Fouille de données
Travail Pratique N°1

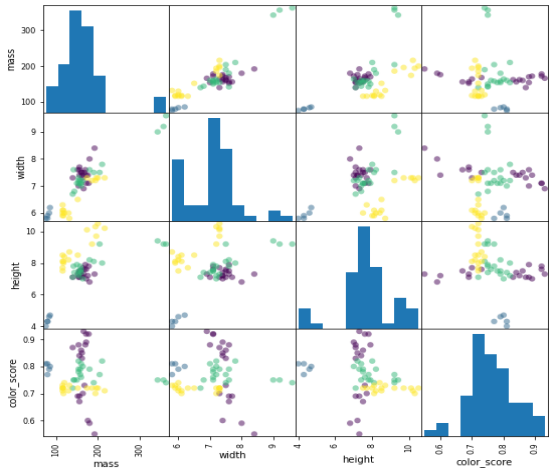
UE Data analytics big data

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

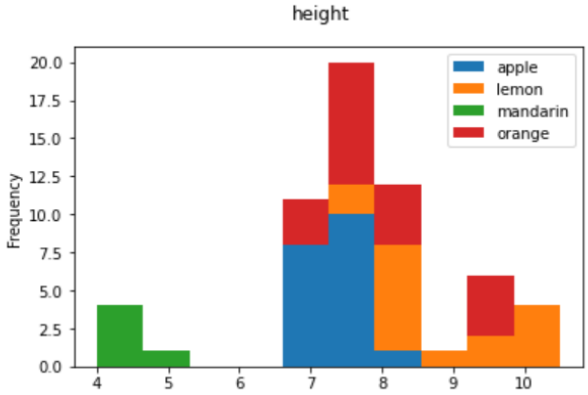
Analyse descriptive des données

- **Combien y a-t-il d'attributs ? Que représentent-ils ? Quel est leur type ? Quelle est la classe à prédire ? Combien y a-t-il d'instances dans le fichier ? Les classes sont-elles équilibrées quant au nombre d'instances ?**
  - Il y a 7 attributs, qui représentent les caractéristiques permettant de décrire les fruits .
    - fruit_label => quantitative
    - fruit_name => qualitative
    - fruit_subtype => qualitative
    - mass width=> quantitative
    - height=> quantitative
    - color_score=> quantitative
  - Les types des attributs :
    - fruit_label : int64
    - fruit_name : object
    - fruit_subtype : object
    - mass : int64
    - width : float64
    - height : float64
    - color_score : float64
  - La classe à prédire c'est la classe auquel chaque instance de fruit appartient [Orange, Citron, Mandarine, Pomme].
  - Il y a 59 instances, le dataset est assez petit.
  - Les classes ne sont pas équilibrées puisque la classe mandarine est moins représentée que les autres avec seulement 5 instances.

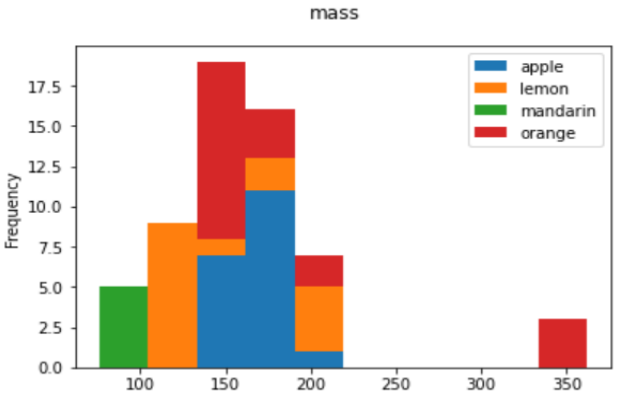
- **Examinez les données de manière graphique à l'aide d'un diagramme de dispersions (fonction plotting.scatter_matrix() de pandas) et d'histogrammes montrant la répartition selon un attribut et la classe.**
  - Ce diagramme nous donne une information sur la corrélation entre attributs et comment les variables interagissent entre elle, par exemple la **mass, width et height** on forte corrélation les instance sont tous réunis au même endroit, mais **mass, width et height ne sont pas du tout corrélé avec color_score on peut remarquer que les groupes de fruits sont éparpillés.**



- Dans les histogrammes, reçu en sortie nous pouvons voir que le groupe des mandarines est facilement distinguable puisque seulement à partir de la hauteur on peut déduire leur classe.



Par exemple pour la **masse** on ne peut pas déduire beaucoup d'information sauf pour les mandarines quand la masse est en dessous de 100 ou pour les oranges quand la **masse** est supérieure à 300.



Prétraitement

- Appliquez à partir des données originelles deux variantes de discrétisation : une première reposant sur des intervalles de même taille (equal-intervaled avec la fonction cut() de pandas) ou de mêmes effectifs (equal-sized bins avec qcut()). En affichant les historiques correspondant à chacune des deux configurations à l'aide de plot.bar(), comparez les histogrammes obtenus pour l'attribut « mass ».
  - Histogramme qcut et cut :

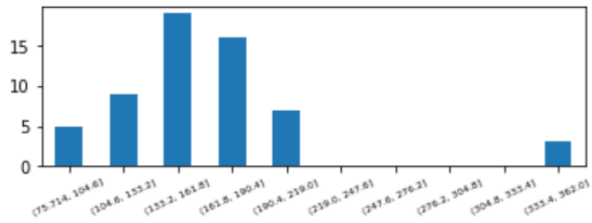


Figure: Histogramme cut

  - On peut voir qu'avec le **cut** la largeur d'intervalle est toujours la même en respectant la valeur affectée à bin, même si des intervalles ou il n'y a pas de données ils sont représentés quand même.

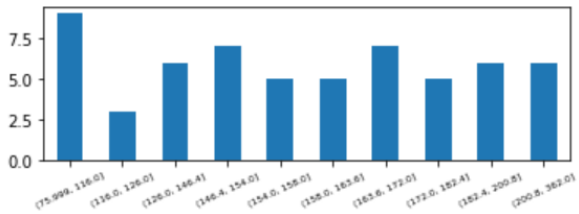


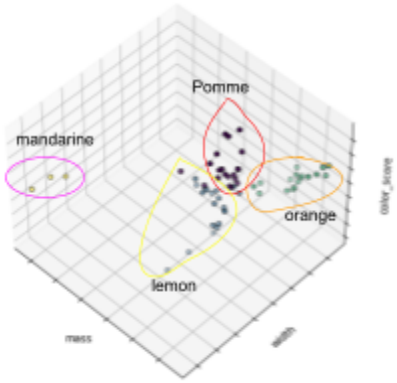
Figure: Histogramme qcut

  - Le histogramme du **qcut** diffère du cut dans qcut, le nombre d'éléments dans chaque bac sera à peu près le même, mais cela se fera au prix de largeurs d'intervalle de tailles différentes.
    puisque les intervalles où on n'a pas de données seront ignorés dans le graphe contrairement que dans le cut normal.

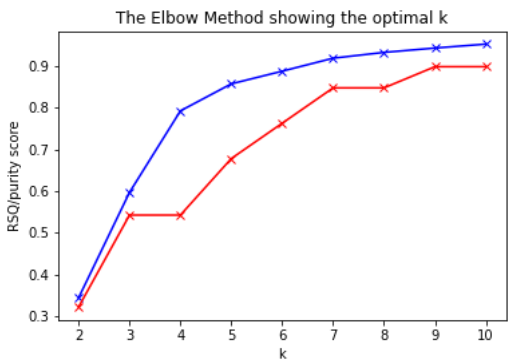
- Reprenez les données originelles. Quel est l'effet du filtre de de normalisation preprocessing.MinMaxScaler de scikit-learn ? Dans quel cas d'utilisation est-il préférable d'appeler la fonction transform( ) de cet objet plutôt que fit_transform() ?
  - MinMaxScaler permet de normaliser les valeurs afin qu'elles soient toutes entre 0 et 1.
  - fit_transform(), permet de lancer l'apprentissage ensuite de les transformer. transform(), va transformer les valeurs en valeurs normées.
  - fit_transform() on l'utilise lors de l'apprentissage avant de transformer, plus particulièrement pour la première fois quand je normalise.Et transform(), lors du teste quand j'aurai déjà appris à normaliser.

Clustering

- Ouvrez le fichier de données en appliquant un filtre de normalisation. Appliquez KMeans sur ces données en fixant successivement le nombre de clusters à 3, 4 et 5. Examinez manuellement les résultats du clustering en traçant les instances dans une espace en 3 dimensions correspondant aux attributs « mass », « width » et « color_score ». Quelle(s) classe(s) représente chacun des clusters ?
  - En ce qui concerne les **oranges** , on peut voir qu'ils ont une masse élevée, une forte couleur bien prononcé et assez large.
    Les **mandarines** sont représentées par le cluster à gauche éloigné, car il s'agit des fruits avec une masse faible et une valeur de couleurs assez élevée. Les **lemon** sont généralement au centre. Les pommes sont larges (width) mais avec une masse moyenne.

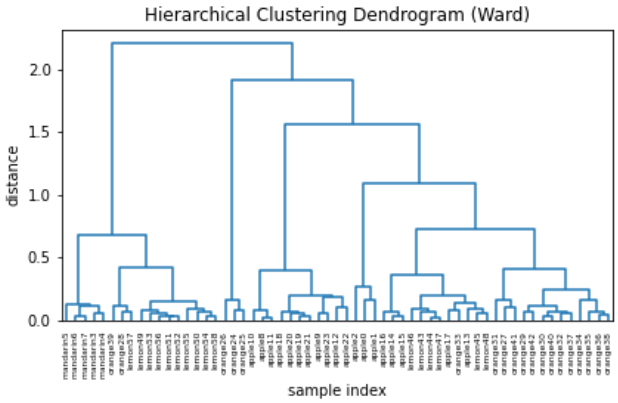


- Question 2 et 3
  - ➢ On a deux courbes représentant purity et Rcarrée.
    - On peut voir que la courbe de R carrée augmente, mais pas de façon considérable puisque à partir du k=7 cette dernière commence à se stabiliser, après cela pas une grande évolution est observée.
    - nombre de groupes qu'on retiendra est 7, car la courbe Rcarrée stagne à partir de 7.
    - Ici, on peut voir qu'il faudrait prendre 7 clusters, car on peut voir qu'à partir de 7, la courbe de pureté commence à diminuer et ne s'améliore pas forcément pour rappelle la pureté des clusters consiste à évaluer la qualité des clusters par rapport au objets étiquetés. Un cluster pur sera un cluster dans lequel tous les objets, dont la classe est connue, appartiennent à une et une seule classe.



- Question 4

  Ce dendrogramme nous permet de voir la composition des différents clusters en fonction de la distance entre eux.
  Si nous prenons en compte la courbe de pureté qui nous a amené à établir un nombre de cluster à 7, nous pouvons en déduire que nous avons besoin d'une distance entre les clusters de 0,5 .



Classement

Question 1

- Quel est le taux de classification obtenu pour chaque méthode ?

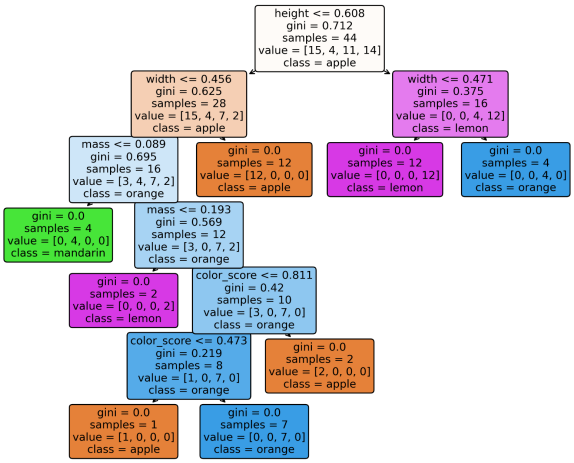
```
Accuracy of Dummy classifier on training set: 0.34
Accuracy of Dummy classifier on test set: 0.27
[[4 0 0 0]
 [1 0 0 0]
 [8 0 0 0]
 [2 0 0 0]]
Accuracy of Naive Bayes classifier on training set: 0.86
Accuracy of Naive Bayes classifier on test set: 0.67
[[4 0 0 0]
 [0 1 0 0]
 [4 0 3 1]
 [0 0 0 2]]
Accuracy of Decision tree classifier on training set: 1.00
Accuracy of Decision tree classifier on test set: 0.87
[[4 0 0 0]
 [0 1 0 0]
 [2 0 6 0]
 [0 0 0 2]]
Accuracy of Random Forest classifier on training set: 1.00
Accuracy of Random Forest classifier on test set: 0.87
[[4 0 0 0]
 [0 1 0 0]
 [1 0 6 1]
 [0 0 0 2]]
Accuracy of Logistic regression classifier on training set: 0.75
Accuracy of Logistic regression classifier on test set: 0.47
[[4 0 0 0]
```

- On peut voir que le meilleur classifieur est : Random Forest classifier et decision tree

  Les taux de classifications sont :
  • Pour la méthode Dummy classifier : 0.27
  • Pour la méthode Naïve Bayes Classifier : 0.67
  • Pour la méthode Decision tree : 0.87
  • Pour la méthode Random Forest Classifier : 0.87
  • Pour la méthode Logistic Regression Classifier : 0.47

La classe pour laquelle le modèle fait le plus d'erreurs est : l'Orange, car on peut voir que sur les différents classifieurs, la ligne des oranges à souvent des erreurs, comment? Les quantités ne sont pas sur la diagonale mais plutôt éparpillées sur les autres colonnes.
Les quantités devraient toutes se trouver dans la diagonale de ligne 3, donc toutes les valeurs hors diagonale correspondent à des erreurs.

- Arbre de décision :
  On peut trouver voir que les fruits sont découpé selon certaines caractéristiques : la hauteur , la largeur et la masse.



Par exemple, si on prend une hauteur en dessous de 0.608, puis ceux dont la largeur est
au-dessus de 0.471 seront soit des oranges ou des lemon selon notre arbre de décision puisque, les différentes décisions possibles sont situées aux extrémités des branches comme on peut le constater ici la valeur des fruits se trouve au niveau des feuilles .

- Question 2 :
  - Résultat cross_validation
    - En utilisant le cross-validation, on peut voir que les scores des différents classifieurs se sont améliorés puisque ce dernier après avoir coupé le dataset sur 5 en alternant entre les **fold** en prenant a chaque fois un pour le test. On peut voir que la méthode la plus performante est : Random Forest classifier avec cross-validation: 0.96

```
Accuracy of Dummy classifier on cross-validation: 0.34 (+/- 0.03)
Accuracy of Naive Bayes classifier on cross-validation: 0.79 (+/- 0.22)
Accuracy of Decision tree classifier on cross-validation: 0.84 (+/- 0.12)
Accuracy of Random Forest classifier on cross-validation: 0.96 (+/- 0.11)
Accuracy of Logistic regression classifier on cross-validation: 0.66 (+/- 0.12)
```

- Question 3
  - GridSearchCV est une méthode d'optimisation des hyperparamètres, c'est-à-dire qu'avec un modèle le classifieur que l'on veut utiliser et un ensemble de données de test , il s'agit d'une méthode permettant de trouver la combinaison optimale d'hyperparamètresé, on lui donne des paramètre différents, afin qu'il réalise son apprentissage dessus, et permet de récu-
  - Ici le modèle s'agit de **LogisticRegression**.
  - La méthode **GSV.best_estimator_.get_params()** nous permet de récupérer les meilleurs paramètres en question.
    Nous avons un grid search pour 2 paramètres seulement C et **Penalty**.
    C et la pénalité dans la régression logistique,
    Une valeur élevée de C indique au modèle d'accorder un poids élevé aux données d'apprentissage et un poids plus faible à la pénalité de complexité.

```
parameters={"C":[-3,0.01,0.05,0.1,0.15,1,3,7,10], "penalty":['l2','l1']}
```

Après avoir lancé GridSearch voilà le résultat :

```
Best score: 0.772
Best parameters set:
  C: 7
  penalty: 'l2'
```

Classement et discrétisation

- Question 1

Lorsqu'on lance la cross validation, sur nos données discrétisées, avec "eqsized_bin", on
obtient les performances suivantes sur le Test

- Accuracy of Dummy classifier on cross-validation : 0.27
- Accuracy of Naive Bayes classifier on cross-validation : 0.80
- Accuracy of Decision tree classifier on cross-validation : 0.73
- Accuracy of Random Forest classifier on cross-validation : 0.67
- Accuracy of Logistic regression classifier on cross-validation : 0.67

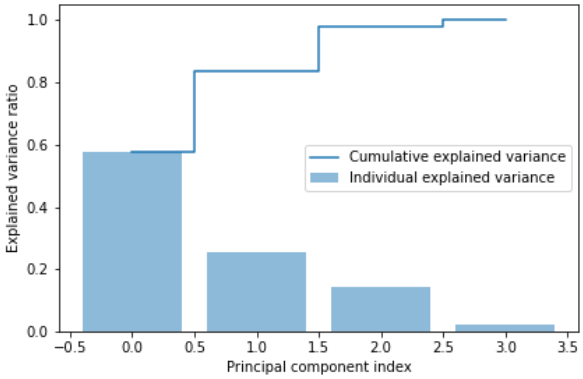
On peut voir que les performances sont meilleures avec les valeurs discrètes pour tous les classificateurs.

ACP

- Question 1
  - Ici, les valeurs propres de notre ACP, sont les variances expliquées par chaque axe principal :

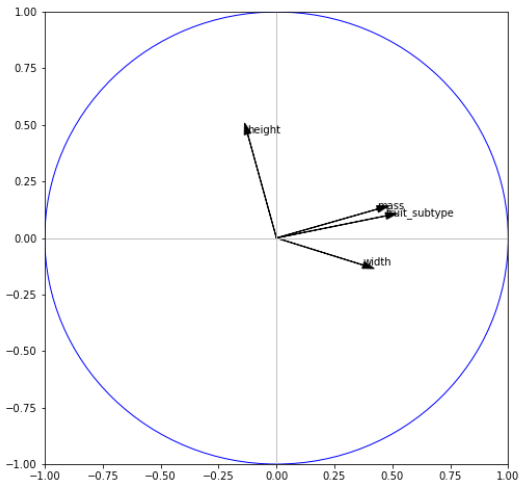
[0.57700681 0.25236307 0.15039404 0.02023607]

  - On remarque que nous avons 4 composantes principales.

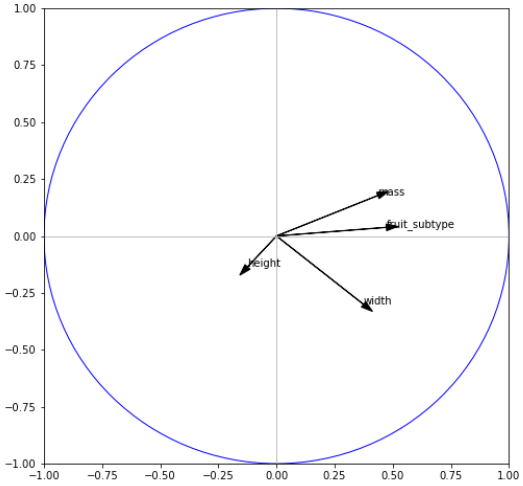


  - Dans le schéma ci- dessus l'axe le plus haut, signifie que plus de valeurs y sont majoritairement représentées.
  - Ainsi, on peut voir qu'avec les 2 premières composantes principales, on peut avoir une assez bonne qualité de représentation des données puisqu' elle renferme le plus de données.

Pour la corrélation des variables avec les axes c'est une information procurée par le cercle de corrélation.



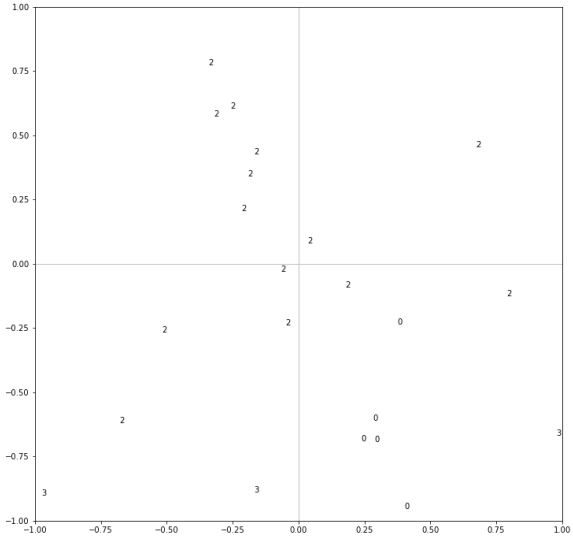
Axe 0 et 1



Axe 0 et 2

On a un cercle entre l'axe 0 et l'axe 1, on peut voir que width, fruit_subtype, et mass sont bien représentés par l'axe 0 et mal représentés sur l'axe 1. En revanche,

on peut voir que height est bien représenté sur l'axe 1 puisque ce dernier est fortement corrélé avec l'axe 1.
En ce qui concerne le cercle de corrélation entre l'axe 0 et l'axe 2, on voit que mass, fruit_subtype et width sont toujours aussi bien représentés sur l'axe 0, cependant ici, on peut voir que width est aussi bien représenté sur l'axe 2.



Pour ce qui est du plot représentant les instances sur les 2 premiers facteurs, on peut voir que seules les classes 0 et 2 sont séparées avec lesquelles on peut constituer des clusters, la classe 3 est totalement mélangée.

- ➢ Question 2

les performances de notre modèle avec les différents classifieurs, en utilisant l'ACP
(Performance avec deux attributs ['mass','width'] n=2 sans ACP)

• Accuracy of Dummy classifier on cross-validation : 0.34
• Accuracy of Naive Bayes classifier on cross-validation : 0.70
• Accuracy of Decision tree classifier on cross-validation : 0.75
• Accuracy of Random Forest classifier on cross-validation : 0.78
• Accuracy of Logistic regression classifier on cross-validation : 0.72

(Performance avec deux composantes n=2 avec ACP)
• Accuracy of Dummy classifier on cross-validation : 0.34
• Accuracy of Naive Bayes classifier on cross-validation : 0.71
• Accuracy of Decision tree classifier on cross-validation : 0.80
• Accuracy of Random Forest classifier on cross-validation : 0.77
• Accuracy of Logistic regression classifier on cross-validation : 0.50
On peut voir que les résultats se sont légèrement améliorés excepté pour la régression logistique.