Devoir 1

Abdoul Sadikou (20158628)
Arsène Mitohade (20094950)

Rapport

Dans ce travail, nous devons proposer deux notions de similarité originales et spécifiquement construites pour être utilisées avec MNIST et ADULT. Il faut donc comparer la performance des algorithmes *partition binaire, KNN, PCoA, k-médoïde, en utilisant les notions de similarité proposées et la distance euclidienne. Nous utiliserons pour ce faire, les données du fichier "mnist.csv" et "adult.csv" trouves sur* *https://www.kaggle.com/datasets/oddrationale/mnist-in-csv Et* *https://www.kaggle.com/datasets/sohaibanwaar1203/adultscsv.*
Nous utiliserons pour la complexité des algorithmes, les 1000 premières lignes de données dans les fichiers afin de les tester. Aussi, plusieurs librairies ont été utilisés dans notre code. N'oubliez pas de les installer afin de voir le code compiler correctement.

MNIST

Tout d'abord nous avons procédé à l'extraction des données du fichier Excel. Comme il a été dit plus tôt, compte tenu de la taille des données dans nos fichiers nous allons utiliser les 1000 premières données de notre fichier "mnist.csv".

1. **Traitements des données**

Après avoir récupéré les données dans des tableaux, nous avons procédé à la normalisation en divisant les valeurs des pixels par 255.0 au lieu de 255 pour convertir en float et remettre chaque image dans le bon format de la phase de construction de notre notion de similarité.

2. **Notion de similarité**

L'idée est d'appliquer des opérations sur les images comme la translation et/ou la rotation, à ajouter ces images à notre jeu de données et ensuite utiliser la distance euclidienne pour calculer la distance euclidienne minimum. Plus la distance se rapproche de 0 plus les deux images sont similaires.
Ainsi, pour notre data augmentation nous avions créé une:

- rotation (image, angle), qui prend une image de taille 28 par 28 et fait une rotation d'angle degré.
- translation (image, direction), qui prend une image de taille 28 par 28 et retourne cette image d'un pixel d'une direction.
- setImagesRotate(): augmente notre jeu de données en ajoutant pour chaque image un tableau de 9 images données dans un angle. On reshape et on flatten les images pour les remettre en une dimension.
- setImagesTranslate(): augmente notre jeu de données pour chaque image ayant subi une rotation, nous procédons à 4 translations dans les directions haut, bas, gauche et droite, plus l'image elle-même. On reshape et on flatten les images pour les remettre en une dimension.

Ensuite, pour la **notion de similarité,** avec la fonction getCustomDistance(num1, num2, matrix). Notre notion de similarité se basera essentiellement sur le concept de dissimilarité en utilisant la distance euclidienne. La façon la plus simple est d'examiner la similarité entre deux images consistant à regarder les différences entre ses points zone par zone. Notre but est donc d'essayer de trouver l'image dans notre jeu de données augmenter qui donne la plus petite différence entre les 2 images. Par exemple une image1 et image2 peuvent avoir une distance de 8 mais image1 et image2 translate peuvent avoir une distance de 5 et sachant que image2 a le même label que image2' la classification sera plus optimisée.

Algorithme de la notion de similarité inspirée de cette source
http://www.lemelin-metho.ucs.inrs.ca/wp-content/uploads/1_5.pdf

Étape 1: Tout d'abord on calcule la distance euclidienne minimale entre les images 1 et les images 2 qui ont subi une rotation.
Étape 2: On calcule la distance euclidienne minimale entre les images 1 qui ont subi une rotation et les images 2.
Étape 3: On trouve le minimum de la distance euclidienne minimale entre les distances euclidiennes calcules à l'étape 1 et l'étape 2.
Étape 4: Nous appliquons l'indice de dissimilarité qui n'est rien d'autre que la somme des 2 valeurs diviser par 2. Et ensuite on divise cette somme par 2.

Comparaison de notre similarité

```
Comparaison: Distances eculdiennes donnees par notre notion - distance ecludienne normale.
Test avec le chiffre   3 et 5 respectivement a la position 569 et 211 du test_dataSet
Ceci est la distance eucludienne: 11.5758
Notion similarite proposee: 11.2879
==================
==================
Test avec le chiffre   6 et 1 respectivement a la position 366 et 918 du test_dataSet
Ceci est la distance eucludienne: 10.198
Notion similarite proposee: 9.5381
==================
==================
Test avec le chiffre   7 et 7 respectivement a la position 684 et 0 du test_dataSet
Ceci est la distance eucludienne: 11.8743
Notion similarite proposee: 11.4884
==================
==================
Test avec le chiffre   1 et 8 respectivement a la position 251 et 998 du test_dataSet
Ceci est la distance eucludienne: 13.1909
Notion similarite proposee: 13.0574
==================
==================
Test avec le chiffre   1 et 7 respectivement a la position 504 et 411 du test_dataSet
Ceci est la distance eucludienne: 7.4833
Notion similarite proposee: 6.5463
==================
==================
Test avec le chiffre   4 et 7 respectivement a la position 238 et 554 du test_dataSet
Ceci est la distance eucludienne: 11.0
Notion similarite proposee: 9.6347
==================
==================
Test avec le chiffre   0 et 8 respectivement a la position 28 et 260 du test_dataSet
Ceci est la distance eucludienne: 12.0
Notion similarite proposee: 11.7897
==================
==================
Test avec le chiffre   5 et 2 respectivement a la position 751 et 823 du test_dataSet
Ceci est la distance eucludienne: 13.8564
Notion similarite proposee: 13.2611
==================
==================
Test avec le chiffre   7 et 6 respectivement a la position 144 et 568 du test_dataSet
Ceci est la distance eucludienne: 11.1803
Notion similarite proposee: 10.8828
==================
==================
Test avec le chiffre   3 et 0 respectivement a la position 912 et 871 du test_dataSet
Ceci est la distance eucludienne: 13.3041
Notion similarite proposee: 12.4496
==================
==================
```

```
Test avec le chiffre   7 et 4 respectivement a la position 810 et 85 du test_dataSet
Ceci est la distance eucludienne: 13.1149
Notion similarite proposee: 12.6416
==================
==================
Test avec le chiffre   5 et 5 respectivement a la position 779 et 478 du test_dataSet
Ceci est la distance eucludienne: 9.2736
Notion similarite proposee: 9.0
==================
==================
Test avec le chiffre   2 et 3 respectivement a la position 278 et 173 du test_dataSet
Ceci est la distance eucludienne: 12.0416
Notion similarite proposee: 11.8517
==================
==================
Test avec le chiffre   0 et 9 respectivement a la position 997 et 966 du test_dataSet
Ceci est la distance eucludienne: 13.7113
Notion similarite proposee: 13.3229
==================
==================
Test avec le chiffre   7 et 1 respectivement a la position 666 et 388 du test_dataSet
Ceci est la distance eucludienne: 9.434
Notion similarite proposee: 8.6368
==================
==================
Test avec le chiffre   1 et 7 respectivement a la position 196 et 950 du test_dataSet
Ceci est la distance eucludienne: 10.0
Notion similarite proposee: 9.377
==================
==================
Test avec le chiffre   8 et 8 respectivement a la position 691 et 465 du test_dataSet
Ceci est la distance eucludienne: 13.0384
Notion similarite proposee: 12.0819
==================
==================
Test avec le chiffre   8 et 7 respectivement a la position 232 et 482 du test_dataSet
Ceci est la distance eucludienne: 12.2066
Notion similarite proposee: 12.1655
==================
==================
Test avec le chiffre   1 et 7 respectivement a la position 430 et 328 du test_dataSet
Ceci est la distance eucludienne: 8.8318
Notion similarite proposee: 8.447
==================
==================
Test avec le chiffre   1 et 9 respectivement a la position 276 et 639 du test_dataSet
Ceci est la distance eucludienne: 9.4868
Notion similarite proposee: 9.0205
==================
==================
```

```
==================
Test avec le chiffre   4 et 5 respectivement a la position 121 et 352 du test_dataSet
Ceci est la distance eucludienne: 12.7279
Notion similarite proposee: 12.5688
==================
==================
Test avec le chiffre   2 et 2 respectivement a la position 875 et 868 du test_dataSet
Ceci est la distance eucludienne: 9.5917
Notion similarite proposee: 9.2399
==================
==================
Test avec le chiffre   8 et 6 respectivement a la position 146 et 737 du test_dataSet
Ceci est la distance eucludienne: 14.3527
Notion similarite proposee: 13.4047
==================
==================
Test avec le chiffre   3 et 0 respectivement a la position 574 et 126 du test_dataSet
Ceci est la distance eucludienne: 12.083
Notion similarite proposee: 11.8294
==================
==================
Test avec le chiffre   3 et 0 respectivement a la position 938 et 567 du test_dataSet
Ceci est la distance eucludienne: 10.4881
Notion similarite proposee: 10.244
==================
==================
Test avec le chiffre   3 et 2 respectivement a la position 453 et 35 du test_dataSet
Ceci est la distance eucludienne: 12.7279
Notion similarite proposee: 11.9104
==================
==================
Test avec le chiffre   1 et 4 respectivement a la position 889 et 56 du test_dataSet
Ceci est la distance eucludienne: 12.2474
Notion similarite proposee: 11.1125
==================
==================
Test avec le chiffre   3 et 0 respectivement a la position 334 et 297 du test_dataSet
Ceci est la distance eucludienne: 12.4499
Notion similarite proposee: 12.1033
==================
==================
Test avec le chiffre   1 et 4 respectivement a la position 46 et 511 du test_dataSet
Ceci est la distance eucludienne: 9.6954
Notion similarite proposee: 9.5131
==================
==================
Test avec le chiffre   8 et 1 respectivement a la position 391 et 716 du test_dataSet
Ceci est la distance eucludienne: 11.3137
Notion similarite proposee: 10.9078
==================
==================
```

```
==================
Test avec le chiffre   1 et 5 respectivement a la position 388 et 692 du test_dataSet
Ceci est la distance eucludienne: 10.198
Notion similarite proposee: 9.2447
==================
==================
Test avec le chiffre   6 et 3 respectivement a la position 786 et 334 du test_dataSet
Ceci est la distance eucludienne: 11.4018
Notion similarite proposee: 10.9087
==================
==================
Test avec le chiffre   4 et 2 respectivement a la position 968 et 892 du test_dataSet
Ceci est la distance eucludienne: 11.2694
Notion similarite proposee: 11.2025
==================
==================
Test avec le chiffre   0 et 3 respectivement a la position 324 et 938 du test_dataSet
Ceci est la distance eucludienne: 10.4403
Notion similarite proposee: 10.2451
==================
==================
Test avec le chiffre   8 et 8 respectivement a la position 355 et 787 du test_dataSet
Ceci est la distance eucludienne: 12.6095
Notion similarite proposee: 11.9556
==================
==================
Test avec le chiffre   4 et 7 respectivement a la position 428 et 413 du test_dataSet
Ceci est la distance eucludienne: 12.2882
Notion similarite proposee: 11.7677
==================
==================
Test avec le chiffre   6 et 3 respectivement a la position 341 et 925 du test_dataSet
Ceci est la distance eucludienne: 10.4881
Notion similarite proposee: 10.1976
==================
==================
Test avec le chiffre   7 et 9 respectivement a la position 468 et 773 du test_dataSet
Ceci est la distance eucludienne: 11.6619
Notion similarite proposee: 11.1574
==================
==================
```

Interprétation: En comparant les résultats de notre notion de similarité avec celle de la distance euclidienne, les notre donne un résultat meilleur à celle de la distance euclidienne. Cela dit grâce à l'augmentation des données et le principe de la dissimilarité, on arrive à déterminer à quel point les images se

ressemblent en prenant tout simplement le minimum de leur différence. Aussi, puisque notre notion de similarité se base sur les données augmentées pour calculer la distance, elle identifie une image qui donne la distance minimale avec la seconde image. Par ce fait on optimise la capacite à classifier les images avec nos divers algorithmes.

Étude sur les algorithmes

Ces algorithmes reçoivent en paramètre une matrice de distance, donc la *matriceDistanceSimilarite* pour notre notion de similarité et la matrice MatriceEucludienne pour la distance euclidienne.
Les fonctions getMatriceSimilarite et getMatriceDistEcludienne créent ses matrices et les exportent dans des fichiers csv.

- Utilisation de v-mesure comme métrique

Nous avons choisi v-mesure comme métrique pour les algorithmes afin de mesurer la performance de celles-ci car elle tient compte de l'homogénéité et de la complétude.

- K-médoïdes

Code se trouvant dans le fichier MINST/k_medoides.py

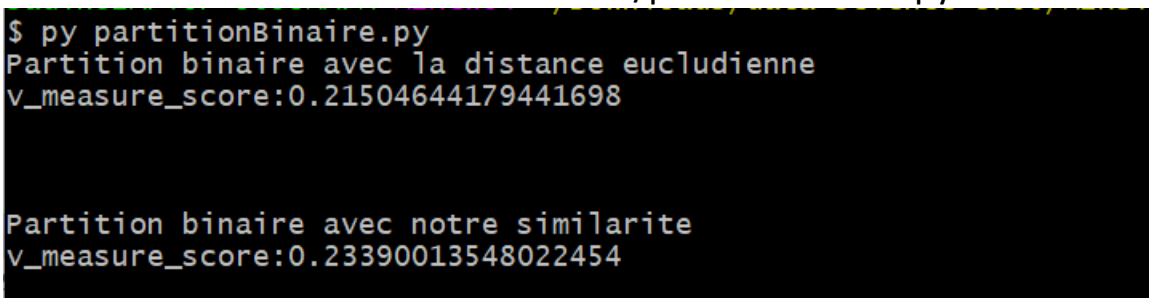
```
$ py k_medoides.py
Avec la distance Eucludienne
v_measure_score: 0.007829004860863O2
Avec la notion de similarite
v_measure_score: 0.009199940251879002
```

On peut remarquer que dans le cas de notre notion de similarité utilise pour construire la matrice de distance on obtient une v-mesure qui est beaucoup

mieux compare à la distance euclidienne. Ici on peut donc conclure que notre notion améliore la performance de k-médoïde.

- Partition binaire

Code se trouvant dans le fichier MINST/partitionBinaire.py

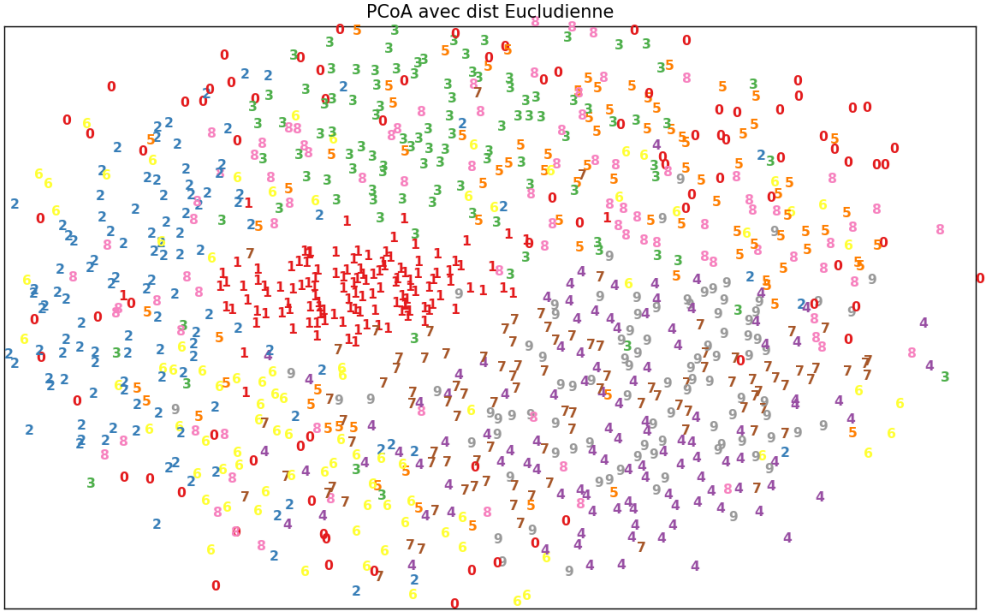


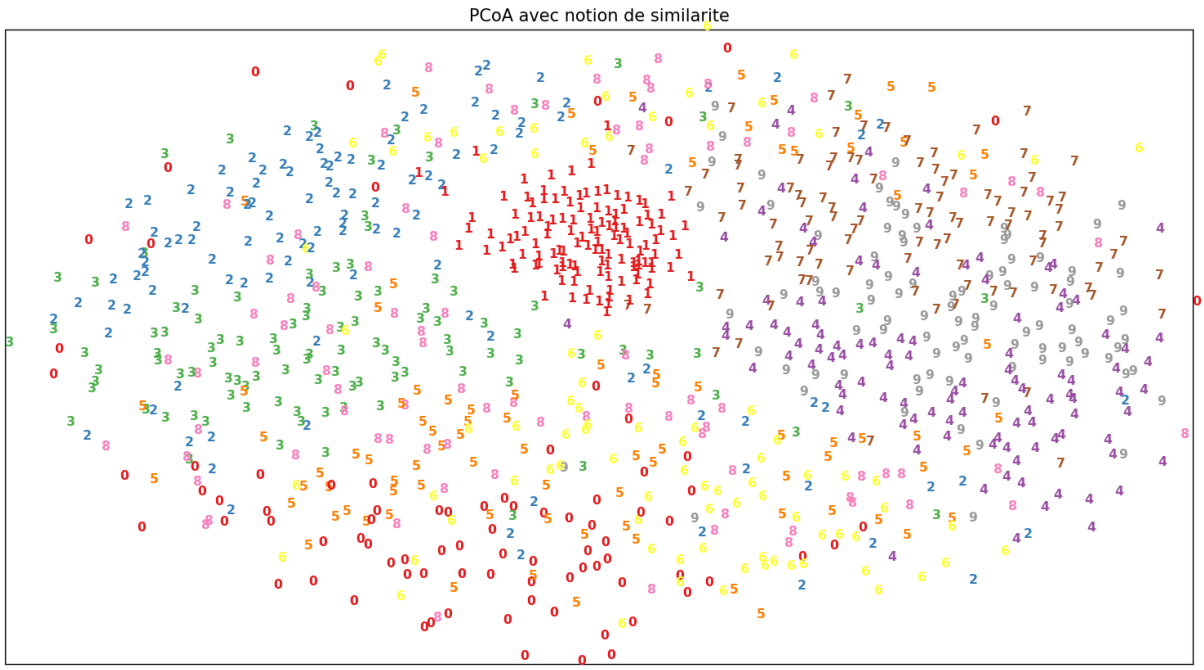
Même si ici le score de notre notion de similarité n'est pas meilleur, elle est quand même mieux que celle avec la distance euclidienne. Ainsi notre notion de similarité améliore les performances de la partions binaire.

- PCoA

L'analyse des coordonnées principales (Multidimensional Scaling, MDS) est une méthode pour explorer et visualiser les similitudes ou les dissemblances des données. Nous allons donc utiliser la méthode MDS pour comparer le cas d'utilisation de notre notion de similarité a la distance euclidienne.
Code se trouvant dans MNIST/pcoa.py

Figure 1

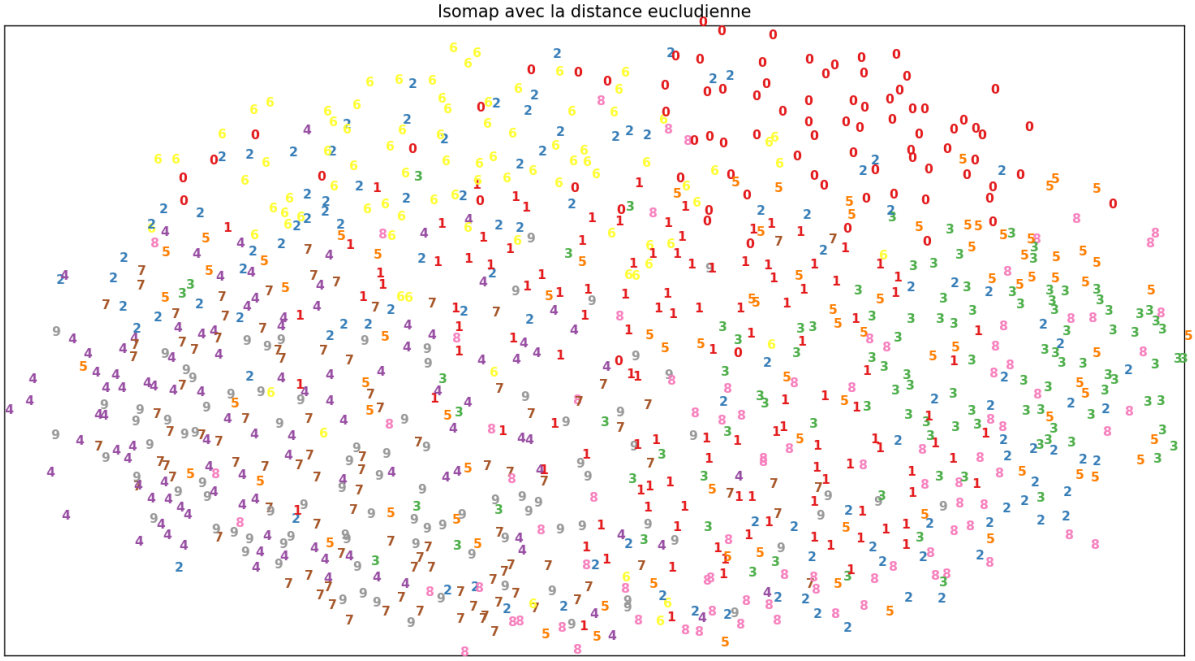


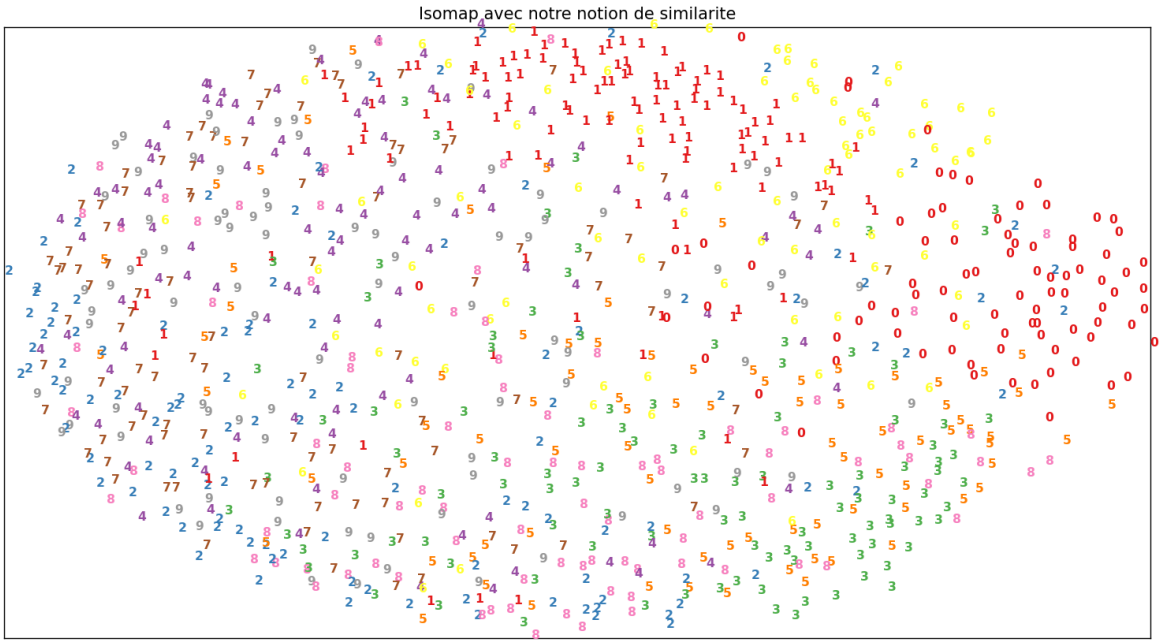


Ainsi de ces différents graphes, on constate directement que notre notion permet de mieux regrouper les “1” mieux qu’avec la distance euclidienne. Également, on remarque que les autres chiffres de mêmes labels sont regroupés plus ou moins de manière plus condenses avec notre notion de similarité qu’avec la distance euclidienne. On en déduit ainsi que notre notion offre une meilleure visualisation.

- Isomap

Code se trouvant dans le fichier MNIST/isomap.py
Quelques changements ont été effectués avant d’utiliser les matrices de distance. Tout d’abord à l’aide de la matrice de distance, pour chaque point, il faut trouver la distance des k plus proches voisins et pour le reste des distances on initialise avec “infini”. Ensuite, il faut exécuter Dijkstra et pour finir exécuter MDS sur la matrice trouvée.
Voici les résultats pour avec la distance euclidienne et la notion de similarité.


Isomap avec la distance eucludienne

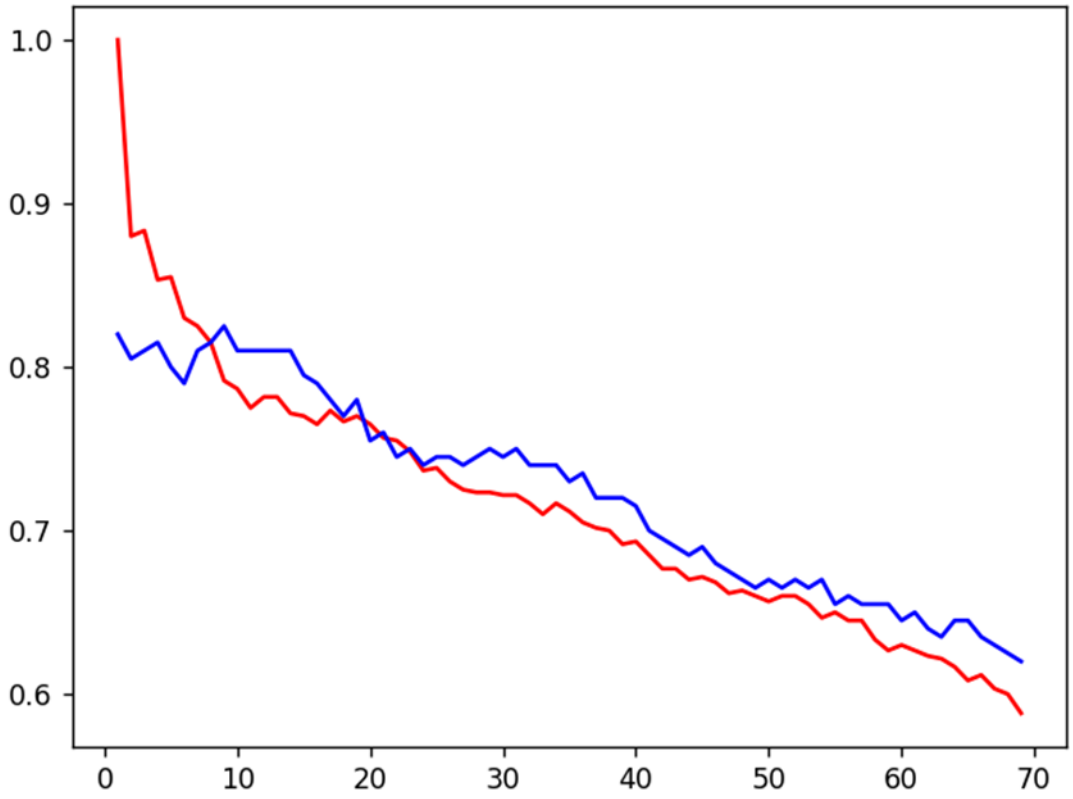


Pour l'algo d'Isomap, le choix des k-voisins doit être fait de manière qu'il ne soit ni trop petit, ni trop grand. Nous avons donc choisi k-voisins = 20. D'après les graphes les groupes sont mieux formes pour notre notion de similarité que pour la distance euclidienne. Les "1" sont légèrement mieux regroupes pour notre notion et ils se chevauchent. Les "4", les"7" et les "9" sont légèrement mélangés ensemble pour le cas qui utilise la distance euclidienne, mais pour notre notion ils sont mieux regroupés ainsi que les "0".

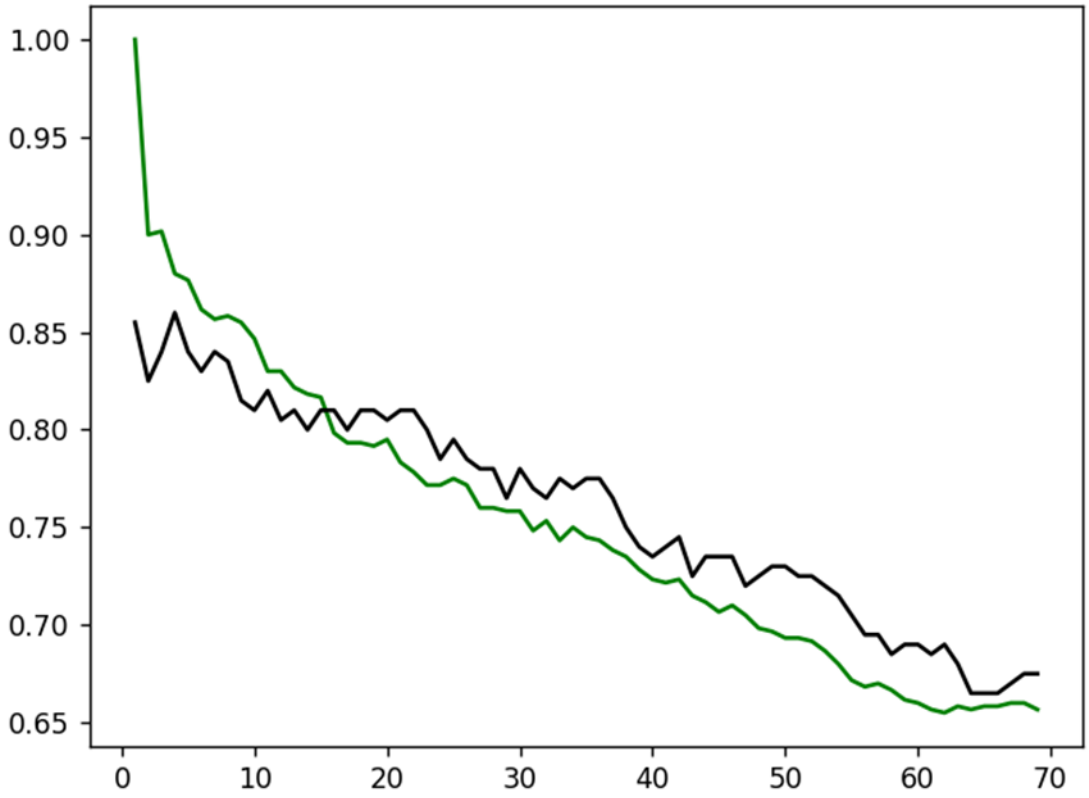
- Knn

Plusieurs tests ont été fait pour avoir une meilleure valeur pour le k. Si on utilise un échantillon de 1000 lignes du jeu de données, on obtient un k très petit mais sensible aux données qui sont bruitées. On aussi augmenter la taille de notre échantillon a 3000 et a 6000 mais le résultat du k demeure inchangée. Aussi on a eu à séparer le jeu de données (1000 lignes) pour pouvoir entrainer notre algorithme knn avec 600 lignes, 200 lignes sur les 1000 lignes du jeu de données on servit à la validation et les 200 lignes

restantes on servit à tester notre algorithme. Les graphes ci-dessous illustre nos résultats.

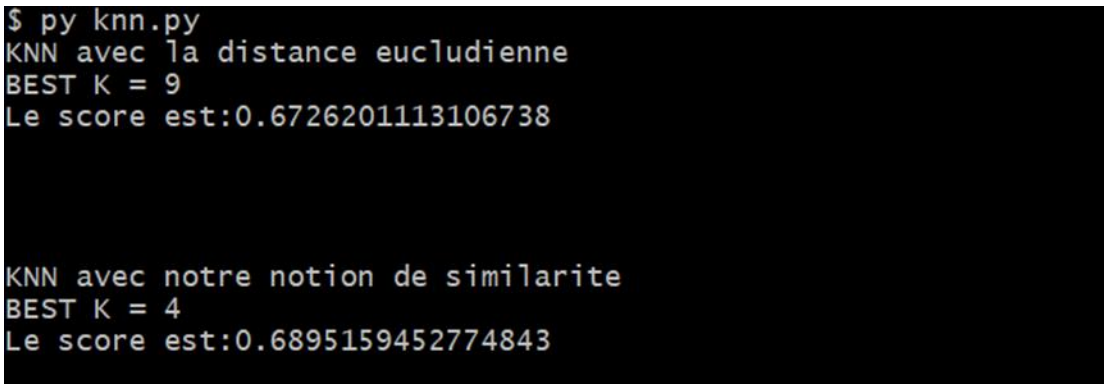


Ici, nous utilisons notre algorithme avec la distance euclidienne. La courbe en rouge est celle de notre test et celle en bleue pour notre validation.



Pour ce graphe, nous utilisons notre notion de similarité. La courbe en verte est celle de notre test et celle en verte notre validation.

On remarque que sur chaque figure que toutes les courbes sont décroissantes ainsi on ne peut pas aisément conclure si notre notion de similarité est meilleure que la distance euclidienne. Donc on choisit un k légèrement grand pour tester encore notre algorithme. On choisit k= 20. Voici les résultats trouvés.



On peut donc déduire que notre notion de similarité améliore légèrement la performance de l'algorithme knn.

Forces et faiblesses de notre notion de similarité

Notre notion de similarité améliore la performance de l'algorithme PCoA mais tout de même moins performant que Isomap, ce qui peut être compréhensif car le prétraitement de ces deux derniers sont effectués avant l'application de MDS.

Et si on compare pour chaque algorithme nos résultats utilisant la notion de similarité avec ceux utilisant la distance euclidienne, notre notion donne une meilleure performance.

ADULT

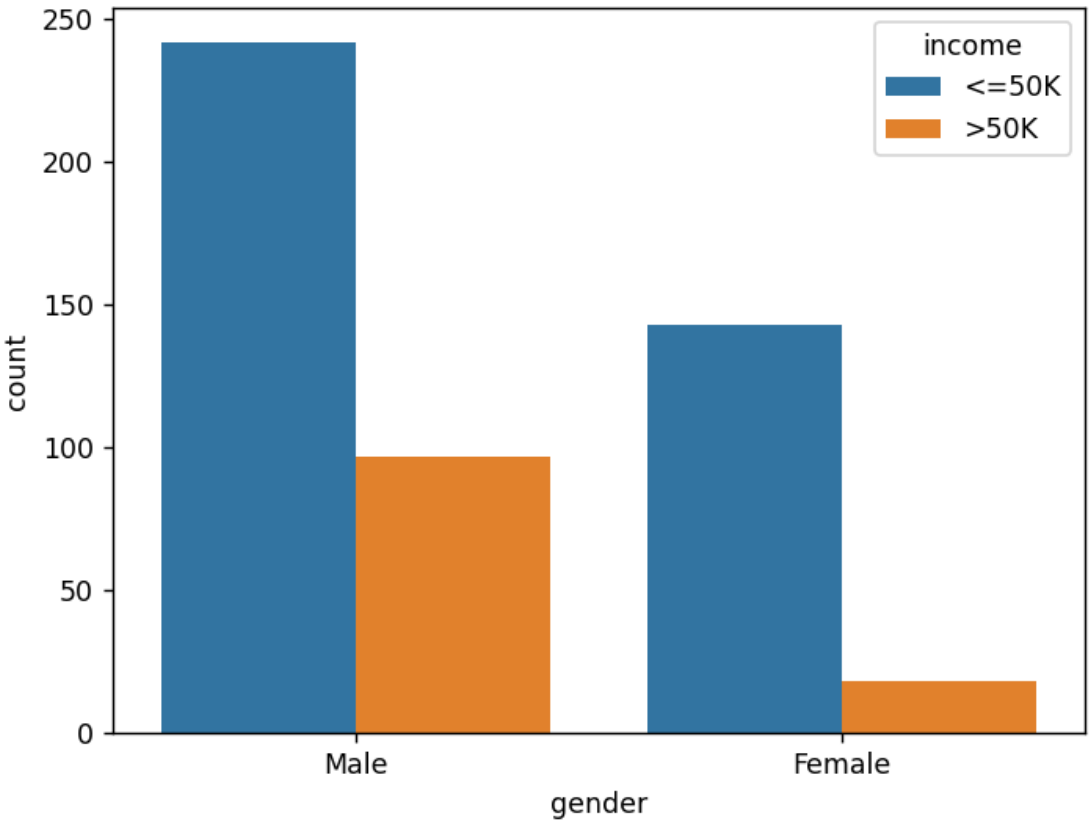
Notre notion de similarité avec le jeu de données *ADULT se* base sur les connaissances apprises dans le cours. Nous avions voulu représenter nos jeux de données sous la forme d'une matrice ou les lignes sont

indépendantes et identiquement distribuées ***iid.*** Chaque colonne représentera une caractéristique du jeu de données et les lignes seront les exemples.

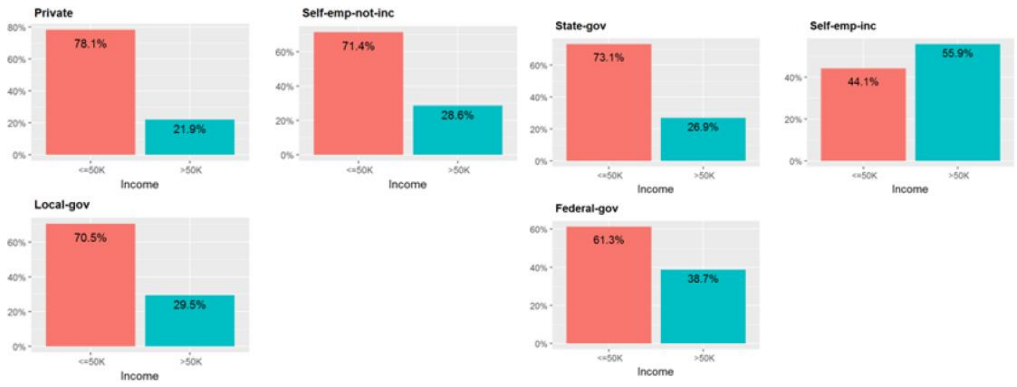
Une première étape du processus a été de nettoyer les données. En effet, nous avions choisi de faire le nettoyage de manière judicieuse pour rendre l'apprentissage de la machine plus simple et rapide. Premièrement, nous avions réduit la taille de nos données a 500 au lieu de 48842 car utilise tout le jeu de données consommait beaucoup de mémoire que nous n'avions pas à disposition et utiliser 500 lignes de données nous donner une exécution plus rapide. Ensuite, nous avions cherché les lignes de données qui contenait des valeurs manquantes caractérise par un "?" et on les a remplacés par ***"np.nan".*** Ainsi il devient plus facile d'utiliser la fonction *"dropna"* de la librairie pandas pour se débarrasser des lignes qui contiennent des valeurs NaN. Sachant que notre étude se portera sur le fait de savoir si une personne gagne un salaire plus ou moins de 50000$, nous nous sommes débarrassés de colonnes non essentielles telles que l'âge, la colonne sur le pays de naissance… Rendue à cette étape, on se retrouve uniquement avec les colonnes qu'on aura besoin ou toutes les lignes n'ont pas de valeurs manquantes. Aussi nous avions donne des poids à certaines colonnes et on expliquera plus tard pourquoi nous avions fait cela.

<u>Analyse des résultats</u>

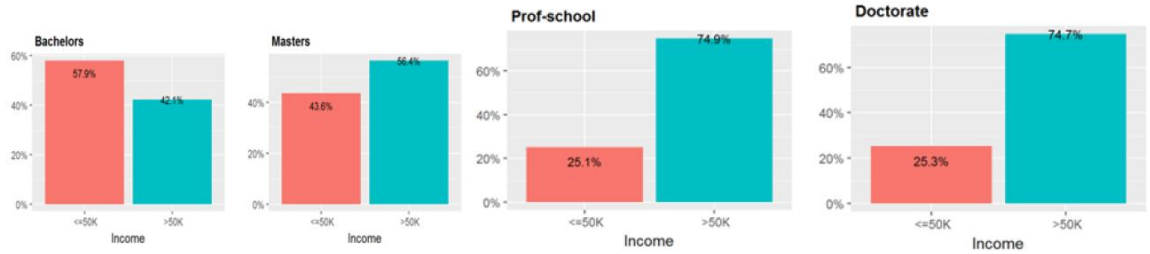
Nous avions utilisé des librairies de Python nous permettant de visualiser à l'aide de graphes les données qui nous ont permis de mieux analyser les résultats. Pour commencer sur l'échantillon de 500 personnes, nous avions à peu près 242 hommes avec un salaire de moins de 50000$ et 96 hommes avec un salaire plus de 50000$. Pour les femmes nous avions près de 143 femmes avec un salaire de moins de 50000$ et près de 17 femmes ayant plus de 50000$. Ces résultats peuvent être vues comme suit:



On remarque que plus de 74% des participants sont des hommes. De plus on estime que 30% des hommes font un salaire de plus de 50000$ et 14% des femmes font un salaire de plus de 50000$. Cela nous a mène à dire qu'il y'a 75% plus d'hommes que de femmes qui font un tel salaire. Cela peut être exprime par le fait que on est décidé de donner un poids aux hommes de 0.75 et aux femmes 0.25 sur la colonne gender. Cette manière de faire nous a permis d'avoir une matrice contenant ces valeurs-là. Nous verrons comment on s'est servi de cela pour améliorer notre similarité.



On remarque ici que seul *“self-emp-inc”* dans la colonne *workclass* sont les personnes ayant cette valeur dans leurs lignes de donnees suspecible d’avoir un salaire plus de 50000$.



Avoir un plus qu’un *“Bachelor” donne aussi une probabilité d’avoir un salaire plus de 50000$, mais considérant quand même que on a une probabilité*

presque égale d'avoir soit un salaire plus de 50000$ et moins de 50000$ avec un "Bachelor".

Bases sur plusieurs corrélations comme les deux identifies plus hauts, nous avions modifié les valeurs de certaines de nos colonnes, qui sont les colonnes "gender, race, marital-status, worklass, education, occupation". La raison pour laquelle on a décidé de modifier ces colonnes particulièrement était le constat de chiffres significatifs qui revenaient comme par exemple, 75% des personnes ayant un doctorat ont une grande probabilité d'avoir un salaire plus de 50000$. Sur cette logique, nous avions construit une notion de similarité. Pour mieux comprendre l'approche utilise, un exemple sur le fait que 75% des personnes ayant un doctorat favorise grandement leur chance d'avoir un salaire voulu, alors que la colonne sur l'éducation nous avions remplacé la valeur doctorat par la valeur 0.75 représentant la probabilité que nous recherchons. Pour généraliser et classifier toutes les colonnes et les différentes valeurs possibles, nous avons catégorise les tranches suivantes: 0.75 pour probable, 0.5 pour moins probable, 0.4 pour improbable et 0.2 pour hors sujet.
Ainsi nous avions généraliser les données en les arrondissant le plus logiquement possible en se basant sur les graphes et les statistiques observes. Cela a rendu notre *"data frame"* avec les colonnes contenant uniquement des valeurs entre 0.2 et 0.75. Par la suite nous avions additionne pour chaque individu toutes ces valeurs pour avoir une valeur score et ainsi on se retrouve avec une matrice contenant le score de chaque individu dans notre jeu de donnees. Maintenant, on calcule une notion de distance sur ces valeurs. Pour cette partie nous avions pris la décision de diviser nos scores en 2 parties. Bien-sûr, en ayant pris le temps de choisir avec attention un chiffre logique comme référence. Une personne ayant un score plus de 5.0 a plus de chance de gagner un salaire supérieur à 50000$ et l'inverse pour une personne ayant une probabilité de gagner un salaire inferieure a 50000$. 5.0 étant le chiffre de référence. Cela nous donne en sortie une matrice avec des 1 et de 0; 1 lorsque la condition de score supérieure à 5.0 est respectée et 0 pour l'autre condition. Donc en tout ce qu'on a vraiment tente de faire c'est de réduire le plus possible toutes les

donnees. Notre hypothèse est que **notre notion de similarité que nous avions pensée, allait e grande majorité fonctionne sur les différents algorithmes sur lesquels il fallait tester. Malheureusement, des problèmes de compilation au niveau du code et juste k-nn donner un résultat trivial.** Cependant, nous allons essayer d'expliquer les forces et les faiblesses du mieux possible en se basant sur notre notion de similarité.

Les forces et les faiblesses

Puisque notre notion de similarité se base principalement sur les probabilités des différentes catégories, penchant sur le fait si une personne gagne un salaire de plus ou moins que 50 000$, cela transforme nos données catégoriques ainsi que numériques en pourcentage, et lorsque on a additionné pour donner le score de chaque individu comme expliqué plus haut, nous donne un pourcentage de probabilité que cet individu ait ou pas un salaire de 50 000$ et plus. Un autre point positif, c'est le fait que nous avons réussi avec beaucoup de traitement des données et d'analyse, que les diverses possibilités des données que nous avions, ont fini par être traiter d'une manière à seulement avoir deux valeurs numériques possibles.

Cette manière de procéder semble intuitive et logique puisqu'elle provient des statistiques observées, cependant, puisque notre étude fait un peu de la généralisation sur les données en les arrondissant à une des catégories tel que (0.75, 0.5, 0.4 ou 0.2), cela va affecter nos résultats de ne pas avoir une très haute précision puisque certaines valeurs seront pénalisées puisqu'elles vont probablement être arrondie vers le bas dans certains cas.

Ressources:

https://rstudio-pubs-static.s3.amazonaws.com/265201_a7ba37d372cb4206a8412ae788dd5d9d.html#312_the_variable_%E2%80%9Csex%E2%80%9

http://www.lemelin-metho.ucs.inrs.ca/wp-content/uploads/1_5.pdf