BERTRAND, Simon

Promotion 2023

Année 2020-2021

Diplôme d'ingénieur Télécom Physique Strasbourg

Mémoire de stage de 1re année

Intégration d'outils d'évaluation de clustering dans la plateforme FoDoMuST

ICube UMR 7357 - Laboratoire des sciences de l'ingénieur, de l'informatique et de l'imagerie
300 bd Sébastien Brant
CS 10413 - F-67412 Illkirch Cedex

Tél : +33 (0)3 68 85 45 54

GAN ÇARSKI, Pierre
gancarski@unistra.fr
0368854576

01/07/2021 au 31/08/2021

Intégration d'outils d'évaluation de clustering dans la plateforme FoDoMuST

La fouille de données est un domaine des sciences des données où l'on cherche à obtenir des connaissances supplémentaires sur des phénomènes observés à l'aide de données provenant de tout horizon : images, séries temporelles, tableaux de données, etc. L'importante quantité des données nécessaires à traiter rend la fouille manuelle impossible pour un expert. Il est alors inévitable de passer par des algorithmes pour retrouver des tendances, des valeurs aberrantes ou des phénomènes non observés directement. Parmi les méthodes existantes : le clustering. Le clustering est une technique algorithmique qui permet de regrouper des observations selon leur similarité. Ainsi, à chaque observation d'un jeu de données, on peut attribuer un numéro de cluster qui traduit l'appartenance à ce dernier. La mission qui m'a été confiée durant ce stage concerne le développement d'un logiciel Python qui sera implémenté dans la plateforme FoDoMuST développée par l'équipe Science des Données et Connaissances (SDC) du laboratoire ICube. L'objectif de la bibliothèque développée pour le stage est de fournir des outils pour évaluer la qualité des clusters générés par la plateforme principale FoDoMuST dans le cadre de l'apprentissage non supervisé. Pour aboutir à une analyse cohérente de ces clusters, nous utiliserons des indices de validation interne trouvés par des chercheurs mathématiciens ainsi qu'une multitude d'opérations algébriques afin de permettre à l'utilisateur d'analyser au mieux de lui-même la qualité de ses clusters. Dès lors que la bibliothèque est créée, il faudra rendre cette dernière accessible au public en y ajoutant une documentation complète ainsi que des options d'exécution depuis le terminal.

Integration of clustering evaluation tools in the FoDoMuST platform

Data mining is about increasing knowledge by using data of different types: image, time-series or even simple datasets. To consider manually analysing this large amount of data seems impossible for a simple human. That's why researchers and engineers use algorithms to achieve their goals. In the unsupervised learning field, there are many techniques to increase the knowledge of current data. We can look for the outliers, the general trend or even facts that are not visible at first sight. The technique in this field that concerns me is clustering. Clustering is about making subgroups of data based on their similarity. For each observation, we can assign it a cluster label which means that this observation belongs to this cluster. For my internship, I had to create a tool that analyses the quality of the clusters generated by the main algorithm FoDoMuST. In order to achieve my ends, I had to use several types of algebraic operations as are the internal validation indices and I had to create functions that estimate cluster density, cluster confusion, cluster shape and cluster similarity. Since the library is made to be used by everyone, it is fully in English. A complete documentation is also provided. Each user should be able to understand how the library works and use it for his own purposes. A command line interface - CLI – is provided to allow users to use the library directly from the terminal

Table des matières

Préambule

La loi de Moore a été annoncée en 1965. Cette dernière stipulait un doublement des capacités des semi-conducteurs tous les deux ans, le tout à prix constant. Mais depuis la dernière décennie, ce gain de puissance dans les calculs électroniques et informatiques ouvre un immense champ de possibilités aux scientifiques. À mi-chemin entre les statistiques et l'informatique, la science des données se nourrit de l'immense quantité de données que nous produisons au quotidien. Le scientifique des données procède à tout type de tâches : du nettoyage, au pré-traitement en passant par l'analyse ou encore la visualisation. Certains sous-domaines vont même jusqu'à élaborer des modèles par inférence statistique, ce qu'on appelle communément l'apprentissage automatique ou le machine learning.

On distingue trois principaux types d'apprentissage en science des données : l'apprentissage supervisé, l'apprentissage non supervisé et l'apprentissage par renforcement. Le dernier n'est pas abordé durant ce stage. On dit que l'apprentissage est supervisé lorsque nous fournissons la réponse aux questions que nous posons à notre algorithme. Celui-ci peut alors utiliser les solutions aux questions pour les comparer avec les réponses qu'il a émises et ainsi modifier son mécanisme de décision. L'objectif de l'algorithme est de maximiser sa note en modifiant quelques-unes de ses réponses et ce, jusqu'à obtenir la meilleure notation possible. Par exemple, si l'algorithme est entraîné avec un million d'images de chats et de chiens, le fait de lui indiquer à l'algorithme les correspondances entre l'image et le nom du mammifère visible correspond à une phase supervisée. L'intérêt par la suite est de fournir une nouvelle image d'un des deux mammifères et l'algorithme reconnaitra dans l'idéal celui dont il s'agit.

Contrairement à la phase supervisée, l'apprentissage non supervisé ne nécessite pas à priori de connaissances. L'algorithme cherche lui-même des tendances, des cycles, des valeurs aberrantes ou encore des groupes de similarités - nommés clusters - sans même connaître la signification physique des différentes données. Ainsi, sur nos précédentes images, l'algorithme peut regrouper ces images en deux clusters. Dans l'idéal, il devra séparer les chats des chiens mais il se peut qu'il distingue des groupes de données que l'homme ne verrait pas aux premiers abords. Ce phénomène est une augmentation des connaissances par la donnée. Il est alors important de connaître la structure de ces groupes pour interpréter leur sémantique. L'évaluation de la qualité des clusters générés est un travail mathématique et algorithmique minutieux qui se réalise sur des espaces de dimensions élevées impossibles à visualiser.

1. Introduction

Afin de pouvoir détailler au mieux le travail réalisé durant le stage, ce mémoire est construit de sorte qu'il amène progressivement le lecteur vers le centre de la mission du stage. Dans ce cadre, le mémoire débutera avec une présentation du contexte en décrivant le laboratoire ***ICube*** et le logiciel ***FoDoMuST*** ainsi que l'objectif de la mission. Nous ferons par la suite la démonstration d'un exemple de classification non supervisée en deux dimensions pour permettre une visualisation des objets manipulés et des clusters accessibles même aux moins initiés de la science des données. Nous introduirons ensuite la bibliothèque développée et finalisée par mes soins en prenant le temps de détailler chacune des fonctionnalités, des dépendances et des utilitaires de la librairie. Ces explications ont pour but d'éclairer les différentes sorties de la bibliothèque et proposent en particulier des figures graphiques pour visualiser les données calculées. Enfin, il sera établi un bilan personnel ainsi qu'une conclusion concernant le stage d'exécution.

Certaines parties du mémoire présentent des aspects techniques issus de la modélisation mathématique et séparés du texte principal afin de permettre une lecture fluide pour ceux qui ne souhaitent pas lire les différentes équations. Ces équations sont la traduction des fonctionnalités que j'ai implémentées et du travail de réflexion réalisé. En effet, les équations semblent être la manière la plus précise de fournir une description des travaux d'évaluation des clusters effectués durant ce stage. Pour autant, je conçois que ces dernières puissent être ignorées. Cet effort de modélisation mathématique est un point que j'apprécie particulièrement, c'est pourquoi je souhaite le partager aux différentes personnes qui liront ce mémoire. Pour les plus experts, je souhaite mettre en avant le fait que je n'ai pas suivi d'enseignements officiels de clustering durant mon année scolaire. Je me suis, par conséquent, auto-formé durant les derniers mois pour me permettre d'établir un travail de qualité. Mes connaissances associées à ce domaine étant récentes, il est probable que je manque de recul sur certains points.

2. Présentation du laboratoire ICube

ICube est un laboratoire de recherche public des sciences de l'ingénierie principalement situé au Parc d'Innovation de Strasbourg sur le second campus de l'Université de Strasbourg dans la ville d'Illkirch-Graffenstaden.

Le laboratoire est implanté au cœur du pôle API du Parc d'Innovation de Strasbourg aux côtés de l'école d'ingénieur Télécom Physique Strasbourg et de l'école des biotechnologistes ESBS (École Supérieur de Biotechnologie de Strasbourg). Nous pouvons retrouver un peu plus loin sur ce second campus la Faculté de Pharmacie, l'UFR de Mathématique et d'Informatique et l'IUT Robert Schuman.

De plus, ***ICube*** est présent dans six autres sites autour de Strasbourg. On compte la Faculté de Médecine, les Hôpitaux Universitaires ou encore Cronenbourg Est, à seulement deux kilomètres de l'accélérateur de particules de Strasbourg.

ICube est un laboratoire récent qui a vu le jour en 2013 dans le cadre d'une collaboration entre le CNRS, l'INSA de Strasbourg, l'ENGEES et l'Université de Strasbourg et vise à regrouper deux mondes scientifiques : ceux provenant de la physique et ceux provenant du numérique. Il s'intéresse à de nombreux domaines d'activités : imagerie, photonique, informatique (intelligence artificielle, réseaux, modélisation, calculs parallèles, …), robotique, technologie de la santé, télédétection, électronique et systèmes, répartis au sein de plusieurs départements. Chaque département est composé de plusieurs équipes opérant sur des sous-domaines différents. Au total, ***ICube*** décompte quatre départements :

- Département Informatique Recherche **(D-IR)** – Pierre Gançarski
- Département Imagerie, Robotique, Télédétection & Santé **(D-IRTS)** – Fabrice Heitz
- Département Electronique du Solide, Systèmes & Photonique **(D-ESSP)** – Paul Montgomery
- Département Mécanique **(D-M)** – Yannick Hoarau

Ces départements comptabilisent au total dix-sept sous-équipes. Pour le département de recherche informatique **D-IR** dirigé par Pierre Gançarski, les équipes installées sont :

- Informatique Géométrique et Graphique **(IGG)** – Dominique Bechmann
- Réseaux **(R)** – Thomas Noël
- Informatique et Calcul Parallèle Scientifique **(ICPS)** – Jens Gustedt
- Science des Données et Connaissances **(SDC)** – Nicolas Lachiche

- Systèmes Complexes, Bioinformatique Translationnelle **(CSTB)** – Olivier Poch et Pierre Collet
- Machine Learning, Modélisation et Simulation **(MLMS)** – Hyewon Seo et Stéphane Cotin
- Images, Modélisation, Apprentissage, Géométrie et Statistique **(IMAGeS)** – Fabrice Heitz

L'équipe dans laquelle j'effectue mon stage est celle de la Sciences des Données et Connaissances (***SDC***). J'ai pu collaborer étroitement avec Harrison Vernier et Chenglin XU. Harrison Vernier est un étudiant apprenti en Master 2 responsable du développement de l'application et Chenglin XU, un futur alternant Chinois qui reprendra le rôle d'Harrison Vernier lorsque celui-ci aura terminé son cycle. Le Pr. Pierre Gançarski reste le superviseur de cette équipe qui gère le projet ***FoDoMuST***.

Le laboratoire possède une multitude de plateformes de recherche. On dénote la nouvelle plateforme de recherche, ayant vu le jour en 2021, ***de Géométrie, d'Analyse et d'Intelligence Artificielle*** **(*GAIA*)** à fort engagement en sciences des données.

Ces départements sont accompagnés par des pôles administratifs, de sécurité ainsi que d'informatique. L'ensemble de la structure est dirigé par le directeur Michel de Mathelin, accompagné de ses collaborateurs du secrétariat, de l'administratif et du financier et de la coordination de projets.

Au total, ***ICube*** comptabilise près de 650 membres et plus de 5 846 publications scientifiques depuis 2013. La structure possède plus de 83 partenaires industriels nationaux et 30 partenaires industriels internationaux. On peut notamment y apercevoir des sociétés françaises comme ***Renault, AREVA, ARKEMA, SNCF, Suez Environnement, EDF***, mais aussi des groupes internationaux tels que ***Novartis, Daimler, Volkswagen, General Electric***, etc. ***ICube***, acteur des activités en liaison avec Télécom Physique Strasbourg, est membre de l'Institut Carnot Télécom & Société numérique et participe au programme « Futur & Ruptures » de l'Institut Mines-Télécom. On compte par exemple parmi ses membres actifs dans la recherche, le Pr. Pierre Collet, directeur et professeur de l'école Télécom Physique Strasbourg (TPS), Pr. Fabrice Heitz professeur en traitement du signal (TPS), Pr. Bernard Bayle professeur d'automatique (TPS), Pr. Jihad Zallat professeur de rayonnement et image (TPS) et Pr. Sylvain Lecler professeur de propagation des ondes (INSA&TPS). Certains chercheurs enseignent à l'Université de Strasbourg comme le Pr. Pierre Gançarski, professeur en informatique du master Sciences des données et systèmes complexes (SDSC) de l'UFR Math.-Info. de Strasbourg.

ICube est au centre du savoir théorique et pratique, la structure recense les meilleurs scientifiques en informatique, en ingénierie et en analyse d'images de la région du Grand-Est et participe activement à l'éducation et à la transmission du savoir des futurs ingénieurs et chercheurs.

3. Présentation de la plateforme FoDoMuST

FoDoMuST est un environnement recueillant une collection d'algorithmes des sciences des données de haute technologie. Il permet d'exécuter des algorithmes de classification non supervisée sur des jeux de données complexes, de toutes natures et sans nécessité de compétences techniques. Doté d'une interface graphique et de nombreux boutons d'actions, il met en fonction les principaux algorithmes de classification tels que ***K-Means, Cobweb*** ou encore des algorithmes de classification hiérarchique. ***FoDoMuST*** intègre la méthode innovante de ***SAMARAH***, une classification collaborative multistratégie sous contrainte incrémentale qui s'avère particulièrement efficace notamment avec les séries temporelles d'images.

L'application principale de ***FoDoMuST*** concerne la télédétection. Sur des jeux de données composés d'une série d'images fixes prises à intervalle temporel quelconque, les outils mis à dispositions par le logiciel permettent d'établir une classification sur l'ensemble de la série d'images. On peut ainsi observer des groupes de plusieurs façons afin d'en tirer des conclusions. Dans certains cas qui dépendent du type de donnée sur lequel le travail est effectué, les conclusions ne peuvent être émises uniquement par l'utilisateur mais nécessitent une interprétation externe d'un spécialiste du domaine. Cependant, ***FoDoMuST*** n'est pas réservé qu'à la télédétection, le logiciel se généralise à tout type de données.

Le Pr. Pierre Gançarski, en charge de l'exécutif, réalise les présentations de la plateforme lors des différentes conférences. ***FoDoMuST*** a remporté le prix pour la catégorie « démonstration » de l'EGC (Extraction et Gestion des Connaissances) 2020 à Bruxelles avec la publication « FODOMUST - Une plateforme de clustering collaboratif sous contraintes incrémentales de séries temporelles ».

FoDoMuST fonctionne sous le langage de programmation Java et est développé en collaboration sous le programme ***Git*** **(*Forge*),** hébergé sur les serveurs d'***ICube***. Il intègre la bibliothèque ***Java Clustering Library*** **(*JCL*).** L'architecture du logiciel a été renouvelée cette année en suivant le modèle-vue-contrôleur (MVC) à l'aide des services en génie logiciel d'Harrison Vernier. Ce dernier a aussi finalisé le développement d'une version client-serveur du logiciel afin d'alléger le client et de distribuer les calculs sur le supercalculateur du campus. Durant tout le stage, il a été mon référent afin que je développe l'application d'évaluation des clusters en compatibilité avec les entrées et les sorties de données réalisées par ***FoDoMuST***.

4. La fouille de données : une augmentation du savoir

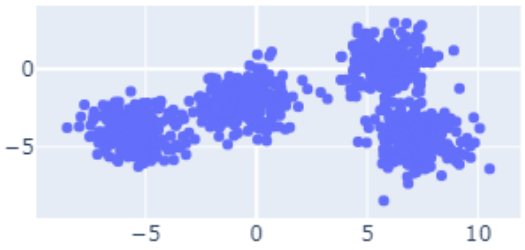
4.1. Exemple d'une classification non supervisée

Nous allons présenter une classification non supervisée dans sa version la plus simple possible, à savoir un jeu de données en deux dimensions. Cela devrait permettre une meilleure compréhension de l'objectif de la mission et du poste occupé. On génère ainsi un tableau aléatoire de deux dimensions composé de mille éléments. Géométriquement, on interprète cette table comme une combinaison de mille points sur un plan à coordonnées x pour la première colonne et y pour la seconde.

	0	1
0	7.898725	-5.169588
1	-7.117671	-4.084779
2	5.766493	-1.110315
3	6.731771	-3.299167
4	-0.776267	-1.776232
...	...	...
995	1.297537	-2.134081
996	6.029418	2.003129
997	8.332663	-4.167202
998	0.519522	-1.934230
999	-2.861718	-2.776254

1000 rows × 2 columns

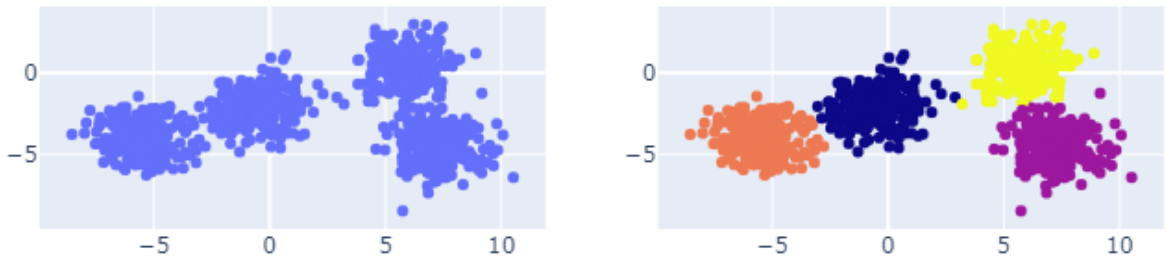
Tableau de données Pandas



Visualisation des données avec Plotly

(1)

On peut d'ores et déjà identifier quatre groupes distincts. Nous pouvons exécuter l'algorithme du ***K-Means*** sur le tableau précédent avec comme paramètre de l'algorithme, le nombre de noyaux qui correspond au chiffre quatre.



(à gauche) Visualisation des données avec Plotly

(à droite) Visualisations des quatre clusters générés par K-Means avec Plotly

(2)

Le travail de classification non supervisée est ainsi réalisé. Chacun des points est associé à un cluster.

4.2.Objectif de la mission & poste occupé

L'exemple précédent est un cas idéal et peu fréquent. En effet, on peut identifier à l'avance le nombre de clusters, la dimension n'est que de deux ce qui permet une visualisation complète des données, les clusters ont tous la même forme, ils sont séparés et la donnée n'est pas temporelle. On peut observer les exemples de la réponse de ***K-Means*** pour différents jeux de données de deux dimensions et de formes particulières.

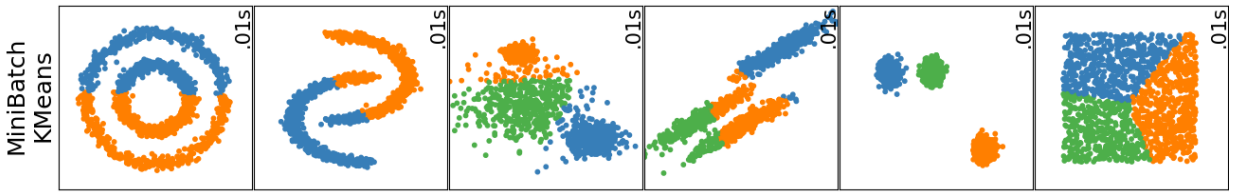


Image extraite de la documentation de la bibliothèque Python scikit-learn ***(3)***

Cependant dans le cadre de la classification réalisée par ***FoDoMuST***, les données peuvent être temporelles ou géométriques et de dimension quelconque.

L'objectif de la mission est de permettre aux utilisateurs de ***FoDoMuST*** d'interpréter les clusters générés en proposant une multitude d'indicateurs mathématiques justifiant de la densité des clusters, de leur taille, leur séparation, leur homogénéité, leur compacité ou encore de leur convexité.

Pour réaliser cet objectif, j'ai choisi de créer une bibliothèque indépendante de ***FoDoMuST*** qui prend en entrée les sorties de données de la plateforme principale. Cette dernière sera développée en Python, un langage de programmation que j'ai appris en classe préparatoire mais aussi parce que les utilisateurs de ***FoDoMuST*** souhaitent étendre la plateforme au langage Python. Cela permettra d'utiliser à l'avenir des scripts faciles d'utilisation pour obtenir rapidement les caractéristiques des clusters générés.

J'ai occupé le poste de Développeur Python – Back end - avec une orientation en sciences des données où j'ai utilisé les librairies standards tels que ***Numpy***, ***Pandas***, ***Scipy***, ***Plotly***, ***Statsmodels*** et ***Scikit-learn***. Au niveau de la programmation, j'ai principalement fait usage de la programmation orientée objet et du calcul vectoriel. Cependant pour implémenter les différents indices que fournit mon application, j'ai fait appel à des notions mathématiques d'algèbre euclidienne ainsi que de modélisation qui m'ont permis la lecture des différentes références [1-3] mais aussi l'écriture de la modélisation mathématique de ce document. Des notions de systèmes informatiques ont aussi été nécessaires pour la création de la bibliothèque et son déploiement sur l'ensemble des systèmes d'exploitation.

5. La bibliothèque Clusters-Features

Pour répondre à mes objectifs, j'ai créé entièrement la bibliothèque Python nommée ***Clusters-Features***. De son identité graphique au moteur, en passant par la documentation, cette bibliothèque fournit une multitude d'indicateurs et de données qui permettent l'évaluation des clusters générés. C'est une extension indépendante du logiciel ***FoDoMuST*** et qui peut être utilisée en dehors de la plateforme. Les entrées et sorties de ***Clusters-Features*** sont hautement compatibles avec celles de ***FoDoMuST***.

5.1. Dépendances

La bibliothèque ***Clusters-Features*** est dépendante d'autres bibliothèques Python. La version de base possède quatre dépendances : ***Numpy***, ***Pandas***, ***Scipy*** et ***Scikit-learn***. La version avec partie graphique ajoute une dépendance de la librairie ***Plotly*** et la version avec utilitaires externes ajoute une dépendance aux librairies ***Statsmodels***, ***Umap-learn*** et ***Numba***.

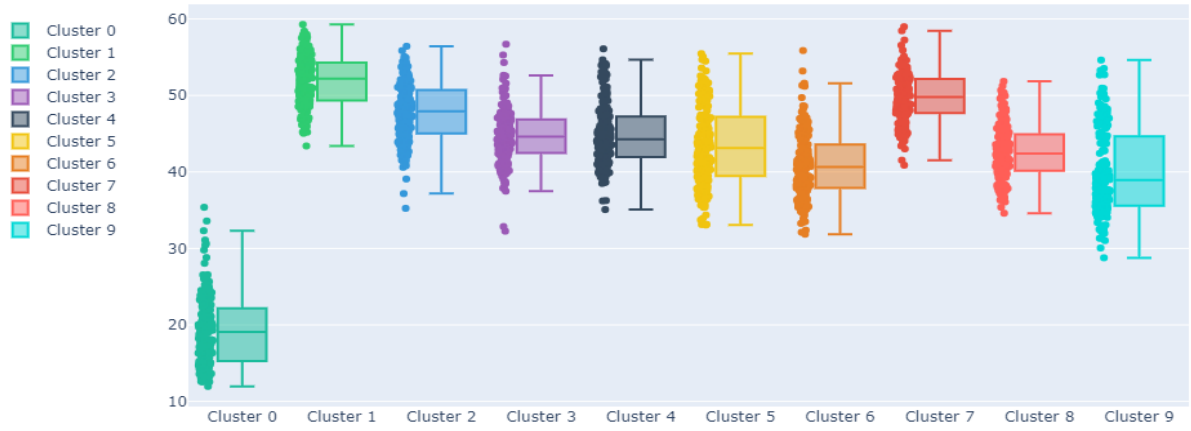
Ces librairies ont été choisies pour leur popularité, leur efficacité, leurs mises à jour régulières et elles sont particulièrement reconnues dans le monde de la science des données sur Python.

5.2. Fonctionnalités

5.2.1. Caractéristiques des clusters

Comme nous l'avons vu précédemment à l'aide de l'exemple (2), la géométrie nous permet d'identifier des informations supplémentaires. Il est possible de trouver les centres de chacun des clusters et de calculer les distances les séparant. Une fois que ces centres sont considérés, il est possible de calculer toutes les distances séparant les points du cluster et le centre du même cluster. Sur ces distributions, nous pouvons faire des moyennes, des sommes, des produits, etc. Il existe un grand nombre de caractéristiques avec autant de possibilités que de formules mathématiques. Pour interpréter ces informations, prenons l'exemple du critère de séparation. En considérant deux clusters, le critère de séparation peut être établi en comparant les distances séparant les deux centres avec la moyenne des distances moyennes séparant les points d'un cluster et de leur centre. Ainsi, si la distance séparant les deux centres est bien plus grande que la moyenne des distances moyennes de chacun des clusters, cela signifie que les deux clusters sont espacés et compacts. D'autres caractéristiques sont possibles, mais difficiles à lister. La bibliothèque ***Clusters-Features***

implémente une liste complète de caractéristiques variées et disponibles rapidement en sortie. En guise d'exemple, sur un jeu de données de 1797 images, de dimension 64 et avec 10 clusters où les coefficients du jeu de données sont compris entre 0 et 255, nous pouvons tracer un graphique en boîte de moustache (4) composé de l'ensemble des distances séparant tous les éléments d'un centre de cluster donné afin d'observer la distribution unidimensionnelle des rayons.



Librairie Plotly - Distances séparant tous les éléments de chaque cluster du centre du cluster 0. Données sortantes de Clusters-Features. ***(4)***

Partie technique :

Pour faciliter la définition des actions réalisées durant ce stage, nous définirons mathématiquement les objets développés dans la bibliothèque ***Clusters-Features.*** Pour commencer, munissons-nous du jeu de données Δ tel que le coefficient $\delta_{i,j}$ représente la $i$ème ligne et la $j$ème colonne du jeu de données pour tout $(i,j) \in [1,n] \times [1,m]$:

$$\Delta = (\delta_{i,j})_{(i,j)\in[1,n]\times[1,m]} \in M_{n,m}(\mathbb{R}) \qquad (5.2.1.1)$$

Les indices n et m étant des entiers naturels représentant respectivement le nombre d'observations et la dimension du jeu de données Δ, on peut définir les colonnes C_j de cette matrice ainsi que les lignes O_i pour tout $(i,j) \in [1,n] \times [1,m]$:

$$\forall j \in [1,m],\ \mathbf{C}_j = \begin{pmatrix} \delta_{1,j} \\ \dots \\ \delta_{n,j} \end{pmatrix} \qquad \forall i \in [1,n],\ \mathbf{O}_i = \begin{pmatrix} \delta_{i,1} \\ \dots \\ \delta_{i,m} \end{pmatrix} \qquad (5.2.1.2)$$

On appelle T le vecteur composé des numéros des clusters générés pour chaque point du jeu de données. On pose K comme étant le nombre de clusters.

$$\mathbf{T} = \begin{pmatrix} t_1 \\ \dots \\ t_n \end{pmatrix} \in [1, K]^n \qquad (5.2.1.3)$$

On peut définir ensuite le jeu de données $\Delta^{\{k\}}$ associé à un numéro de cluster. Les numéros de cluster sont des entiers notés k et appartenant à $[1, K]$, on définit ensuite les colonnes $C_j^{\{k\}}$ associées au jeu de données du cluster k :

$$\begin{aligned} \forall k \in [1, K], \quad \Delta^{\{k\}} &= \left(\mathbf{O}_i^{\top} \mid t_i = k\right)_{i \in \left[1, n^{\{k\}}\right]} \in M_{n^{\{k\}}, m}(\mathbb{R}) \\ \Delta^{\{k\}} &= (\mathbf{C}_j^{\{k\}})_{j \in [1,m]} \end{aligned} \qquad (5.2.1.4)$$

La définition des centres des clusters $G^{\{k\}}$ est telle que :

$$\forall k \in [1, K], \quad \mathbf{G}^{\{k\}} = (\mathbb{E}(\mathbf{C}_j^{\{k\}}))_{j \in [1,m]} \in \mathbb{R}^m \qquad (5.2.1.5)$$

Chacun des clusters $\Delta^{\{k\}}$ possède $n^{\{k\}}$ points, soit $n^{\{k\}}$ lignes. Comme les $(\Delta^{\{k\}})_{k \in [1,K]}$ forment une partition de Δ, nous avons :

$$\sum_{k=1}^{K} n^{\{k\}} = n \qquad (5.2.1.6)$$

Une fois que ces objets mathématiques sont définis, nous pouvons définir les premières caractéristiques comme la matrice Mat_{pw_d} des distances paires à paires de tous les points :

$$Mat_{pw_d} = (\|\mathbf{O}_{i_1} - \mathbf{O}_{i_2}\|)_{(i_1, i_2) \in [1,n]^2} \qquad (5.2.1.7)$$

Ou encore la matrice Mat_{ic_d} des distances entre les centres :

$$Mat_{ic_d} = (\|\mathbf{G}^{\{k_1\}} - \mathbf{G}^{\{k_2\}}\|)_{(k_1, k_2) \in [1,K]^2} \qquad (5.2.1.8)$$

La matrice des distances séparant tous les éléments de chacun des centres est définie telle que :

$$Mat_{pw_{cd}} = (\|\mathbf{O}_i - \mathbf{G}^{\{k\}}\|)_{(i,k) \in [1,n] \times [1,K]} \qquad (5.2.1.9)$$

5.2.2. Scores & Indices de validation interne

Il existe de nombreux indices appelés indices de validation interne proposés par des chercheurs mathématiciens qui ont pour vocation de calculer un score pour chaque cluster afin d'évaluer certains critères comme la séparation, la compacité, la dispersion du cluster ou encore sa taille. La bibliothèque implémente plus de 40 indices différents dont plus de 27 figurants dans la référence [1]. Le travail de recensement et de modélisation mathématique réalisé dans la précédente référence a permis une intégration rapide de ces indices dans la bibliothèque ***Clusters-Features***. Certains de ces indices ont été modifiés pour afficher une version de l'indice pour chaque cluster. Parmi les indices implémentés provenant de la référence [1], on compte notamment l'indice de Dunn (et sa version généralisée), l'indice de Calinski-Harabasz, de Davies-Bouldin, de Banfeld-Raftery, l'indice C ou encore celui des chercheurs d'***ICube*** : l'indice de Wemmert-Gançarski. Une liste complète de tous les indices calculés par la bibliothèque est disponible dans la documentation associée.

5.2.3. Confusions des hypersphères centrées

La mesure de la confusion d'un cluster avec un autre peut se réaliser en comptant le nombre de points appartenant à un cluster qui sont contenus dans l'hypersphère de rayon r centrée sur le centre même du cluster. Géométriquement, en deux dimensions et en correspondance avec la figure (2), on peut tracer un cercle de rayon r centré sur le centre du cluster et compter le nombre de points pour chaque cluster qui sont inclus dans ce cercle. Dans l'idéal, le nombre de ces éléments pour des clusters différents doit rester très bas et le nombre de ces éléments pour des clusters identiques doit être le plus haut possible et le plus tôt possible (c'est-à-dire pour des petits rayons). On peut observer la réponse de cette sortie pour différents rayons : rayon moyen d'un cluster, médian ou même des rayons à percentiles proches de 100 afin d'exclure les valeurs aberrantes du cluster. Il est possible de conclure sur la séparation des clusters et de trouver les clusters les plus confondus. Cette analyse repose sur la convexité des hypersphères, elle est alors adaptée pour des clusters convexes, ce qui reste une hypothèse idéale.

Partie technique :

Soit $r \in \mathbb{R}^+$. Nous définissons l'hypersphère de rayon r centrée sur le cluster k ainsi :

$$\forall k \in [1, K]\,,\ HS_r^{\{k\}} = \left\{ \mathbf{x} \in \mathbb{R}^m \mid \left\| \mathbf{x} - \boldsymbol{G}^{\{k\}} \right\| < r \right\} \qquad (5.2.3.1)$$

En considérant toutes les paires de clusters possibles, on peut définir le nombre d'éléments $n^{\{k1\}}_{\in HS_r^{\{k2\}}}$ appartenant au cluster $k1$ qui sont inclus dans l'hypersphère de rayon r centrée sur le cluster $k2$:

$$\forall (k_1, k_2) \in [1, K]^2 ,$$
$$n^{\{k_1\}}_{\in HS_r^{\{k_2\}}} = Card(\left\{ d_i = \left\| \mathbf{O}_i^{\{k_1\}} - \mathbf{G}^{\{k_2\}} \right\| \in \mathbb{R}^+ |\ \forall i \in \left[1, n^{\{k_1\}}\right] ,\ d_i < r \right\})$$

Équation (5.2.3.2)

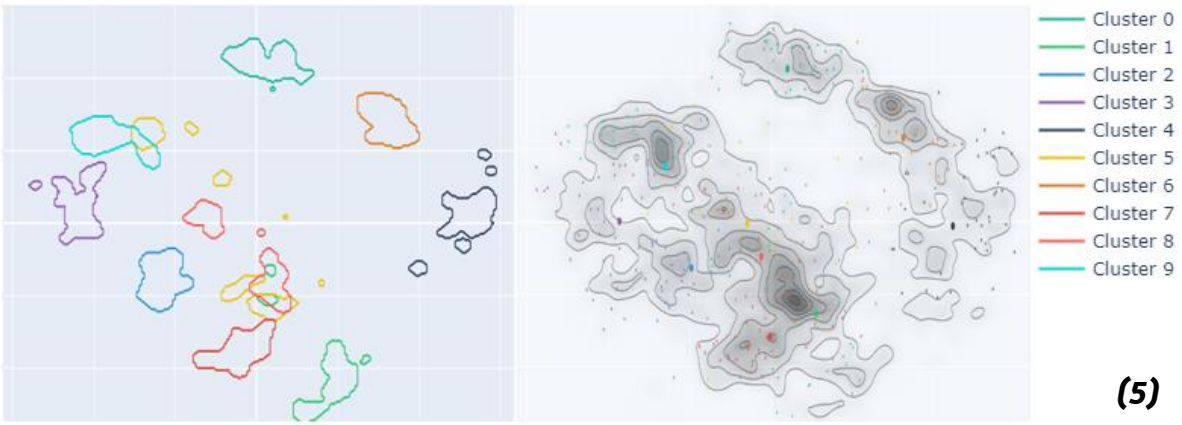
La matrice de confusion pour un rayon r renvoyé par la bibliothèque est définie telle que :

$$Mat_{(conf.HS_r)} = \left(n^{\{k_1\}}_{\in HS_r^{\{k_2\}}} \right)_{(k_2, k_1) \in [1,K]^2} \qquad (5.2.3.3)$$

Elle n'est évidemment pas symétrique. On peut mettre en œuvre une variante proportionnelle en divisant le précèdent terme général de la matrice par $n^{\{k1\}}$.

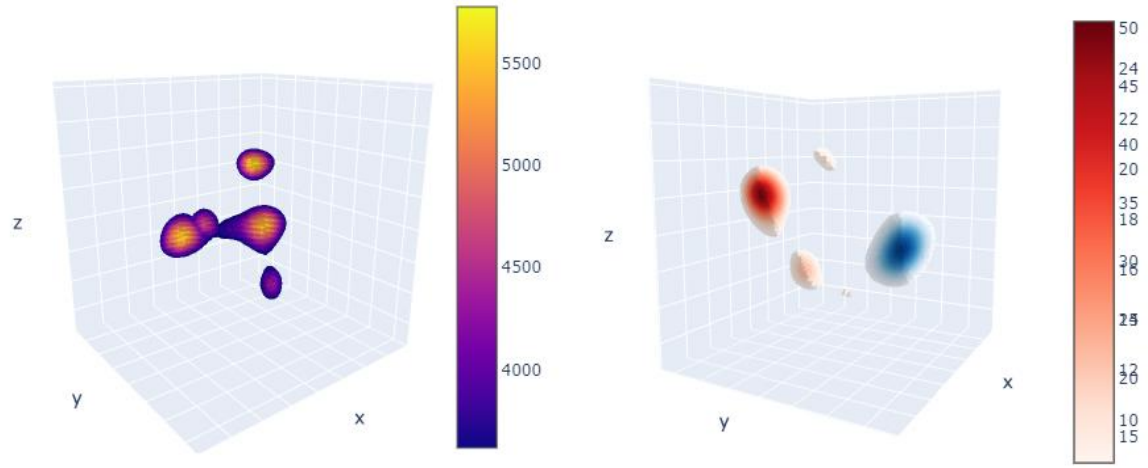
5.2.4. Estimation de la densité

Estimer la densité des clusters est l'origine de nombreuses thèses. La principale limite est celle annoncée par Richard Bellman en 1961 et qui porte le nom du fléau de la dimension. En effet, lorsque le jeu de donnée possède une faible dimension (inférieur ou égale à 3), il n'est pas difficile d'obtenir une densité pour chaque point et d'afficher une visualisation. Cependant, dans les espaces de très grande dimension où les différentes directions sont nombreuses, les distances peuvent atteindre des valeurs pharamineuses, ce qui fait perdre tout sens d'interprétation. Les solutions se résument dans un premier temps à réduire la dimension du jeu de données en gardant le maximum d'informations, ce qu'on appelle la réduction de dimensionnalité. Dans ce cas, nous pouvons afficher des densités sur des plans ou des espaces à trois dimensions mais qui reflètent une représentation résumée du jeu de données. Et dans un autre temps, nous pouvons essayer d'estimer une densité pour chaque point du jeu de données sans avoir à générer des grilles d'espace à très haute dimension qui demandent des puissances de calculs que même les supercalculateurs d'aujourd'hui ne peuvent posséder. La première solution par réduction de dimensionnalité avec ***Clusters-Features*** renvoie la donnée associée à la figure (5) et (6) pour un jeu de données d'images de dimension 64 contenant plus de 1797 éléments. Avec ***Clusters-Features***, on peut décider d'afficher la carte des densités totale, la carte des densités pour chaque cluster ou simplement les contours d'un minimum de la densité totale après avoir réduit la dimension du jeu de données, comme le montrent les figures suivantes :



(5)

Librairie Plotly – Estimation de la densité par grillage après réduction 2D PCA du jeu de données d'images. A gauche les contours, à droite la carte de densité totale. Données sortantes de Clusters-Features.



Librairie Plotly – Estimation de la densité par grillage après réduction 3D PCA du jeu de données d'images. A gauche tous les clusters, à droite deux différents clusters sont affichés.

(6)

Le principe utilisé pour estimer la densité repose sur le remplacement des points du jeu de données par des lois gaussiennes multivariées. Pour modéliser la densité en très haute dimension, il serait sans doute pertinent de faire varier les variances de la loi multivariée en fonction de la dimension du jeu de données pour obtenir une échelle des gaussiennes qui reste cohérente même à très haute dimension. Cependant, cette idée demande d'établir une expression d'une distribution fortement complexe et presque impossible à formaliser. Une fois l'ensemble des points remplacés par des gaussiennes multivariées, il suffit de se placer sur un point quelconque de l'espace et de sommer les résidus en ce point de toutes les précédentes gaussiennes. Ces dernières étant centrées sur chacun des points du jeu de données.

Partie technique :

Soit $x \in \mathbb{R}^m$, un point quelconque de l'espace du jeu de données. On définit la loi gaussienne multivariée centrée sur l'observation O_i à l'aide de la fonction en x définie telle que pour tout $i \in [1, n]$, nous avons :

$$f_{\mathbf{O}_i,\Sigma}(\mathbf{x}) = \frac{1}{(2\pi)^{m/2}\ det(\Sigma)^{1/2}} exp(-\frac{1}{2}(\mathbf{x}-\mathbf{O}_i)^\top \cdot \Sigma^{-1} \cdot\ (\mathbf{x}-\mathbf{O}_i))$$

Équation (5.2.4.1)

Cette fonction est utilisée lorsque le jeu de données est réduit en deux ou trois dimensions pour estimer la densité en chaque point de l'espace de petite dimension. Un grillage deux ou trois dimensions est généré par la bibliothèque.

En posant que la matrice de variance-covariance Σ de la loi gaussienne vaut l'identité pour simplifier les calculs, nous avons :

$$\Sigma = Id_m \in M_m(\mathbb{R}) \qquad \textit{(5.2.4.2)}$$

Ce qui permet de simplifier l'équation précédente en :

$$f_{\mathbf{O}_i,Id_m}(\mathbf{x}) = \frac{1}{(2\pi)^{m/2}} exp(-\frac{1}{2}\left\|\mathbf{x}-\mathbf{O}_i\right\|^2) \qquad \textit{(5.2.4.3)}$$

On estime la densité en sommant les résidus en x pour toutes les observations. On définit aussi une densité provenant d'un cluster en sommant sur l'ensemble des observations appartenant à ce cluster :

$$d_{total}(\mathbf{x}) = \sum_{i=1}^{n} f_{\mathbf{O}_i,Id_m}(\mathbf{x}) \qquad d_{\Delta^{\{k\}}}(\mathbf{x}) = \sum_{i=1}^{n^{\{k\}}} f_{(\mathbf{O}_i \ \mid\ t_i=k),Id_m}(\mathbf{x})$$

Équation (5.2.4.4) *Équation (5.2.4.5)*

Le principal problème de cette modélisation est lorsque la norme $\left||x - O_i|\right|$ prend des valeurs élevées. Cette dernière réalisant une somme sur la dimension du jeu de données de nombres strictement positifs, il est convenu que l'échelle de cette norme augmente lorsque la dimension du jeu de données augmente. L'idée serait d'appliquer un facteur correctif à cette norme qui dépend de la dimension. Il faudrait alors poser que $\Sigma = diag\left(\left(\frac{1}{\alpha(m)}\right)_{j\in[1,m]}\right) = \frac{1}{\alpha(m)} Id_m$ où $\alpha(m)$ est la fonction en m qui corrige la norme. Les variances des différentes gaussiennes seraient égales en tout axe et les covariances nulles. Cette fonction α nécessiterait d'être décroissante en m.

5.2.5. Utilitaires externes

La bibliothèque ***Clusters-Features*** implémente des fonctionnalités d'autres bibliothèques. C'est le cas pour différentes méthodes de réduction de dimensionalité provenant notamment de la librairie ***Scikit-learn*** pour l'algorithme du ***Principal Component Analysis*** (***PCA***) ainsi que l'algorithme ***Uniform Manifold Approximation & Projection*** (***UMAP***) de la bibliothèque ***Umap-learn***. On note aussi l'utilisation de la fonction ***Kernel Density Estimation*** (***KDE***) présente dans ***Scikit-learn*** qui permet d'obtenir une estimation différente de la densité des points du jeu de données. Ces utilitaires sont disponibles avec le préfixe « utils » dans la bibliothèque ***Clusters-Features*** et permettent d'obtenir des données informatives supplémentaires.

5.2.6. Pré-traitement du jeu de données

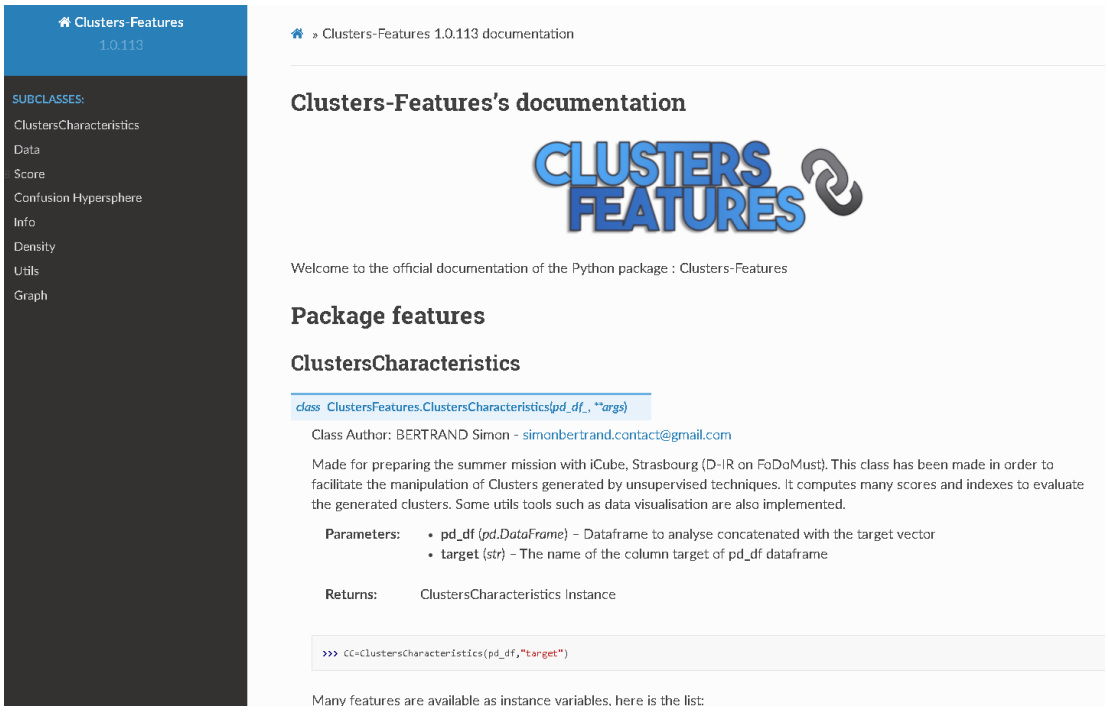
Il existe des jeux de données normalement non compatibles avec la bibliothèque ***Clusters-Features***. Parmi eux, on compte les jeux de données possédant des valeurs manquantes et ceux possédant des valeurs non numériques. Pour le premier cas, à l'aide de la librairie ***Scikit-learn***, on peut imputer les valeurs manquantes par d'autres valeurs prédites par différents modèles de régression. Ces derniers étant entrainés sur le jeu de données où les valeurs manquantes existent, cela permet de faire une prédiction pour les valeurs manquantes. Dans le deuxième cas, les colonnes et valeurs non numériques sont supprimés du jeu de données, ce qui restreint l'ensemble des jeux de données admis par la bibliothèque. Une normalisation et mise à l'échelle est réalisée sur ces jeux de données afin de prévenir les différentes divergences et explosions dans les calculs.

5.2.7. Visualisation des données

Enfin, la dernière fonctionnalité de la bibliothèque, qui est facultative au déploiement, permet de visualiser les données sortantes de l'algorithme. Indispensable pour vérifier les algorithmes internes mais dispensable pour le logiciel ***FoDoMuST*** qui possède sa propre librairie de visualisation de données (***JFree Chart*** combinée à ***Orson Chart***), cette dernière fonctionnalité s'appuie sur la librairie déclarative ***Plotly*** de Python qui offre des graphiques interactifs codés en Javascript sur lesquels l'utilisateur peut effectuer une série d'action lui permettant de visualiser de différentes façons les sorties de la bibliothèque.

5.3.Documentation

Le principal objectif de cette bibliothèque est qu'elle puisse être modifiée et utilisée simplement par les futurs collaborateurs de ***FoDoMuST***. C'est pourquoi une documentation sous le format de site internet statique est générée automatiquement à partir des commentaires effectués dans le code source de chacune des fonctions programmées de la bibliothèque. Le programme ***Sphinx*** est utilisé pour transcoder les commentaires en document HTML. La documentation recense toutes les informations nécessaires quant à l'utilisation de la librairie. De l'ensemble des fonctions disponibles, en passant par le détail des arguments en paramètres ainsi qu'une description des sorties et des actions réalisées par la bibliothèque, la documentation est la clé pour comprendre et maîtriser ***Clusters-Features*** dans sa totalité. On peut observer le début de la page d'accueil de la documentation sur la figure (7) :



Extrait de la documentation de Clusters-Features générée par Sphinx et le thème Read The Doc située à l'adresse simonbertrand.pages.unistra.fr/Clusters-Features. Le simple logo de Clusters-Features a été créé par mes soins. ***(7)***

5.4.Interface par ligne de commande

En suivant l'idée proposée par Harrison Vernier, j'ai développé une interface par ligne de commande qui permet de générer des sorties de données en format ***JSON*** issus des calculs proposés par la librairie. Cela permet une extension de la librairie par un simple script et peut généraliser son utilisation à tout type de langage de programmation ayant accès aux commandes du terminal.

5.5.Déploiement

Enfin, le déploiement se réalise avec deux outils différents. Le premier concerne la compilation de la librairie sous la forme d'un paquet Python installable par le programme ***pip*** à l'aide de la librairie ***setuptools***. Le second utilise le ***GitLab*** d'Unistra et notamment le service ***GitLab Pages*** et l'intégration continue pour déployer la documentation sur une page internet à chaque mise à jour du projet. Enfin, la version pour les programmeurs Python est téléchargeable directement sur le ***Gitlab*** d'Unistra où on peut l'utiliser sans l'installer.

5.6.Finalités et recommandations

La version déployée à la fin de mon stage est une version finalisée. Une partie d'optimisation a été nécessaire pour réduire au maximum les différents temps de calcul des indices disponibles dans ***Clusters-Features***. Le programme réalise une gestion des erreurs qui peut être perfectionnée à l'avenir pour couvrir la totalité des différentes erreurs provoquées par une mauvaise manipulation des arguments. Si cette bibliothèque devait être approfondie, je recommanderais l'intégration d'une fenêtre ***Jupyter Notebook*** dans ***FoDoMuST*** avec une instance de la classe principale de ***Clusters-Features*** créée dès le lancement pour permettre aux utilisateurs d'utiliser leurs propres scripts directement sur la plateforme. Ils auraient ainsi accès à toutes les caractéristiques des clusters qu'offre ***Clusters-Features*** et pourraient développer leurs propres critères à partir des données sortantes, et ce, de façon totalement interactive. Je recommanderais l'utilisation de l'interface par ligne de commande uniquement lorsque les contraintes techniques l'obligent, ce dernier étant moins complet et moins intuitif que la version par script Python. Un approfondissement de cette interface par ligne de commande est recommandé. L'ajout d'autres indicateurs à la bibliothèque pour la rendre encore plus complète n'est point à proscrire.

6. Bilan personnel et conclusion

La première année scolaire que j'ai réalisé au sein de Télécom Physique Strasbourg a été très généraliste et offrait des enseignements en physique, électronique, informatique et mathématiques. L'unique cours en rapport avec la science des données est celui de l'option facultative de découverte Images, Signaux et Sciences des données (ISSD) où j'ai pu réaliser des modèles de prédiction supervisée sous Python. En probabilités, nous avons modélisé théoriquement l'apprentissage par inférence statistique en passant par le maximum de vraisemblance, fondé sur la théorie Bayésienne. Cependant l'apprentissage non supervisé n'a jamais été enseigné. Il aura fallu que j'acquière les notions de bases du domaine de la science des données quelques mois avant mon stage. Durant cette préparation, j'ai réalisé une multitude de microprojets pédagogiques en rapport avec la science des données dont certains sont publics afin de prendre l'habitude des librairies comme ***Pandas*** ou ***Scikit-learn***. J'ai été très actif durant les cinq derniers mois et fortement motivé par l'apprentissage et la modélisation des bases du domaine. Ce stage m'aura aussi permis de prendre connaissance de l'importance des connaissances en systèmes informatiques, notamment avec la gestion de serveurs via ***SSH***. J'ai pu découvrir plusieurs services récents et utiles dont ***Docker*** et l'intégration continue à travers les différentes discussions que j'ai eu avec Harrison Vernier. Pour que la programmation soit un réel atout en entreprise, il faut aussi savoir déployer son application afin de permettre aux utilisateurs concernés de s'en servir. Sans connaissances des systèmes informatiques, une application ne peut être utile à la majorité. Les différents membres de l'équipe ***SDC*** présents sur le développement de ***FoDoMuST*** ont de fortes connaissances en génie logiciel, ce qui m'aura permis de davantage comprendre les enjeux et profils des métiers associés au développement de logiciels, plus particulièrement ceux qui procèdent à de gros calculs. L'administration et l'architecture d'une telle plateforme représentent un travail qui nécessite une organisation conséquente. L'optimisation se doit d'être extrêmement rigoureuse.

Pour conclure, après une longue année d'enseignements suivis à distance, j'ai été heureux d'avoir pu obtenir un rythme de travail normal effectué en présence sur le site durant tout l'été. J'ai pu améliorer mes connaissances en développement Python et pratiquer avec plaisir des journées de développement comme j'aurais pu le faire sur mon temps personnel. La jonction entre les mathématiques et l'informatique ne cesse de me fasciner et je souhaite approfondir davantage mes compétences sur les nombreux sous-domaines qui en découlent.

7. Bibliographie

Références :

[1] Bernard D., University Paris Ouest, Lab Modal'X, *Clustering Indices*, 2017, pp.3-26

[2] Shyam Kumar K., Dr. Raju G., *Study on Different Cluster Validity*, IEEE vo.13, no. 11, 2018, pp. 9364-9376

[3] Yanchi L., Zhongmou L., Hui X., Xuedong G., Junjie W., *Understanding of Internal Clustering Validation Measures,* IEEE International Conference on Data Mining, 2010.