0.76
[[24720     0]
 [ 7841     0]]
Accuracy of Naive Bayes classifier on cross-validation: 0.51
[[ 9033 15687]
 [  397  7444]]
Accuracy of Decision tree classifier on cross-validation: 0.82
[[22373  2347]
 [ 3586  4255]]
Accuracy of Random Forest classifier on cross-validation: 0.82
[[22524  2196]
 [ 3512  4329]]
Accuracy of Logistic regression classifier on cross-validation: 0.83
[[22908  1812]
 [ 3725  4116]]
Accuracy of SVM classifier on cross-validation: 0.83
[[22909  1811]
 [ 3729  4112]]
```

Ici le naive bayes donne des résultats pas très satisfaisant pour les attributs catégoriel : La principale différence est que naive bayes suppose que les attributs sont indépendants les uns des autres et qu'il n'y a pas de corrélation eux, ici ce n'est pas le cas par exemple : education et education-num représente la même information et peut être une redondance dans les informations.

5. On peut voir que le meilleur classifieur en considérant les attributs numériques et catégoriels est : SVM
    a. Les taux de classifications sont :

• Pour Dummy classifier on cross-validation: 0.76
• Pour la méthode Naïve Bayes Classifier : 0.51
• Pour la méthode Decision tree : 0.82
• Pour la méthode Random Forest Classifier : 0.83
• Pour la méthode Logistic Regression Classifier : 0.83
• Pour la méthode SVM classifier : 0.83

Les résultats en considérant que les attributs catégorielles et numériques:

```
Accuracy of Dummy classifier on cross-validation: 0.76 (+/- 0.00)
Accuracy of Naive Bayes classifier on cross-validation: 0.54 (+/- 0.03)
Accuracy of Decision tree classifier on cross-validation: 0.81 (+/- 0.01)
Accuracy of Random Forest classifier on cross-validation: 0.85 (+/- 0.01)
Accuracy of Logistic regression classifier on cross-validation: 0.85 (+/- 0.01)
Accuracy of SVM classifier on cross-validation: 0.86 (+/- 0.01)
Accuracy of Dummy classifier on cross-validation: 0.76
[[24720     0]
 [ 7841     0]]
Accuracy of Naive Bayes classifier on cross-validation: 0.54
[[10145 14575]
 [  388  7453]]
Accuracy of Decision tree classifier on cross-validation: 0.81
[[21617  3103]
 [ 2978  4863]]
Accuracy of Random Forest classifier on cross-validation: 0.85
[[22914  1806]
 [ 2936  4905]]
Accuracy of Logistic regression classifier on cross-validation: 0.85
[[23029  1691]
 [ 3151  4690]]
Accuracy of SVM classifier on cross-validation: 0.86
[[23257  1463]
 [ 3225  4616]]
```

Les SVM sont des modèles efficaces pour les tâches de classification binaire, bien que par défaut ils ne soient pas efficaces pour la classification déséquilibrée ici dans notre cas il a donné de meilleur résultat car la marge de l'hyperplan favorise la classe majoritaire sur les jeux de données déséquilibrés, bien qu'elle puisse être mise à jour pour prendre en compte l'importance de chaque classe.

6. Le meilleur modèle testé lors des configurations précédentes est le SVM ce dernier a donné une accuracy de sur l'ensemble de test est de 0.86 avec le nombre d'erreur de 2309 avec des erreurs majoritairement pour la classe la moins représentée ou les revenus sont supérieure à 50k .