UNIVERSITÉ DE LORRAINE

# PROJET DE CLUSTERING

## Master 2 Ingénierie Mathématique pour la Science des Données

2022-2023

---

## Jeu de données : Auto-mpg

---

***Réalisé par :***
Marcel MOUDILA

***Enseignant :***
Laurent Bougrain
BOUGRAIN

11 décembre 2022

# Table des matières

**1 Analyse des données** **3**
1.1 Variables prédictives quantitatives . . . . . 3
1.2 Variables prédictives qualitatives . . . . . 4

**2 Régression** **5**

**3 Forêts aléatoires de régression** **8**

**4 Clustering** **9**

**Bibliographie** **11**

# Introduction

1. Title : Auto-Mpg Data
2. Sources : (a) Origin : This dataset was taken from the StatLib library which is maintained at Carnegie Mellon University. The dataset was used in the 1983 American Statistical Association Exposition. (c) Date : July 7, 1993
3. Past Usage : - See 2b (above) - Quinlan,R. (1993). Combining Instance-Based and Model-Based Learning. In Proceedings on the Tenth International Conference of Machine Learning, 236-243, University of Massachusetts, Amherst. Morgan Kaufmann.
4. Relevant Information :
This dataset is a slightly modified version of the dataset provided in the StatLib library. In line with the use by Ross Quinlan (1993) in predicting the attribute "mpg", 8 of the original instances were removed because they had unknown values for the "mpg" attribute. The original dataset is available in the file "auto-mpg.data-original".
"The data concerns city-cycle fuel consumption in miles per gallon, to be predicted in terms of 3 multivalued discrete and 5 continuous attributes." (Quinlan, 1993)
5. Number of Instances : 398
6. Number of Attributes : 9 including the class attribute
7. Attribute Information :
1. mpg : continuous 2. cylinders : multi-valued discrete 3. displacement : continuous 4. horsepower : continuous 5. weight : continuous 6. acceleration : continuous 7. model year : multi-valued discrete 8. origin : multi-valued discrete 9. car name : string (unique for each instance)
8. Missing Attribute Values : horsepower has 6 missing values

Chapitre 1

# Analyse des données

## 1.1 Variables prédictives quantitatives

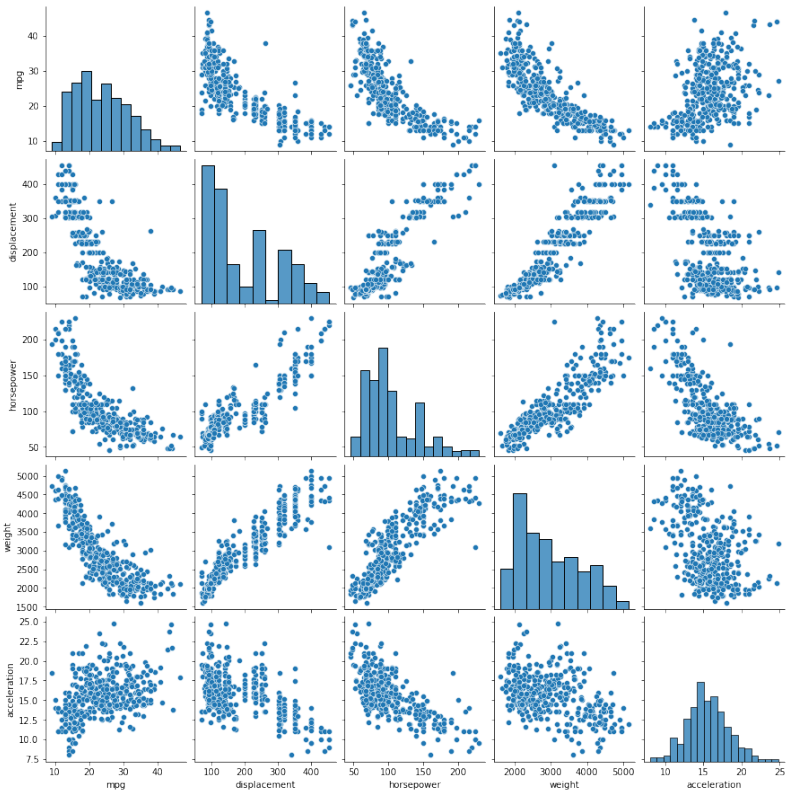



FIGURE 1.1 – relation variables quantitatives et target

la valeur de "mpg" augmente quand la valeur de "acceleration" augmente, On parle de corrélation positive. Pour les autres attributs quantitatifs, la valeur de "mpg" diminue quand leurs valeurs augmentent, on parle de corrélation négative.

| | mpg | displacement | horsepower | weight | acceleration |
|---|---|---|---|---|---|
| **count** | 398.000000 | 398.000000 | 392.000000 | 398.000000 | 398.000000 |
| **mean** | 23.514573 | 193.425879 | 104.469388 | 2970.424623 | 15.568090 |
| **std** | 7.815984 | 104.269838 | 38.491160 | 846.841774 | 2.757689 |
| **min** | 9.000000 | 68.000000 | 46.000000 | 1613.000000 | 8.000000 |
| **25%** | 17.500000 | 104.250000 | 75.000000 | 2223.750000 | 13.825000 |
| **50%** | 23.000000 | 148.500000 | 93.500000 | 2803.500000 | 15.500000 |
| **75%** | 29.000000 | 262.000000 | 126.000000 | 3608.000000 | 17.175000 |
| **max** | 46.600000 | 455.000000 | 230.000000 | 5140.000000 | 24.800000 |

FIGURE 1.2 – statistiques descriptives des variables quantitatives

on voit par cette figure que les valeurs de ces variables ne sont pas distribuées à la même échelle. On voit aussi que count de horsepower est 392, donc on peut déduire qu'il manque 6 valeurs manquantes dans la variable horsepower.

## 1.2 Variables prédictives qualitatives

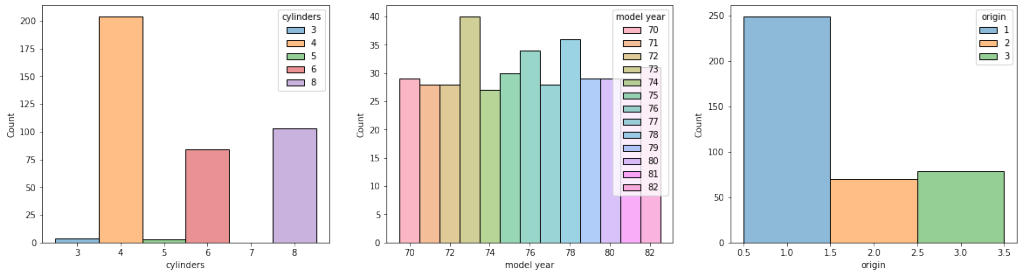



FIGURE 1.3 – effectifs des modalités des variables qualitatives

"cylinders" : les voitures avec 3 cylindres , les voitures avec 5 cylindres sont très peu dans le jeu des données.

"model year" : on constate un bon mélange des années de construction des voitures entre 1970 et 1982 dans le jeu des données.

"origin" : les voitures d'origine le label 1 sont majoritaires dans le jeu de données.

Chapitre 2

# Régression

Dans cette section, nous présentons les résultats pour la tâche de régression. Le modèle utilisé est un arbre de décision de régression construit par la méthode CART. Avant tout, nous avons effectué un prétraitement des données, c'est-à-dire supprimer les valeurs manquantes par la méthode dropna(), écarter la variable "car name", du corpus global des données, car elle a plus de 300 modalités distinctes. Vu la petite taille de nos données, la variable "car name" a peu d'intérêt pour notre apprentissage supervisé.
Une seconde étape a consisté a séparé nos données en corpus d'apprentissage (X_train et Y_train) et en corpus de test (X_test et Y_test). Nous avons pris 30% des données globales comme volume pour les données de test. Il faut toujours appliquer l'entraînement et les tests sur des données différentes, c'est le seul moyen de vraiment savoir si le modèle peut bien s'appliquer à des nouvelles données.

Dans un premier temps, nous avons pris les paramètres par défaut.
L'arbre de décision obtenu est le suivant, où nous avons fixé max_depth à 3 pour réduire sa profondeur.

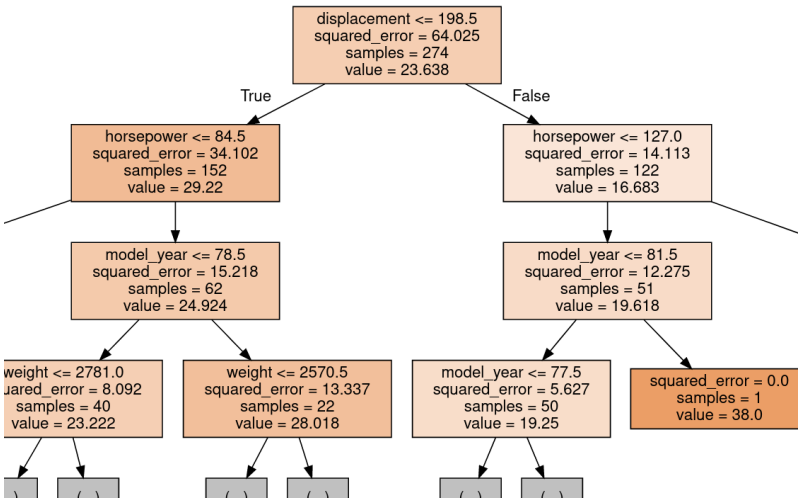



FIGURE 2.1 – Arbre de décion

L'importance des variables est donnée par la figure ci-dessous :
Il y'a un lien entre les noeuds de l'arbre et l'importance des variables. Plus la variable est importante, plus elle se trouve dans les noeuds de dessus. La variable displacement par exemple étant la plus importante, elle constitue ainsi la racine de l'arbre : c'est la variable qui a réalisé le test le plus discrimant. L'interprétation d'un arbre est très facile. On sait calculer le gain d'entropie d'un attribut, plus il est faible, plus l'attribut sera dans les noeuds de dessus.

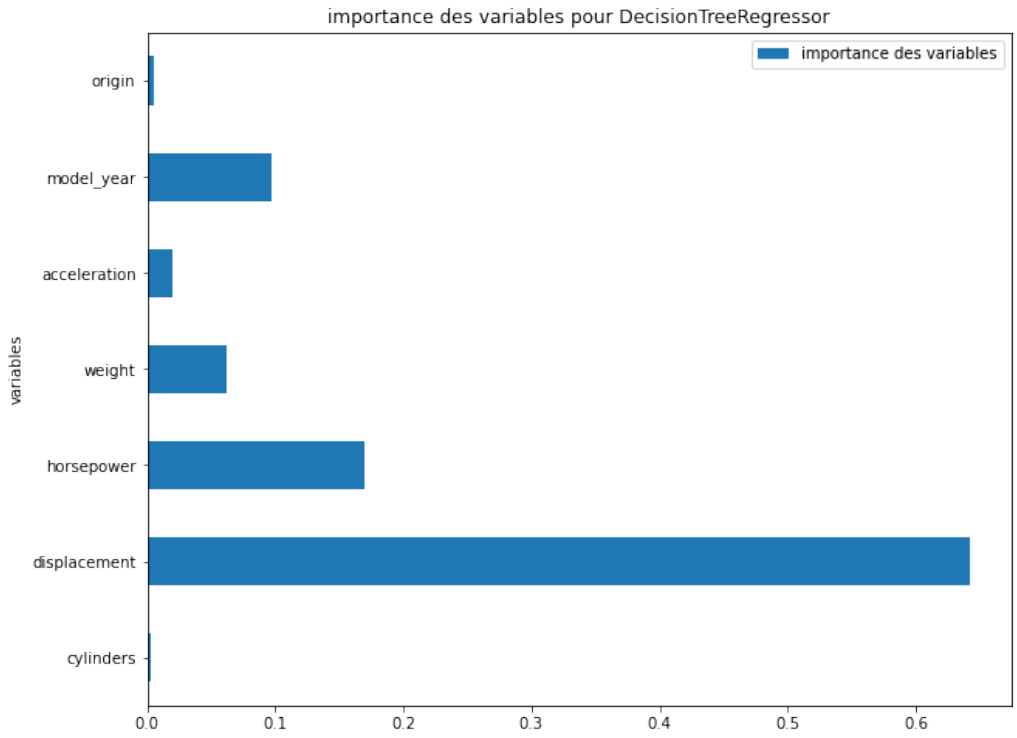



FIGURE 2.2 – Importance des variables

Les résultats des métriques sont les suivantes :

| | mean squared error | mean absolute error | R2 |
|---|---|---|---|
| DecisionTreeRegressor | 12.160254 | 2.411017 | 0.770149 |

TABLE 2.1 – résultats des métriques

Pour améliorer ces résultats, nous avons réalisé par la méthode Grid Search la recherche des meilleurs hyperparamètres, et nous avons trouvé ceci :
Avec donc, max_depth calibré à 5, R2 sur les données d'entrainement par la méthode validation croisée avec 10 folds vaut 0.812.
Ainsi, avec cette modification des hyperparamètres, nous obtenons ceci sur les données de test

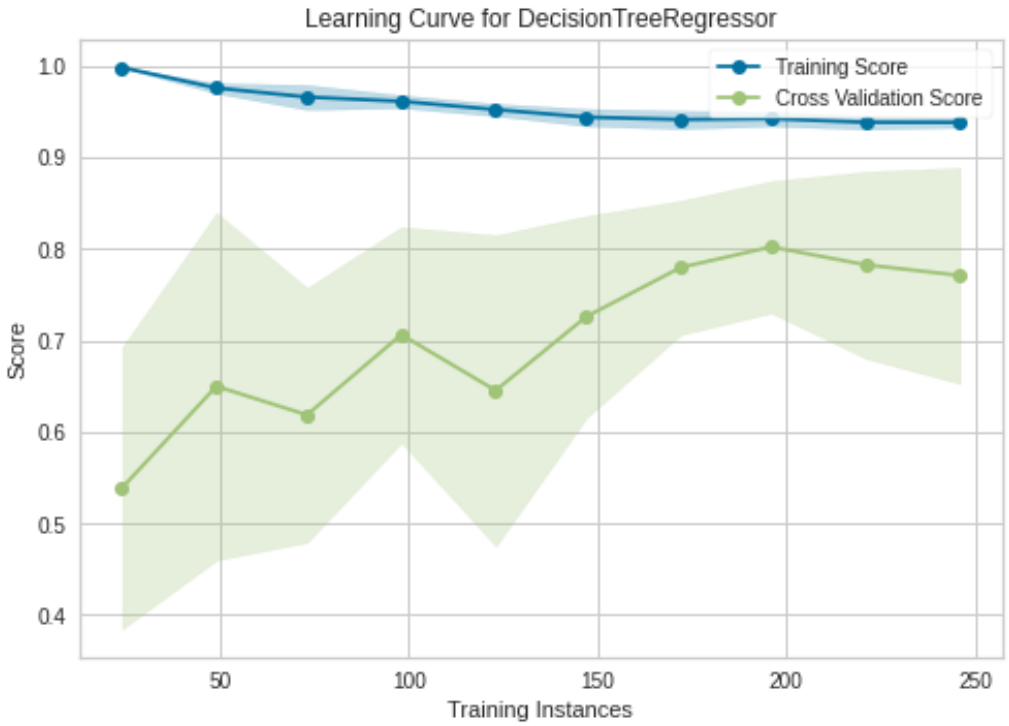
| | mean squared error | mean absolute error | R2 |
|---|---|---|---|
| DecisionTreeRegressor | 10.094 | 2.304 | 0.809 |

TABLE 2.2 – résultats des métriques

```
Meilleur(s) hyperparamètre(s) sur le jeu d'entraînement:
{'max_depth': 5}
Résultats de la validation croisée :
r2 = 0.700 (+/-0.174) for {'max_depth': 2}
r2 = 0.746 (+/-0.127) for {'max_depth': 3}
r2 = 0.803 (+/-0.104) for {'max_depth': 4}
r2 = 0.812 (+/-0.062) for {'max_depth': 5}
r2 = 0.796 (+/-0.116) for {'max_depth': 6}
r2 = 0.789 (+/-0.074) for {'max_depth': 7}
r2 = 0.760 (+/-0.127) for {'max_depth': 8}
r2 = 0.778 (+/-0.084) for {'max_depth': 9}
r2 = 0.772 (+/-0.124) for {'max_depth': 10}
```

FIGURE 2.3 – R2 pour différents max_depth

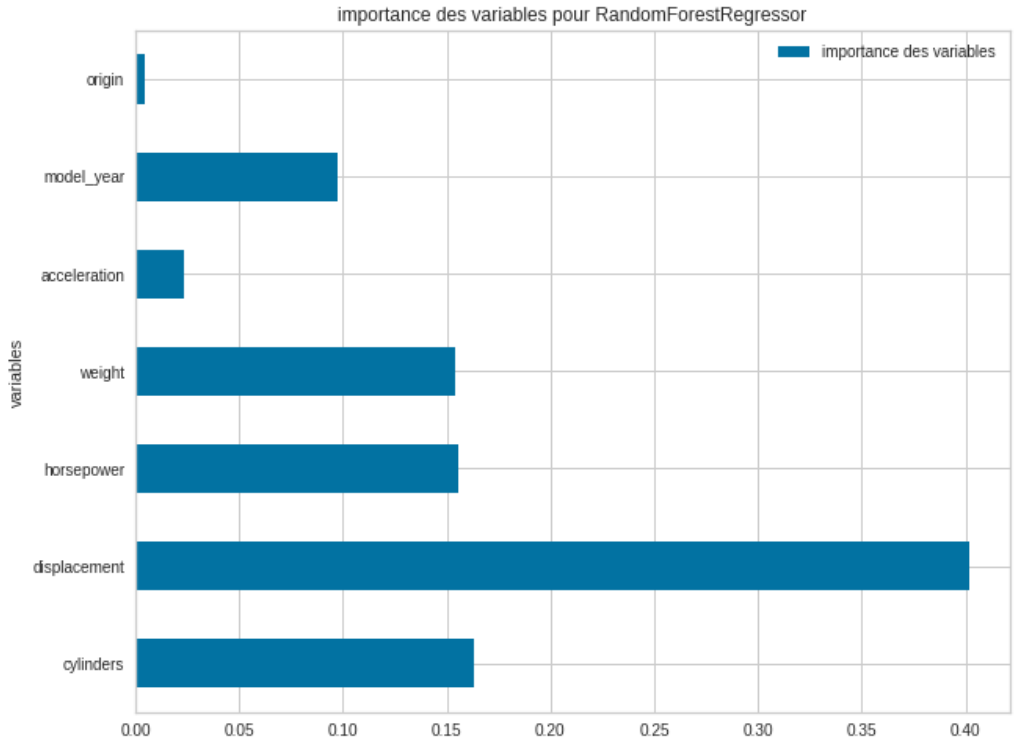
La figure ci-dessous montre que le modèle n'est pas trop biaisé (ombre bleue petite), mais par contre souffre d'un problème de variance (ombre verte grande), il est donc très sensible aux variations de données. On savait déjà ce problème pour les arbres de décisions. Ce n'est pas une surprise. Aussi, la figure montre que le modèle a besoin de plus de données pour que la courbe des scores (R2) d'apprentissage et la courbe des scores de validation puissent converger.



Chapitre 3

# Forêts aléatoires de régression

L'objectif de cette section est de présenter les résultats pour les forêts aléatoires.




Notre méthode est similaire à la section précédente, c'est-à-dire méthode Grid Search pour rechercher les hyperparamètres optimaux, avec validation croisée 10 folds sur données d'entrainements, puis évaluation du modèle calibré avec les meilleurs hyperparamètres sur les données de test. Les résultats fnaux sont les suivants avec meilleurs hyperparamètres ci-après :
n_estimators calibré à 250 et max_depth calibré à 10

| | mean squared error | mean absolute error | R2 |
|---|---|---|---|
| RandomForestRegressor | 6.951 | 1.832 | 0.869 |

TABLE 3.1 – résultats des métriques

Chapitre 4

# Clustering

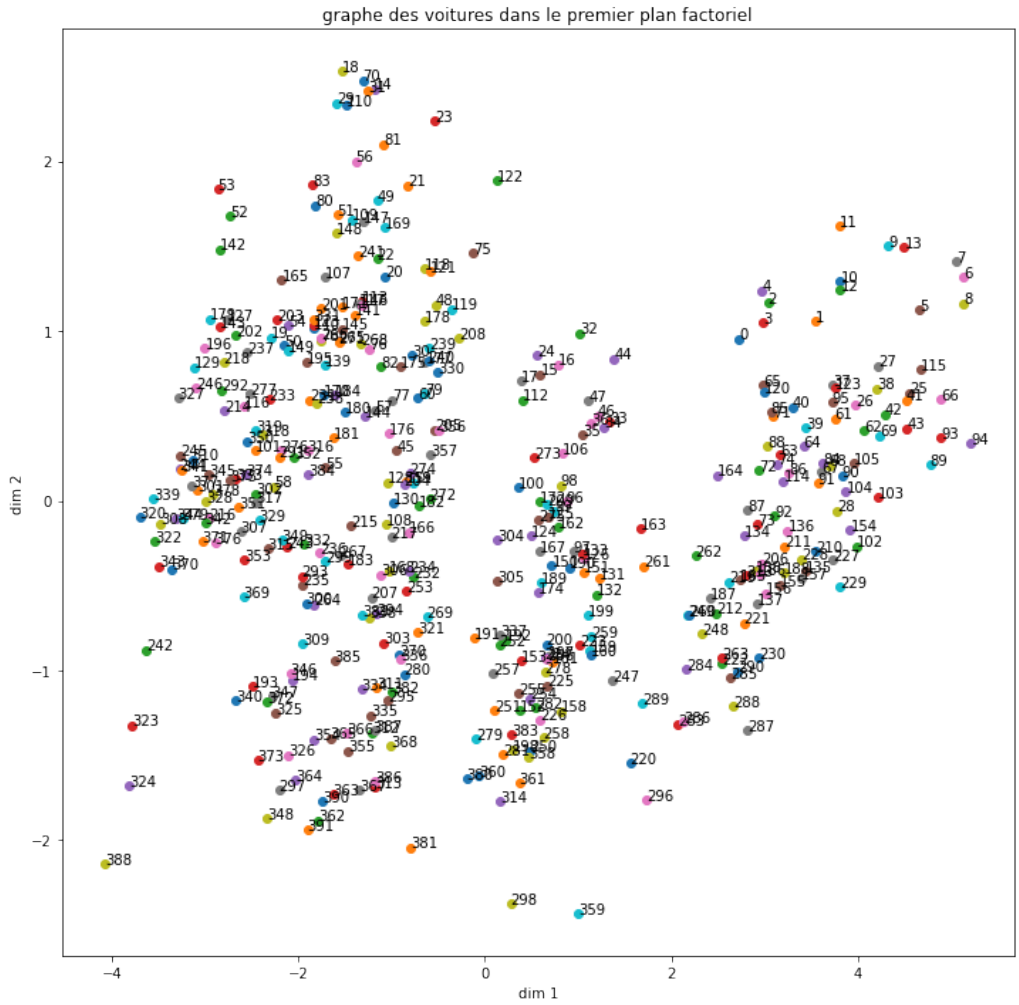



FIGURE 4.1 – projection des voitures
dans le premier plan facoriel

Les résultas des métriques d'évaluation sont :

| | silhouette_score | calinski_harabasz_score | davies_bouldin$_score$ |
|---|---|---|---|
| Kmeans | 0.380776 | 451.488594 | 0.989109 |

TABLE 4.1 – résultats des métriques

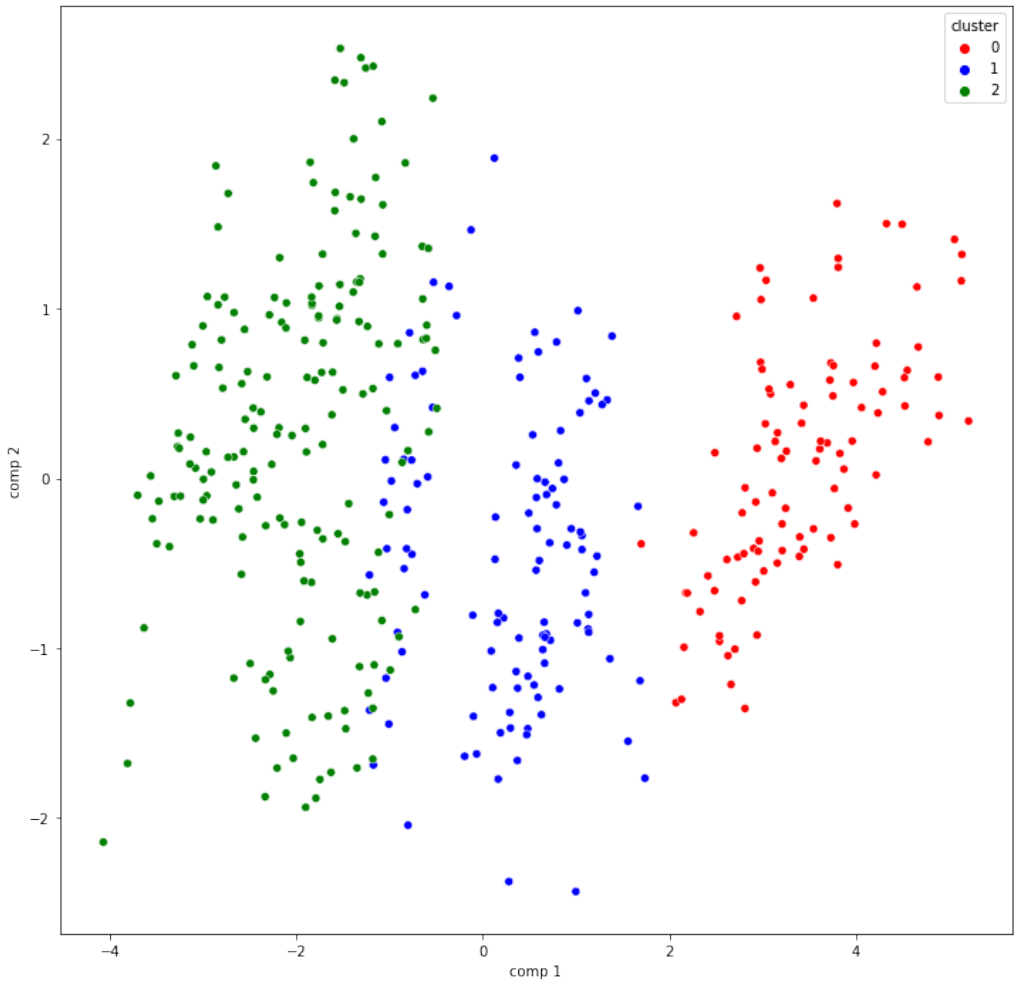



FIGURE 4.2 – clusters obtenus avec Kmeans

# Bibliographie

[Bre01] Leo Breiman. Random forests. *Machine learning*, 45(1) :5–32, 2001.