

Licence Informatique

La Rochelle Université

*Projet plateforme GALACTIC
Les Données Quantitatives Discrètes*



Alexandre Buffard

2022

Utilisation du portrait d'Évariste Galois réalisé par M. Yann Gautreau
aimablement autorisée dans un cadre universitaire

Table des matières

1	Introduction	6
1.1	L'entreprise	6
1.2	La plateforme GALACTIC	7
1.3	Outils utilisés	9
2	Présentation du Stage	10
2.1	Encadrants	10
2.2	Le Sujet de Stage	10
2.2.1	Travaux à effectuer	10
2.2.2	Qu'est-ce qu'une donnée Quantitative Discrète ?	11
2.2.3	Des similitudes avec les données Catégorielles	11
2.3	Mise en place du projet	12
3	Création du Plugin sur les caractéristiques	13
3.1	1er modèle de compréhension : une valeur	13
3.2	2ème modèle de compréhension : un tableau à double entrées	14
3.3	3ème modèle de compréhension : un dictionnaire	15
3.4	4ème modèle de compréhension et modèle retenu : un dictionnaire "inversé"	15
3.5	Tableau comparatif des différentes versions	16
3.5.1	Tableau comparatif	16
3.5.2	Tableau comparatif - analyse	16
3.5.3	Pourquoi le nom index ?	17
3.6	Tests réalisés avec tox.ini	18
3.6.1	Qu'est-ce que tox ?	18
3.6.2	Modifications à apporter	19
4	Objectifs initiaux non encore atteints : raisons et analyse	21
4.1	Raisons	21
4.2	Analyse	21
5	Planning prévisionnel contre planning réel	22
5.1	Planning d'origine	22
5.1.1	Diagramme de Gantt sur le planning d'origine	22
5.2	Planning réel	23
5.2.1	Modifications du planning prévisionnel d'origine	23
5.2.2	Du travail personnel	23
5.2.3	Diagramme de Gantt sur le planning réel (date du 1er juin)	24
5.3	Analyse du planning	24
5.4	Diagramme de Gantt en prévision de la suite du stage (après dépôt et soutenance)	25
6	Conclusion de fin de stage	26
6.1	Résultats du stage et apports à l'entreprise	26
6.1.1	Résultats	26

6.1.2	Ce qui reste à traiter	26
6.2	Ce que j'ai pu apporter à l'entreprise	26
6.3	Évolutions à prévoir	27
6.4	Ce que le stage m'a apporté	27
6.5	Mes projets	28
7	Annexe	29
7.1	Sources	29

Liste des figures

1	logo de la plateforme	10
2	Logo des données quantitatives discrètes sur GALACTIC	10
3	tox.ini du projet	18
4	Réussite des tests tox	18

Résumé / Abstract

Résumé en français

Je m'appelle Alexandre Buffard, j'ai 25 ans et je suis étudiant en 3ème année de Licence Informatique à l'Université de La Rochelle. J'effectue actuellement un **stage de 8 semaines au Laboratoire Informatique, Image et Interaction (L3i)**.

Mon rôle au L3i est d'**apporter 3 nouveaux plugins** (au langage de programmation Python) à la **plateforme GALACTIC** mise en place par différents chercheurs et enseignants-chercheurs ainsi que des doctorants, masters et stagiaires qui participent à l'apport de contenus sur cette plateforme.

GALACTIC (**G**alois **L**Attices **C**oncept **T**heory Implicational system and **C**losures organization) est une **bibliothèque open source** dont l'objectif est de pouvoir mettre en place un framework permettant de travailler sur l'**Analyse Formelle des Concepts**, plus précisément sur l'**algorithme NextPriorityConcept**, ainsi que sur la **Théorie des treillis**.

Récemment, des publications ont lieu dans des revues scientifiques.

Mon rôle en tant que stagiaire au L3i est de **développer 3 plugins traitant des données quantitatives discrètes**, plus précisément :

- un plugin sur les **caractéristiques** des données Quantitatives Discrètes,
- un autre pour la **description** des données Quantitatives Discrètes,
- un dernier sur la **stratégie** à mettre en oeuvre pour des données Quantitatives Discrètes.

Durant cette période de 4 semaines de stages sur les 8, avant la date de rapport, j'ai pu développer **un des trois plugins** pour les données quantitatives discrètes :

Le premier des trois : ses caractéristiques → py-characteristic-quantitative.

Je suis passé par **4 modèles de classes différents** pour la caractéristique, celui qui a été retenu est le suivant :

- **Sa caractéristique** : `*self.__characteristic*` de type `galactic.characteristic.Key*`
- **Son domaine** : `*self.__domain*` de type `str`
- **Un dictionnaire** : `*self.__index*` de type `{ str: int }` avec :
 - comme clé : les éléments un par un du domaine
 - comme valeur : un indice associé dans l'ordre des éléments

Pour finir, je tiens à remercier Yacine GHAMRI-DOUDANE et Muhammad Muzzamil LUQMAN qui m'ont accueilli au L3i, ainsi que Christophe DEMKO, mon maître de stage et Karell BERTET, ma tutrice, enseignants-chercheurs pour leur accompagnement et leur gentillesse.

Je remercie également mes collègues de licence 3 Éléonore THONNEAU, Martin EHLINGER, Jules BRICOU et Kévin VOISIN pour l'aide et le soutien qu'ils m'ont apporté au sein du L3i.

Abstract in English

My name is Alexandre Buffard, I am 25 years old and I am a student in third years of Informatic Licence of the University of La Rochelle. I actually make a **8 weeks internship in the Laboratoire Informatique, Image et Interaction organization (L3i)**.

My rules in L3i is to **bring 3 new plugins** (in the Python programming language) **to the GALACTIC platform** establishing by researcher, teacher-researcher but also doctoral students, master students and other intern who participate in the provision of contents to the GALACTIC platform. This year, we are 5 in L3 intern who work on this platform.

GALACTIC (**G**alois **L**attices **C**oncept **T**heory **I**mplicational system and **C**losures organasation*) is an **open source library** based on **Lattices Theory** and on the **Analysis Formal Concepts principle**, more preciasely on **NextPriorityConcept** algorithm.

It takes into account heterogeneous and complex datas to generate rules of involvement. Recently take place in scienfic journals.

My rule for this internship in the L3i is **to develop 3 plugins which deal with descrete quantitative datas**, more precisely about :

- a plugin about descrete quantitative data's **characteristics**,
- another one about descrete quantitative data's **description**,
- and a last one about descrete quantitative data's **strategy**.

During this period of 4 of internship out the 8, before the report date, I didn't have the time to develop only one of three for descrete quantitative data's plugins. The first one : its characteristics → py-characteristic-quantitative.

I passed between **4 different class models** for the characteristic, the one which was selected is the following :

- **Its characteristics** : `*self.__characteristic* galactic.characteristic.Key*` typed
- **Its domain** : `*self.__domain* str` typed
- **A dictionary** : `*self.__index* { str: int }` typed with :
 - as key : elements one by one in the domain
 - as : an index assigned in the elements order

I wish to thank Yacine GHAMRI-DOUDANE and Muhammad Muzzamil LUQMAN to have welcomed me in the L3i, but also my internship master Christophe DEMKO and my guardianship Karell BERTET for their accompagnement and their kindness.

I thank too my fellows interns in licence 3 Éléonore THONNEAU, Martin EHLINGER, Jules BRICOU et Kévin VOISIN for their help and their welcome in L3i.

1 Introduction

1.1 L'entreprise



La présentation de l'entreprise est issue du site de l'entreprise (*voir Annexe 1.1 Entreprise*) et il s'agit d'un travail fait en commun avec Jules BRICOU, Kévin VOISIN, Éléonore THONNEAU, Martin EHLINGER et moi-même.

Le Laboratoire Informatique, Image et Interaction (L3i) a été créé en 1993, près de 100 personnes travaillent au sein du laboratoire. Il s'agit d'un laboratoire de recherche de Sciences du Numérique de l'Université de La Rochelle, il associe les chercheurs en Informatique de l'IUT ainsi que les chercheurs du Pôle Sciences de l'Université de La Rochelle. Il compte 33 chercheurs et chercheuses, 35 doctorants et doctorantes, 4 personnes permanentes de soutien à la recherche et 25 personnes sur projets et est dirigé par Monsieur Yacine GHAMRI-DOUDANE.

Le laboratoire L3i fait partie de réseaux de recherches régionaux (Fédération CNRS MIRES, ERT " Interactivité Numérique "), nationaux (GDR I3 et GDR IRIS) et internationaux (IAPR) dans les secteurs de visibilité de son action scientifique, autour d'un projet stratégique lié à la gestion intelligente et interactive des contenus numériques. Cela est renforcé grâce à une politique de volontariste de participations ou de pilotages de projets de recherches labélisés (ANR, PCRD, ...), au sein desquels le laboratoire occupe couramment une position de leadership.

Le Laboratoire est également membre de l'Alliance Big data lancée en 2013 pour favoriser le développement, en France, de nouveaux services et projets dans ce domaine.

Ses travaux sont menés en partenariat avec une dizaine de centres de recherches nationaux (dont l'INRIA, institut spécialisé). Le L3i entretient par ailleurs des liens privilégiés avec de nombreux centres de recherche à travers le monde. Il est également engagé dans près d'une vingtaine de partenariats industriels sur l'ensemble du territoire français.

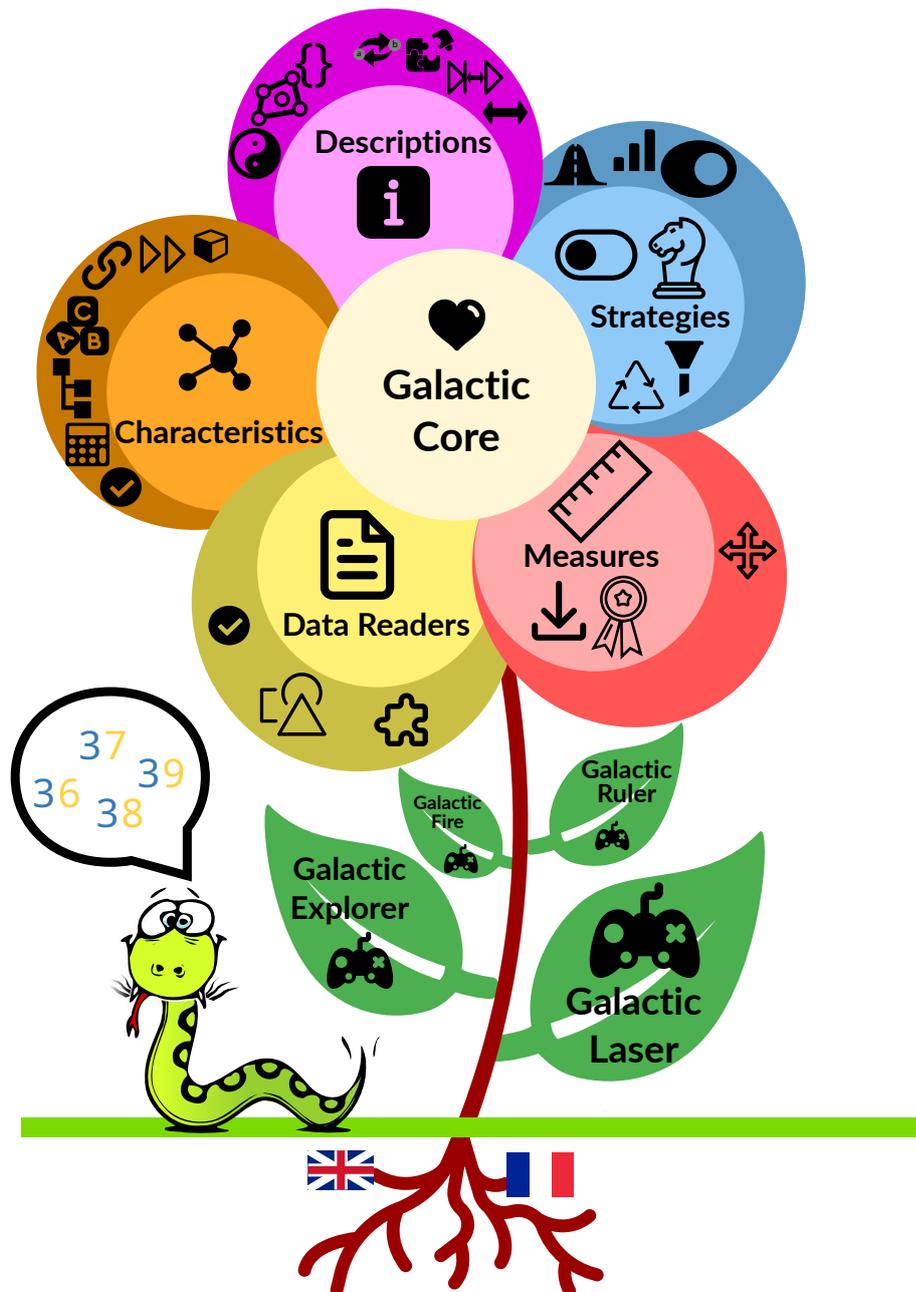
Le laboratoire du L3i est divisé en 3 équipes: * **Modèles et Connaissances** * **Images et Contenus** * **Dynamique des systèmes et Adaptativité**

Mon stage se déroule **au sein de l'équipe Modèles et Connaissances** et **plus précisément autour du projet GALACTIC** de cette équipe.

1.2 La plateforme GALACTIC

La présentation de la plateforme GALACTIC est issue du diaporama de présentation de l'architecture de la plateforme (voir Annexe : 1.2 Entreprise) et il s'agit d'un travail fait en commun avec Jules BRICOU, Kévin VOISIN, Éléonore THONNEAU, Martin EHLINGER et moi-même

GALACTIC est l'acronyme de **G**alois **L**attices **C**oncept **T**heory **I**mplicational system and **C**losures. L'objectif de la plateforme est de pouvoir mettre en place un framework permettant de travailler sur l'Analyse Formelle des Concepts ainsi que la Théorie des treillis. La plateforme GALACTIC est structurée de la façon suivante:



- Un noyau, représenté par le coeur de la fleur, qui contient les opérations de base et structures de données et qui implémente un nouvel algorithme (NextPriorityConcept) inspiré des pattern structures.

- Des plugins, représentés par les pétales de la fleur:
 - **Caractéristiques** : Booléen, Numérique, Catégorique, String, Séquentielle, Chain, Triadic. Ils permettent de définir les caractéristiques.
 - **Descriptions** : Booléen, Logique, Catégorique, String (regex), String (distance), Chain, Séquentielle, Séquentielle (distance), Triadic. Ils définissent les prédicats et les espaces de description utilisés pour représenter et définir les données avec précision.
 - **Stratégies** : Booléen, Logique, Catégorique, Numérique, String, String (distance), Chain, Séquentielle, Séquentielle (distance), Triadic. Ils définissent la manière utilisée pour explorer les données, il utilise des descriptions pour générer des prédécesseurs pour chaque concept dans le treillis.
 - **Mesures** : predecessor Cardinality, sucessor Cardinality, Confidence. Ce sont des paramètres des stratégies de filtrage prédéfinis dans la librairie core.
 - **Data Readers** : Ils sont utilisés pour lire différents types de fichiers de données. Le moteur de base détecte le type de fichier en utilisant son extension. Les plugins Data Readers existants sont: YAML, JSON, CSV, TOML, INI, TXT, SLF, DAT, CXT.
 - **Localization** : Ils sont utilisés pour traduire les applications dans d'autres langues.

- Des applications, représentées par les feuilles de la fleur, qui sont développées afin d'utiliser la librairie de la plateforme. Ce sont des interfaces pour l'utilisateur:
 - **GALACTIC Laser** : pour construire les treillis et explorer les données.
 - **GALACTIC Explorer** : pour explorer de façon interactive les treillis construits.
 - **GALACTIC Ruler** : pour extraire les règles d'implication.
 - **GALACTIC Fire** : pour exécuter un système de règles.

- La bulle au dessus du serpent indique les différentes versions de python avec lesquelles la plateforme GALACTIC est compatible.

- Au niveau des racines on peut voir les différentes langues disponibles sur la plateforme.

1.3 Outils utilisés

- **MSys64** : qui m'a servi dans un premier temps pour **l'installation de la dernière version de GALACTIC (0.5.0)**
 - → *MSys64 est une console dont l'utilité est la même que l'invite de commande, mais qui est plus complète.*

- **PyCharm** : pour **développer les plugins en langage Python**
 - → *Pycharm est un logiciel de programmation en langage Python.*

- **Mozilla Firefox** : qui m'a servi pour **les recherches et l'utilisation de gitlab**
 - → *Mozilla Firefox est un moteur de recherche*

- **L'invite de commande** : pour **l'import des dossiers/fichiers** qui sont sur gitlab
 - → *L'invite de commande permet de faire des manipuler via des commandes sur son ordinateur.*
 - * Par exemple : ouvrir, supprimer un fichier, créer un dossier, etc...

- **Wsl et la console Ubuntu associée** : qui m'a servi au **développer dans Visual Studio Code**, à la manipulation de **tox** et de **git** (voir dans la partie "Qu'est-ce que tox ?" et "Qu'est-ce que git ?")
 - → *une machine virtuelle qui permet de manipuler divers types de consoles ou d'invites de commande (Windows, Linux/Ubuntu)*

2 Présentation du Stage

2.1 Encadrants

- **Maître de Stage : Monsieur Christophe DEMKO**, enseignant-chercheur à l'Université de Sciences et Technologies de La Rochelle.
- **Tutrice de Stage : Madame Karella BERTET**, enseignante-chercheuse à l'Université de Sciences et Technologies de La Rochelle.

2.2 Le Sujet de Stage

2.2.1 Travaux à effectuer

Mon sujet de stage a consisté à **développer 3 plugins sur les données Quantitatives Discrètes** pour la plateforme GALACTIC :



Figure 1: logo de la plateforme

- Un Premier sur ses **caractéristiques** : py-characteristic-quantitative
- Un Second sur sa **description** et les prédicats associés : py-description-quantitative-classic
- Un Troisième sur la **stratégie** à mettre en place : py-strategy-quantitative-classic-basic



Figure 2: Logo des données quantitatives discrètes sur GALACTIC

Ces trois plugins ont été développés en **langage Python objet** avec Napoleon, une **extension de Sphinx pour générer du docstring en "style Numpy"**. Cette dernière va convertir les commentaires au format "Docstring style NumPy" pour le restructurer en un texte formaté juste avant que Sphinx n'agisse lors de la génération de la-dite documentation.

source : Voir Annexe → 2.1 Le Sujet de Stage - Napoléon et Docstring style Numpy. Voir Annexe également → 2.1 Le Sujet de Stage - Exemples de Docstring style Numpy.

2.2.2 Qu'est-ce qu'une donnée Quantitative Discrète ?

Donnée Quantitative : Une donnée est dite quantitative lorsque l'on fait appel à une **notion de quantité et/ou de mesure**. Elle peut être **discrète ou Continue**.

Exemples : * *La masse d'un véhicule (en grammes/kilogrammes/tonnes)* * *Le temps de réalisation d'une tâche (en secondes/minutes/heures)*

Donnée Quantitative Discrète : Par rapport aux données quantitatives "classiques", une donnée quantitative est dite **discrète** lorsqu'il est question d'un nombre fini d'éléments (d'un **espace fini**) avec **une borne inférieure et une borne supérieure**. Il est question, en général, d'**énumérer** ces éléments.

Exemples :

- les notes sur les sites de voyage (1, 2, 3, 4 ou 5)
- l'alphabet (entre A et Z)

Donnée Quantitative Continue : Contrairement aux données Quantitatives Discrètes, les données Quantitatives Continues se situent dans un espace potentiellement infini (qui ne possède pas de limite/de borne). ce qui veut dire qu'une valeur dans cet espace peut théoriquement prendre une infinité de valeurs.

Exemples :

- Nombre de personnes dans une liste
- Nombre de personnes dans une salle de concert

2.2.3 Des similitudes avec les données Catégorielles

Les données quantitatives discrètes ont des **similarités avec les données catégorielles**.

Les **données catégorielles, ou qualitatives**, mesurent **plusieurs états** qu'un élément (ou une donnée/variable) peut prendre.

Exemples :

- les variables booléennes (True et False)
- le sexe d'une personne (Homme ou Femme)

Dans un cas comme dans l'autre, les données ne peuvent **avoir qu'une valeur parmi les "celles possibles"**.

Pour cette raison, je me suis aidé des variables catégorielles, la classe **Category** qui est déjà définie sur la plateforme GALACTIC, ce qui m'a aidé à la création des plugins de données quantitatives discrètes.

2.3 Mise en place du projet

Tout d'abord, il a fallu que je fasse l'**installation des différentes extensions via MSys64**.

Puis j'ai suivi la procédure d'installation de la documentation GALACTIC.

Voir annexe : 2.3 : Mise en place du projet

Problématiques rencontrées : L'installation et la compréhension de l'écosystème GALACTIC ont été longs et difficiles.

Il s'agit d'une nouvelle méthode d'analyse et de fouille de données complexe, cela demande un certain temps d'adaptation et d'apprentissage conséquent afin d'appréhender le contexte.

Aussi, ce domaine n'a pas été abordé dans mon cursus université.

Pour m'aider j'ai aussi utilisé l'utilitaire d'installation Galactic créé par Martin EHLINGER, l'un des stagiaires qui est présent pour GALACTIC. L'utilitaire d'installation a échoué dans un premier temps, pour finalement fonctionner la veille du rendu du rapport de stage.

Avant la réussite de l'installateur, j'ai fini par installer WSL pour pouvoir accéder et manipuler l'environnement GALACTIC, ainsi que l'environnement virtuel Python qui ne voulait pas non plus fonctionner sur ma machine. WSL m'a permis la manipulation de tox et de git que j'aborderai plus loin lors de leurs manipulations.



3 Création du Plugin sur les caractéristiques

Pour ce 1er plugin, je suis d'abord passé par divers modèles de compréhension et de recherches pour obtenir une donnée quantitative discrète adaptée à la programmation objet en Python.

3.1 1er modèle de compréhension : une valeur

La **1er modèle de compréhension** d'une donnée quantitative discrète a donné le résultat suivant :

Sa caractéristique : `self._characteristic` → type `galactic.characteristics.Key`

- Exemple : "x"

Un domaine : `self._domain` → type `str`

- Exemple : "abc"

Ces deux dernières propriétés sont communes aux **données catégorielles** et le seront à **tous les "modèles"** de données quantitatives discrètes.

Une **valeur comprise entre les deux bornes du domaine** : `self._value` → type `int`

- Exemple pour `self._domain = "abc"` : `self._value = "b"`

Problématique : Je pensais, à tort, qu'une variable quantitative discrète ne pouvait être **que numérique**, d'où le nom de variable choisi `*self._value*`.

J'ai compris que ce n'était pas du tout ce qui était entendu par là.

3.2 2ème modèle de compréhension : un tableau à double entrées

Sa caractéristique : `self.__characteristic` → type *galactic.characteristics.Key*

- Exemple : “a”

Un domaine : `self.__domain` → type *str*

- Exemple : “xyz”

Un tableau à double entrées sous la forme **[indice, valeur]** : `self.__values` → type `[[int, str]`

- Exemple pour `self._domain = “xyz”` : `self.__values` : `[[0, ‘x’], [1, ‘y’], [2, ‘z’]]`

Une valeur minimale : `self.__minimumValue` → type *str*

- Exemple pour `self._domain = “xyz”` : `self.__minimumValue = “x”`

Une valeur maximale : `self.__maximumValue` → type *str*

- Exemple pour `*self._domain = “xyz”*` : `self.__maximumValue = “z”`

J’ai fini par comprendre que le résultat attendu était un **ensemble d’indices** et ai donc initialement pensé qu’un tableau à double entrées était la solution la plus adaptée à la situation.

Problématiques : Ce type de structures de données est plus intéressant, cependant, il n’est **pas adapté à la méthode de recherche attendue** (méthode de recherche dont je parlerai dans le modèle de compréhension qui a été retenu).

3.3 3ème modèle de compréhension : un dictionnaire

Sa caractéristique : `self.__characteristic` → type *galactic.characteristics.Key* * Exemple : “x”

Un domaine : `self.__domain` → type *str*

- Exemple : “abc”

Un dictionnaire{indice: valeur} : `self.__values` → type `{ int: str }`

- Exemple pour `*self._domain = “abc”*` : `self._values = { 0: ‘a’, 1: ‘b’, 2: ‘c’ }`

****Une valeur minimale :** `self.__minimumValue` → type *str*

- Exemple pour `self._domain = “abc”` : `self.__minimumValue = “a”`

Une valeur maximale : `self.__maximumValue` → type *str*

- Exemple pour `*self._domain = “abc”*` : `self._maximumValue = “c”`

J’ai opté pour **un dictionnaire**, comme cela il est **plus facile d’accéder à un élément via une recherche** qu’avec un tableau à double entrées.

Problématiques : Je pensais, à tort, que le but de cette structure était de récupérer les valeurs à l’aide des indices.

Mais pour le type de recherches attendues pour une donnée quantitative discrète, nous voulons rechercher l’indice d’un élément (donc sa position dans le domaine). Nous devons donc échanger les clés et les valeurs du dictionnaire.

3.4 4ème modèle de compréhension et modèle retenu : un dictionnaire “inversé”

Le modèle de compréhension est le suivant pour les caractéristiques d’une donnée quantitative discrète :

Les caractéristiques d’une donnée quantitative sont finalement composés de :

****Sa caractéristique :** `self.__characteristic` → type *galactic.characteristics.Key*

- Exemple : “x”

****Un domaine :** `self.__domain` → type *str*

- Exemple : “abc”

Un dictionnaire “inversé” → {valeur: indice} : `self.__index` → type `{ str: int }`

- Exemple avec `self._domain = “abc”` : `self._index = {“a”: 0, “b”: 1, “c”: 2}`

3.5 Tableau comparatif des différentes versions

3.5.1 Tableau comparatif

caractéristiques	1ère version	2ème version	3ème version	4ème version
EN COMMUN				
caractéristique	✓	✓	✓	✓
domaine	✓	✓	✓	✓
minimum/maximum	✓	✓	✓	
PROPRIÉTÉS				
valeur	✓			
tableau		✓		
dictionnaire			✓	✓

3.5.2 Tableau comparatif - analyse

! [tip] Dictionnaire VS tableau Pour une donnée quantitative discrète, nous voulons **savoir à quelle position se trouve un élément dans le domaine donné en argument et s'il s'y trouve bien**. Pour cela, **le dictionnaire est une bonne solution pour rechercher un élément** : par exemple "b" dans l'exemple précédent, il retournera 1 :

```
>>> print(self._index("b"))
1
```

:::

Dans le **cas d'un tableau** cela aurait été l'**inverse pour une recherche**, on aurait dû donner un indice pour récupérer l'index et la valeur, ce qui n'est pas ce qui est recherché pour notre type de données. C'est d'ailleurs **pour pouvoir accéder à cet indice** d'un élément (sa position) qu'il faut **créer une fonction index** telle que ci-dessous :

```
def index(self, value: str) -> int:
    try:
        return self.__index[value]
    except KeyError:
        raise ValueError(f"'{value}' not in domain") # f est un formatage
        ↪ d'affichage
```

Ce qui donnerait les résultats suivants :



```
# exemples utilisés pour les tests de la classe Quantitative

>>> from galactic.characteristics import Key
>>> from galactic_characteristic_quantitative import Quantitative

>>> quantitative = Quantitative(characteristic=Key(name="x"),
↳ domain="ABC")
>>> quantitative.index('B')
     1
----
>>> try:
...     print(quantitative.index('E'))
... except ValueError as error:
...     print(error)
     'E' not in domain
----
>>> quantitative({"x": "A"})
     ['A']
```

3.5.3 Pourquoi le nom index ?

J'utilise **self._index** comme nom de variable pour un dictionnaire pour les raisons suivantes :

- Dans un dictionnaire, il s'agit d'un **système d'indexation de données** organisé dans un ordre donné.
- mais aussi et surtout : Il s'agit d'une **convention tacite** → lorsque l'on veut **avoir accès à l'indice d'un élément** dans un ensemble de données tel que **{clé: valeur}** avec :
 - comme clé le-dit élément (ex : chaîne de caractère, nombre ...)
 - la valeur qui est l'indice de l'élément en partant de 0.

3.6 Tests réalisés avec tox.ini

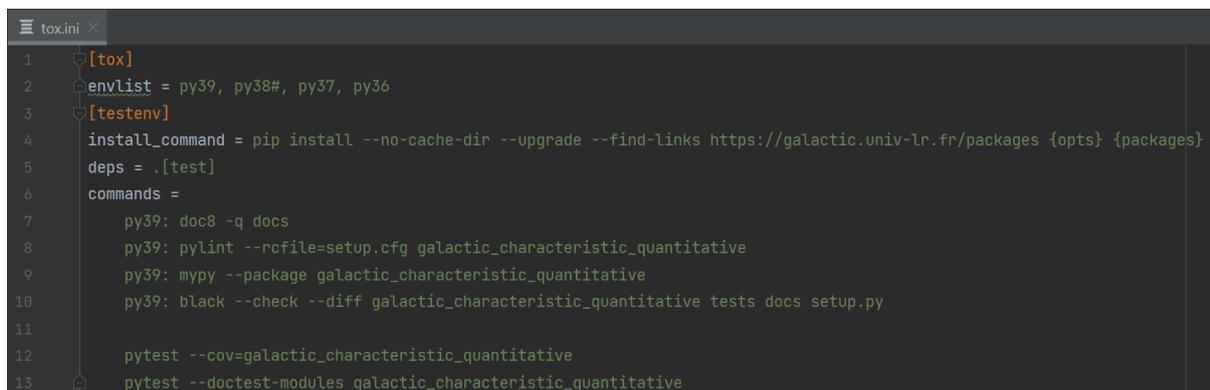
3.6.1 Qu'est-ce que tox ?

voir Annexe : 3.6.1 Qu'est-ce que tox ?

tox est une **machine virtuelle** lançant des tests donnés dans le fichier **tox.ini**. Cela permet :

- de **vérifier que les packages**, nécessaires au bon fonctionnement du/des programme(s), **s'installent correctement**,
- de **lancer et de configurer des tests** dans chaque environnement Python (3.9, 3.8, 3.7, 3.6, etc...),
- d'**agir comme une interface** pour les serveurs d'intégration continus.

Ainsi la commande **tox** fait lien entre le fichier **tox.ini** et **setup.py** :



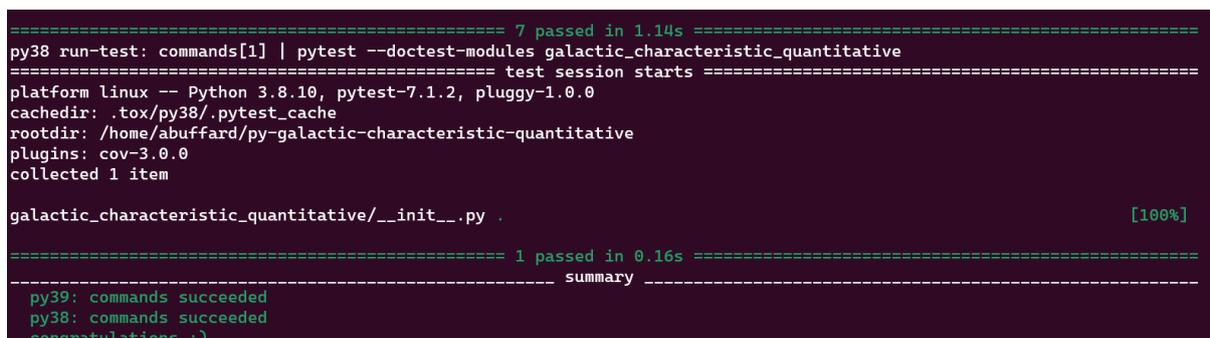
```
tox.ini
1 [tox]
2 envlist = py39, py38#, py37, py36
3 [testenv]
4 install_command = pip install --no-cache-dir --upgrade --find-links https://galactic.univ-lr.fr/packages {opts} {packages}
5 deps = .[test]
6 commands =
7     py39: doc8 -q docs
8     py39: pylint --rcfile=setup.cfg galactic_characteristic_quantitative
9     py39: mypy --package galactic_characteristic_quantitative
10    py39: black --check --diff galactic_characteristic_quantitative tests docs setup.py
11
12    pytest --cov=galactic_characteristic_quantitative
13    pytest --doctest-modules galactic_characteristic_quantitative
```

Figure 3: tox.ini du projet

Le fichier **setup.py**, lui, s'occupe du **paramétrage du projet**, permettant le **bon fonctionnement du plugin**.

Suite à cette série de tests, j'ai envoyé le plugin "py-galactic-characteristic" sur gitlab avec les commandes **git push** et **git commit -am "update"**.

Dans un premier temps, les **tests sur gitlab ont bien voulu fonctionner**.



```
===== 7 passed in 1.14s =====
py38 run-test: commands[1] | pytest --doctest-modules galactic_characteristic_quantitative
===== test session starts =====
platform linux -- Python 3.8.10, pytest-7.1.2, pluggy-1.0.0
cachedir: .tox/py38/.pytest_cache
rootdir: /home/abuffard/py-galactic-characteristic-quantitative
plugins: cov-3.0.0
collected 1 item

galactic_characteristic_quantitative/___init___py . [100%]

===== 1 passed in 0.16s =====
----- summary -----
py39: commands succeeded
py38: commands succeeded
congratulations :)
```

Figure 4: Réussite des tests tox

Qu'est-ce que git ? Git est un **système de contrôle distribué** libre et open source qui a été designé pour gérer des petits comme de grands projets de manière rapide et efficace.

Voir Annexe : 3.6.1.1 Qu'est-ce que git ?

3.6.2 Modifications à apporter

Le lendemain du dépôt j'ai reçu diverses instructions de la part de Christophe DEMKO afin d'apporter des améliorations à certains fichiers, notamment :

- Pour le `**_main.py**`:
 - Rajouter des commentaires, afin de documenter (avec le Docstring style NumPy) le site de documentation GALACTIC.
 - Compléter la fonction `_str_()` pour qu'elle colle plus à l'affichage que l'on attendrait d'une donnée quantitative discrète.
- Pour le **test_quantitative** :
 - Dans la fonction `test__str_()`, modifier le contenu des tests `assertEqual` en conséquence des modifications précédemment citées.
 - * la fonction `assertEqual` vérifie qu'une fonction donnée en paramètres affiche bien un résultat donné également en paramètres.
- Pour le **setup.py**, rendre la syntaxe plus ouverte pour que les éléments des tableaux listés soient sur plusieurs lignes plutôt qu'allignés sur une seule et même ligne comme ci-dessous :

```
 packages=[  
    "galactic_characteristic_quantitative",  
],
```

Problématiques rencontrées lors de ces modifications : Avec ces modifications, les **tests tox** n'ont pas voulu fonctionner correctement :

- Pour `**_main.py**` certains commentaires étaient considérés comme "inutiles" pour les **tests pylint** (teste la syntaxe Python)
 - **Résolution du problème** : il a donc fallu rajouter en commentaires au-dessus du code signifiant que l'on demande à pylint d'ignorer l'erreur apparue :

```
 # pylint: disable=pointless-string-statement
```

- Pour les modifications de **setup.py**, le fait d'avoir un tableau tel que demandé ci-dessus pose problème pour la mise en page attendue par tox.
 - **Problème encore non résolu en ce jour du 3 juin** : laissé en suspens depuis le 31 mai, afin de me concentrer sur mon rapport de stage à rendre et ma présentation de soutenance.

4 Objectifs initiaux non encore atteints : raisons et analyse

A l'heure actuelle, seul le plugin sur les caractéristiques des données quantitatives discrètes a été développé et est partiellement fonctionnel.

4.1 Raisons

Tout d'abord, il faut savoir que mon stage n'a commencé **que le 2 mai** (par rapport à la plupart des stagiaires de L3 qui ont commencé le 11 avril) et ce dernier **dure jusqu'au 24 juin**.

Ce qui fait que le jour du rendu du rapport de stage, pour ma part, **seulement 4 semaines** se seront écoulées, ce qui veut dire que je ne serai qu'à la **moitié du temps de stage**.

Mon stage est donc **loin d'être terminé**, j'aurai encore le temps de **développer un plugin sur les deux, voire même les deux** selon le temps que me prendront les recherches sur les deux plugins restants (voir le diagramme de Gantt plus loin dans la partie "planning prévisionnel") D'autant plus que maintenant je maîtrise suffisamment l'environnement de travail pour avancer plus efficacement.

4.2 Analyse

Pour commencer, les **recherches et la compréhension du sujet ne sont pas à négliger pour avancer dans le développement du plugin**, cela est important afin d'**éviter des erreurs**, il faut donc prendre le temps nécessaire lorsque cela est nécessaire.

Qui plus est, la **création de plugins est pour moi une nouveauté**, il a fallu que je **m'adapte au format, aux nouvelles méthodes de travail et à l'organisation que cela engendre**.

5 Planning prévisionnel contre planning réel

5.1 Planning d'origine

Je bénéficie d'**aménagements liés à mon état de santé**, dont celui de faire mon **stage sous forme de mi-temps** en prolongeant la période de stage (en termes de semaines) :

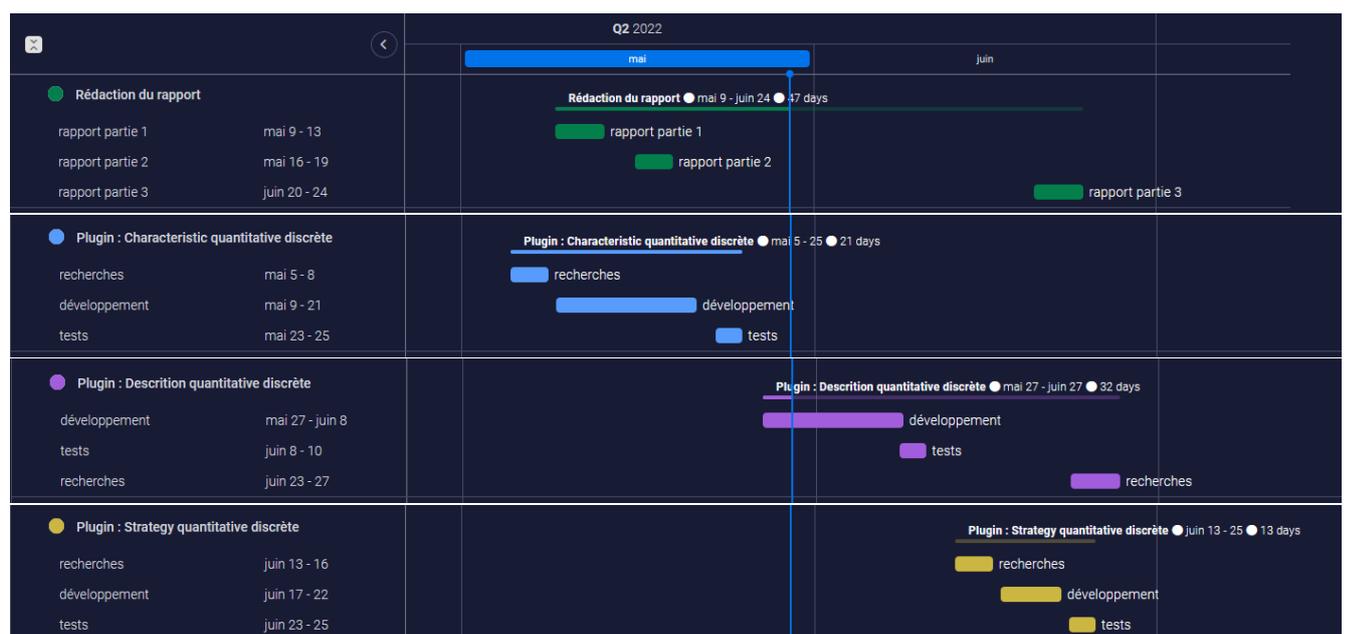
Date de début de stage : 2 mai 2022 **Date de fin de stage** : 24 juin 2022 (8 semaines)

Horaires :	matin (début)	matin (fin)	après-midi (début)	après-midi (fin)
lundi	8h30	12h00	-	-
mardi	8h30	12h00	-	-
mercredi	8h30	12h00	-	-
jeudi	8h30	12h00	-	-
vendredi	8h30	12h00	-	-

Durée hebdomadaire maximale de présence dans l'organisme :18h

5.1.1 Diagramme de Gantt sur le planning d'origine

Ce diagramme de Gantt initial représente le temps prévu pour chaque développement des plugins. Pour ce faire, j'ai divisé la période de stage en 3 parties plus ou moins égales par rapport au planning d'origine.



5.2 Planning réel

5.2.1 Modifications du planning prévisionnel d'origine

Lors de la première semaine, il a été révélé que j'étais la seule personne au L3i à **démarrer ma journée à 8h30**. Ce qui posait une **problématique d'accès au laboratoire**, notamment le lundi et le vendredi où personne n'est disponible pour m'ouvrir avant 9h00.

J'ai donc fait une **demande de changement de planning** à la fin de cette semaine. Maintenant je travaille **tous les jours de 9h00 à 12h30** :

Horaires :	matin (début)	matin (fin)	après-midi (début)	après-midi (fin)
lundi	9h00	12h30	-	-
mardi	9h00	12h30	-	-
mercredi	9h00	12h30	-	-
jeudi	9h00	12h30	-	-
vendredi	9h00	12h30	-	-

Durée hebdomadaire maximale de présence dans l'organisme :18h

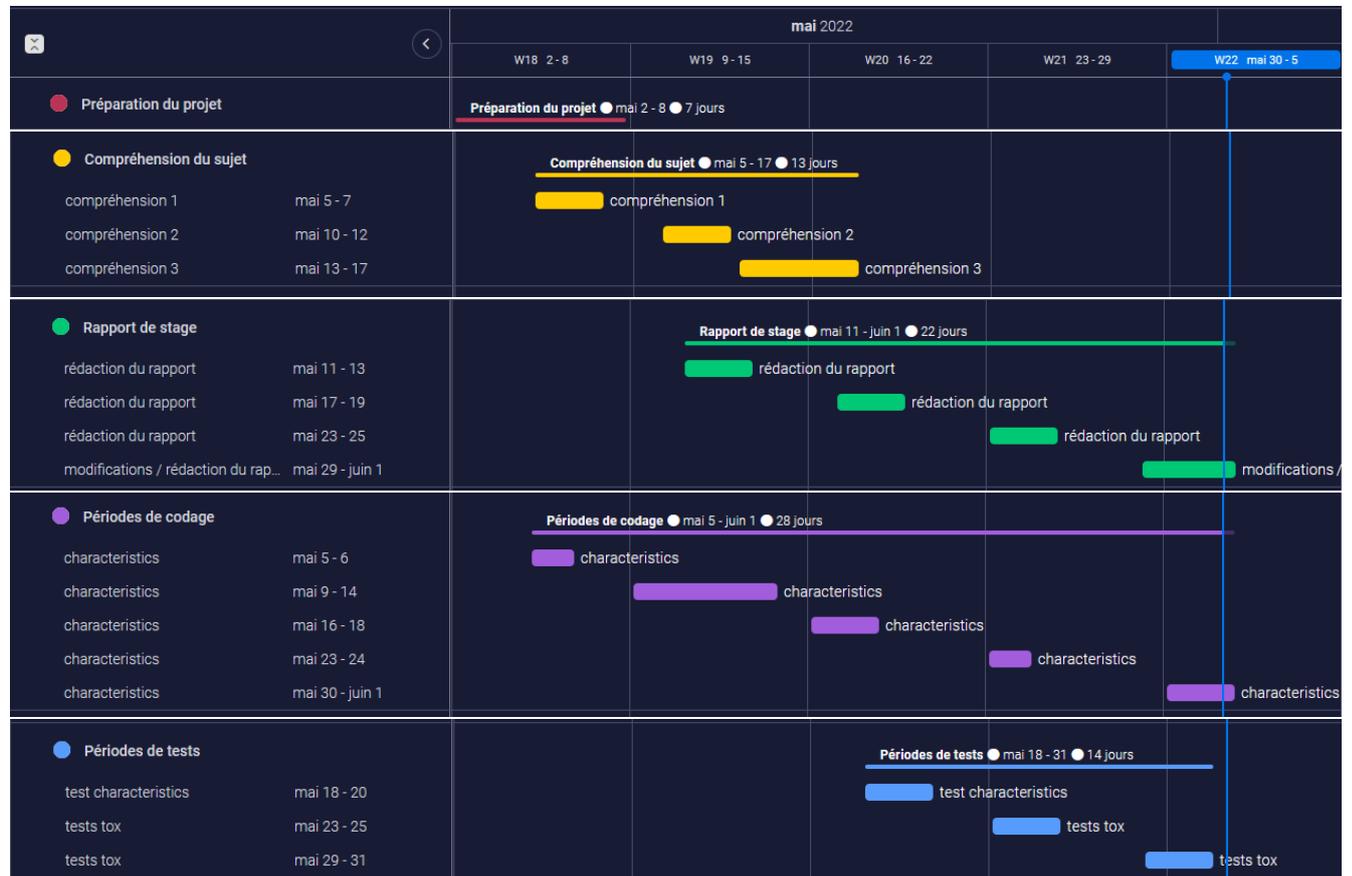
5.2.2 Du travail personnel

Il m'arrive régulièrement d'avancer sur un travail l'après-midi même commencé le matin, en général **entre 14h00/14h30 et 17h00/17h30/18h00**.

Sinon je travaille à la rédaction de mon **rapport** ou le **diaporama pour la soutenance**.

5.2.3 Diagramme de Gantt sur le planning réel (date du 1er juin)

Ce diagramme de Gantt n'en forme qu'un seul, il est divisé en plusieurs parties pour une meilleure lisibilité.



5.3 Analyse du planning

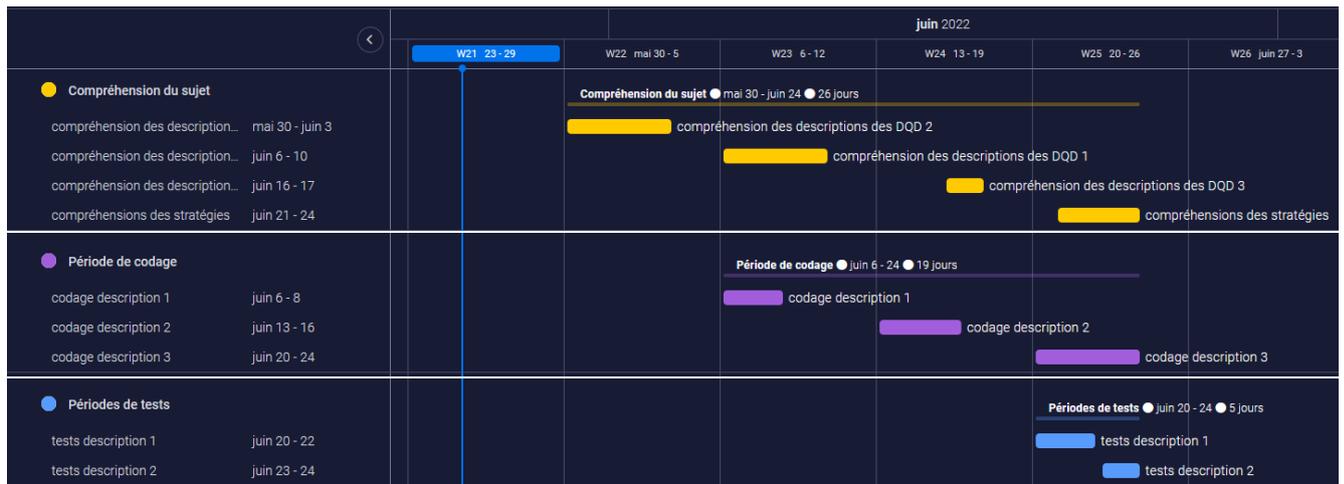
On voit très clairement que **la recherche prend plus de temps** par rapport à ce qui a été prévu initialement, de même pour les **installations de l'environnement de travail et de GALACTIC** (compte tenu des difficultés rencontrées), paramètres que je n'avais pas / mal pris en compte en prévisions du déroulement du stage.

Il a fallu que je m'**habitue aux nouvelles méthodes de programmation** en Python auxquelles je n'étais pas habitué, il a fallu aussi que j'**apprenne à manipuler** les commandes **liées à tox** ainsi que **celles liées à Git**.

5.4 Diagramme de Gantt en prévision de la suite du stage (après dépôt et soutenance)

Même si la date de dépôt du rapport est fixée au 3 juin et que la soutenance a lieu le 10 juin 10h30, mon stage continue, et cela jusqu'au 24 juin.

Voici donc un diagramme de Gantt en prévision des semaines suite au dépôt du rapport et à la soutenance :



Je l'ai créé en fonction du rythme auquel j'ai travaillé en début du stage jusqu'au rendu du rapport. Il est fort probable que :

- le temps de recherches soit moindre que pour le premier plugin.
- de même pour le temps de développement, maintenant que je suis habitué à l'environnement Python et GALACTIC
- que je ne puisse pas développer à temps le plugin sur les stratégies.

6 Conclusion de fin de stage

6.1 Résultats du stage et apports à l'entreprise

6.1.1 Résultats

Lors de ma période de stage, j'ai pu **créer un plugin partiellement fonctionnel** parmi les trois qui m'étaient assignés. J'avance bien dans la réalisation de mon rapport et du diaporama prévu pour le jour de la soutenance.

6.1.2 Ce qui reste à traiter

Il me reste **** 2 plugins à réaliser**** :

- un 1er sur la **description** avec les prédicats,
- un second sur les **stratégies**.

Incluant ainsi les **recherches et les tests à réaliser** autour du **développement de ces plugins** comme j'ai eu à le faire pour les caractéristiques.

6.2 Ce que j'ai pu apporter à l'entreprise

Lors du stage, j'ai pu apporter à l'entreprise au moins un plugin parmi les 3 qui m'étaient assignés.

A part cela, il y a eu une véritable entre-aide entre les stagiaires de Licence 3 informatique. Nous avons pu nous apporter mutuellement des échanges constructifs et positifs, nous partageons nos expériences au sujet du stage, tout comme nous échangeons nos expériences personnelles pour nous guider, nous donner des conseils, dans un sens comme dans l'autre :

Au sujet du stage :

- la plateforme,
- la programmation Python,

En dehors du stage :

- orientations d'études

J'ai également aidé, en dehors des stagiaires travaillant sur la plateforme, une étudiante en filiaire Gestion qui devait apporter des modifications sur un site Web. Je lui ai apporté quelques conseils en programmation HTML/CSS/PHP/MySQL.

6.3 Évolutions à prévoir

A ce jour du 3 juin, je n'ai produit qu'un seul des 3 plugins et le rendre partiellement fonctionnel, celui sur les **caractéristiques**. Il faudrait donc déjà développer les plugins restants.

Cependant il me reste encore suffisamment de temps après la date limite de dépôt et de soutenance, encore 3 semaines de stage au L3i afin de **finir de développer un maximum des plugins restants**.

6.4 Ce que le stage m'a apporté

Ce stage m'a apporté différentes connaissances et améliorations des compétences dans les domaines suivants : * la programmation Python,

- la manipulation de logiciels/de plugins.

J'ai acquis divers connaissances dans le domaine de la recherche plus que du développement avec les notions d'Analyse Formelle des Concepts sur lesquelles se base GALACTIC.

Après une réunion sur le fonctionnement d'un laboratoire de recherche en informatique, tenue par Karell BERTET et Christophe DEMKO, j'ai appris comment fonctionnait une équipe de recherches, son fonctionnement et son organisation.

Une notion qui a attisé ma curiosité lors de cette réunion est le **système d'appel à projet**. À ce moment, il a été aussi question de **verrous scientifique**, cette notion m'a beaucoup intéressé et plus particulièrement un exemple, qui me parle et qui m'a marqué, un **verrou scientifique traitant de la pharmacovigilance**. Si je devais devenir chercheur, probablement que j'aimerais étudier la question, puisque je suis quotidiennement en contact avec cette notion à travers mon état de santé et que cette notion de pharmacovigilance pourrait aider à plus facilement et plus rapidement trouver un traitement pour une pathologie grave, pharmarco-résistante* ou non.

définition de pharmaco-résistance : persistance des crises malgré un traitement bien conduit. Voir Annexe : 6.4 Ce que le stage m'a apporté - pharmacorésistance.

Ce fut une expérience très intéressante qui m'a permis de **mieux connaître mes collègues de licence et les enseignants-chercheurs**, mais surtout d'**appréhender de nouvelles notions que je pourrais développer à l'avenir**.

6.5 Mes projets

Après ce stage en entreprise au L3i, j'ai l'intention de changer de formation pour me diriger vers une licence professionnelle parmi les deux proposées à La Rochelle orientées web.

Celle qui m'intéresse le plus est la **LP MIAW : parcours Développeur Web Full Stack**. J'ai déjà proposé ma candidature pour cette formation et également à l'autre licence professionnelle dans le domaine du Web de l'Université de La Rochelle : Développeur Web **parcours Web Designer Intégrateur**.

7 Annexe

7.1 Sources

1.1 L'entreprise - Présentation du L3i source : <https://l3i.univ-larochelle.fr/Presentation-317>

1.2 La plateforme GALACTIC source : <https://galactic.univ-lr.fr/slides/architecture/Galactic-Architecture-slides-20min-complete.pdf>

2.1 Le Sujet de Stage - Napoleon et Docstring style Numpy source : <https://www.sphinx-doc.org/en/master/usage/extensions/napoleon.html>

2.1 Le Sujet de Stage - Exemples de Docstring style Numpy source : https://sphinxcontrib-napoleon.readthedocs.io/en/latest/example_numpy.html

2.2.2 Qu'est-ce qu'une donnée Quantitative Discrète ? source : <https://blocnotes.iergo.fr/breve/categorielle-quantitative-discrete-ou-continue/>

2.3 Mise en place du projet - guide d'installation et de développement de GALACTIC source guide d'installation : <https://galactic.univ-lr.fr/guides/guide-installation/Galactic-Installation-Guide-latest.pdf> source guide de développement : <https://galactic.univ-lr.fr/guides/guide-developer/Galactic-Developer-Guide-latest.pdf>

3.6.1 Qu'est-ce que tox ? source : <https://tox.wiki/en/latest/>

3.6.1.1 Qu'est-ce que git ? source : <https://git-scm.com/>

6.4 Ce que le stage m'a apporté - pharmacorésistance source : <https://www.em-consulte.com/article/105077/texte-des-experts-definition-de-la-pharmaco-resist>