Promotion 2020
5e année, Majeure IF

iNex Circular
Paris/France

Tuteur école :
GARCIN Mathieu

Tuteur entreprise :
POISSON Titouan

Data Engineer chez iNex Circular

SAOUNERA Mohamed

Remerciements

Tout d’abord, je voudrais remercier mon maitre de stage, M. Titouan POISSON, chief data officer d’iNex, ainsi que les fondateurs d’iNex. Ils ont su me faire confiance lors de cette aventure dans le monde professionnel et ont partagé leurs connaissances de manière très pédagogique. Je les remercie aussi pour leur disponibilité et la qualité de leur encadrement.

Je remercie aussi l’ensemble des personnes travaillant chez iNex pour l’accueil et la gentillesse dont ils ont fait preuve à mon égard durant ces mois passés ensemble.

J’adresse ensuite mes remerciements au corps professoral de l’Ecole Supérieur d’Ingénieurs Léonard de Vinci qui m’ont fourni les outils nécessaires au bon déroulement de mon stage.

Un grand merci à mes parents pour leurs conseils ainsi que pour leur soutien inconditionnel à la fois moral et économique.

Table des matières

Abréviations et sigles employés

- **SAU** : Surface Agricole Utile
- **PAC** : Politique Agricole Commune
- **DPB** : Droit au paiement de base
- **ADEME** : Agence de l'environnement et de la maîtrise de l'énergie
- **NACE** : Nomenclature statistique des Activités économiques dans la Communauté Européenne
- **CA** : Chiffre d'Affaires
- **R&D** : Recherche et Développement
- **INSEE** : Institut national de la statistique et des études économiques
- **GAEC** : Groupement agricole d'exploitation en commun
- **EARL** : Exploitation agricole à responsabilité limitée
- **CCI :** Chambres des Commerce et d'industrie

Résumé

Durant les mois écoulés, j'ai eu l'opportunité d'effectuer un stage au sein de l'équipe Data d'iNex Circular, une jeune start-up œuvrant dans le développement de l'économie circulaire, à Paris. L'ambition de cette entreprise est de favoriser une production responsable et respectueuse de l'environnement via la valorisation de déchets. Depuis des années, il y a une prise de conscience que le rythme auquel les ressources de la planète sont consommées n'est non seulement pas durable mais a aussi un impact négatif sur la faune et la flore. C'est dans ce contexte qu'émerge l'idée de donner de la valeur aux déchets avec leur réutilisation en partie ou en totalité dans un nouveau processus de production permettant ainsi de faire des économies de matières premières, ou encore en les utilisant pour la production d'engrais ou d'énergie. Bien que la France se soit fixé des objectifs ambitieux sur la question pour les années à venir, le manque d'informations sur les gisements disponibles selon les territoires empêche d'alimenter correctement les infrastructures dédiées à la valorisation et freine aussi leur développement.

L'ambition d'iNex est de pallier ce problème en proposant un outil permettant d'avoir une cartographie des producteurs de déchets et des flux sur l'ensemble de la France et cela pour tout type de matières (biodéchets, carton, plastique, …).

Au sein de l'équipe Data d'iNex, mon rôle était de participer à l'automatisation de la collecte, le traitement et le stockage de la donnée, mais aussi à l'amélioration continue de sa qualité. Pour cela j'ai été amené à identifier des sources de données pertinentes, à implémenter des scrapers mais aussi à gérer une base de données. Ces missions m'ont permis de redécouvrir les langages SQL et noSQL tout en renforçant mes connaissances de python.

Dans le cadre d'un projet pour un client intéressé dans la détection de producteurs de fumiers et lisiers ainsi qu'une estimation de leur quantité, j'ai aussi travaillé sur une méthodologie pour pouvoir prédire la SAU et la taille du cheptel d'une exploitation en France. Les données utilisées pour cette étude sont les chiffres du dernier recensement de l'Agreste (2010) et la liste des bénéficiaires de la PAC. Ce projet a abouti à la mise en place d'un modèle simple qui produit des résultats cohérents. Cependant, le manque de données réelles sur les agriculteurs n'a pas permis d'évaluer l'écart moyen de prédiction.

Grâce à ce stage, j'ai appris : à maitriser de nouvelles technologies (Airflow, Sphinx, Gitlab), l'exigence d'un code structuré et documenté, chercher la solution la plus efficace possible pour répondre à un besoin. Il m'a aussi permis d'acquérir des capacités telles que l'organisation, l'autonomie et l'esprit d'équipe.

Summary

During the past months, I had the opportunity to do an internship with the Data team of iNex Circular, a young start-up working in the development of the circular economy, in Paris. Their ambition is to promote responsible and environmentally friendly production through the recovery of waste. For years, there has been an awareness that the rate at which the planet's resources are consumed is not only unsustainable but also has a negative impact on flora and fauna. It is in this context that the idea of giving value to waste has emerged, by reusing some or all of it in a new production process that saves raw materials. It can also be used to produce fertilisers or energy. Although France has set ambitious objectives on this issue for the coming years, the lack of information on the available deposits according to the territories prevents the infrastructures dedicated to recovery from being properly supplied and also hinders their development.

The ambition of iNex is to solve this problem by offering a tool that provides a mapping of waste producers and flows throughout France for all types of materials (bio-waste, cardboard, plastic, etc.).

Within the iNex Data team, my role was to participate in the automation of data collection, processing and storage, but also in the continuous improvement of its quality. To do this, I had to identify relevant data sources, implement scrapers and also manage a database. These missions allowed me to rediscover the SQL and noSQL languages while reinforcing my knowledge of python.

As part of a project for a client interested in the detection of producers of manure and slurry as well as an estimation of their quantity, I also worked on a methodology to be able to predict the usable agricultural area and the size of the livestock of a farm in France. The data used for this study are the figures from the last census of Agreste (2010) and the list of Common Agricultural Policy beneficiaries. This project has resulted in a simple model that produces consistent results. However, the lack of real data on farmers did not allow to assess the average prediction gap.

Thanks to this internship, I learned: to master new technologies (Airflow, Sphinx, Gitlab), the requirement for a structured and documented code, to look for the most efficient solution to meet a need. It also allowed me to acquire skills such as organization, autonomy and team spirit.

1 Introduction

Du 03/02/2020 au 31/07/2020, j'ai effectué mon stage de fin d'études chez iNex Circular, une jeune start-up évoluant dans le monde de l'économie circulaire. Au sein de cette entreprise dynamique et en pleine croissance, j'ai pu travailler aux côtés de personnes passionnées et engagées en faveur d'une production (de biens et de services) plus responsable et respectueuse de notre environnement via la valorisation des déchets.

Effectuer ce stage chez iNex Circular fut pour moi une opportunité formidable de découvrir ce qu'est l'économie circulaire et l'impact positif qu'elle peut avoir sur les problématiques environnementales et les défis écologiques à résoudre afin de produire de façon durable pour notre planète. En effet, depuis la fin du 18^{e} siecle, notre croissance a reposé majoritairement sur l'utilisation d'énergie fossile non renouvelable et à partir des années 80, on assiste à une prise de conscience collective que ce modèle n'était pas soutenable et qu'il fallait le changer. C'est ainsi qu'en 2002, le concept d'économie circulaire va naitre de la théorie de Michael Braungart et de William McDonough : « Du berceau au berceau ». Aujourd'hui, l'ADEME définit l'économie circulaire comme « comme un système économique d'échange et de production qui, à tous les stades du cycle de vie des produits, vise à augmenter l'efficacité de l'utilisation des ressources et à diminuer l'impact sur l'environnement tout en développant le bien être des individus […] Il s'agit de faire plus et mieux avec moins » [1]. Avec l'émergence du numérique, de nombreux projets mêlent économie circulaire et digital et c'est dans ce contexte qu'est créé iNex Circular en 2014 dans le but d'apporter sa pierre à l'édifice de la révolution environnementale. J'ai aussi pu mesurer à quel point les entreprises du secteur privé étaient en retard sur ces questions même si elles commencent à petits pas leur transformation.

Au-delà d'enrichir mes connaissances, ce stage m'a permis de renforcer mon goût pour l'informatique et la Data, que ce soit le traitement ou l'extraction d'informations, tout en me permettant de donner un sens à mon action, ce qui me tenait à cœur.

Mon stage chez iNex Circular a essentiellement consisté à automatiser la collecte et la mise à jour de données via l'implémentation et l'amélioration de modules sur python mais aussi à développer des algorithmes de prédiction. Ces données vont constituer la base pour prédire les types et les quantités de déchets produits au sein d'un territoire, ce qui intéresse les clients d'iNex.

L'élaboration de ce rapport a pour principale source les différents enseignements tirés de l'exécution de mes différentes missions. Enfin, la documentation sur le fonctionnement et l'organisation de l'entreprise ainsi que les nombreuses discussions que j'ai pu avoir avec mes collègues m'ont permis de donner une cohérence à ce rapport.

Afin de restituer fidèlement les mois passés au sein d'iNex Circular, je vais tout d'abord présenter l'entreprise, puis j'aborderai le contenu et les résultats des différentes missions que j'ai réalisées.

2 iNex Circular : acteur de l'économie circulaire

2.1 Présentation

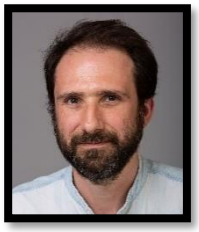
Chaque année 2,5 milliards de tonnes de déchets sont produits en Europe. Ceci représente l'équivalent de 250 000 tours Eiffel tous les ans. On pourrait penser que de nos jours la majorité des déchets sont recyclés, mais ce n'est pas le cas. Seulement 40% des déchets produits en Europe sont recyclés, ce qui équivaut à 150 000 tours Eiffel qui sont chaque année jetés définitivement. 80% d'entre eux sont générés par l'industrie et le BTP (Secteur économique du bâtiment et des travaux publics). Une question se pose alors: Pourquoi si peu de déchets sont-ils recyclés? Cela est principalement lié à un manque de connaissance des acteurs du recyclage, c'est-à-dire les recycleurs et les territoires. Les recycleurs ne savent pas où sourcer des déchets localement. Ce « sourcing » de déchets est primordial lors du lancement d'une nouvelle usine ou pour optimiser les existantes. En effet, actuellement de nombreuses usines tournent à 70% de leur capacité tandis que la mauvaise connaissance des gisements entrave la création des usines. En ce qui concerne les territoires, ils n'ont aucune connaissance des flux de déchets sortants et de ressources entrantes des activités économiques alors qu'ils sont à la manette de la planification du recyclage. L'idée développée avec iNex Circular est donc de favoriser une vision locale du recyclage des déchets, une économie circulaire dans laquelle les déchets des uns sont les matières premières de leurs voisins.

Ainsi est née iNex Circular en 2014 sous l'impulsion de Olivier Gambari et Pascal Hardy. Les déchets représentent un gigantesque potentiel de ressources inexploitées. Afin de faire progresser l'économie circulaire industrielle, ils ont créé une plateforme s'appuyant sur une base de données de substitutions possibles qui ferait « matcher » les entreprises entre elles afin que les déchets des unes deviennent les ressources des autres. Ils ont créé le « Tinder » des déchets.

Associé à une entreprise de développement digital, et grâce à un financement de l'ADEME, la première version de la plateforme voit le jour en 2015 : iNex Analytics. La plateforme est depuis en constante évolution afin de répondre au mieux aux enjeux des clients et de l'économie circulaire, et au jour d'aujourd'hui iNex Circular se dote d'une nouvelle plateforme plus performante à destination des recycleurs sur toute l'Europe : iNex Sourcing, l'outil de détection des gisements de déchets.

2.2 Equipe

L'équipe d'iNex est aujourd'hui constituée de quatre personnes :



Olivier Gambari est le C.E.O. Ingénieur de Telecom ParisTech, il a plus de 15 ans d'expérience dans la création d'outil métier complexe et sur mesure. Il a cofondé iNex Circular en 2014.

Pierre Beuret est le C.O.O. Ingénieur de Grenoble INP, il est le spécialiste en économie circulaire après 5 années passées dans le cabinet Deloitte Développement Durable. Il a rejoint iNex Circular en 2017.

Titouan Poisson est le C.D.O et aussi mon maitre de stage. Ingénieur de centrale Paris, il est spécialiste de la data et son traitement informatique. Il a rejoint iNex Circular en 2019.

Paul Toniolo, Data Analyst. Ingénieur de chimie ParisTech. Il est l'expert métier de la boite et possède des connaissances importantes en ce qui concerne la valorisation des déchets. Il a rejoint iNex Circular en 2019.

iNex Circular compte aussi beaucoup sur ses stagiaires. Ils sont bien souvent en césure ou en stage en fin d'étude et apportent un grand dynamisme à l'entreprise :

- Lisa Luce en tant que commerciale
- Marine Furet en soutient métier

2.3 Services

Analytics

Analytics, première plateforme de iNex Circular est à destination des territoires. C'est une plateforme collaborative permettant de recenser les acteurs et les ressources d'un territoire pour identifier et suivre des synergies de substitution de matières et de massification de services. Son fonctionnement repose sur des profils sectoriels et ceux-ci peuvent être affiliés à chacune des entreprises disponibles dans la base de données d'iNex. Ils permettent ensuite d'affecter aux entreprises des déchets théoriques et des ressources théoriques aux entreprises sans que celles-ci disposent de données réelles. S'ajoute à cela une base de connaissances sur de potentiels liens entre les déchets et les ressources. Cette base s'élève à 1500 substitutions.

La plateforme Analytics peut ainsi simuler les flux de déchets et ressources des entreprises sans interaction avec elles et faire ressortir des « matchs » potentiels entre elles. C'est pour cela que iNex se surnomme « le Tinder des déchets ». Cet outil peut ainsi être utilisé pour proposer des solutions de valorisation locales de leurs déchets grâce à aux « matchs ». Ceci est extrêmement utile pour des utilisateurs comme les collectivités territoriales qui cherchent à créer des interactions bénéfiques au sein de territoire sur le sujet de la valorisation des déchets : mutualisation collective, valorisation locale. iNex Circular propose aussi avec cet outil des méthodologies pour son utilisation et la mise en place de projets au travers d'ateliers et d'études de terrain.

Plusieurs collectivités territoriales ont déjà utilisé cet outil. Son utilisation lors d'une mission dans la Drome a permis d'éviter 1 an d'analyse de territoire, éviter l'émission de 250t/CO2 et la production de 200t de déchets.

Sourcing

Cette plateforme-ci s'adresse plus précisément aux recycleurs comme Veolia. Cet outil est tout nouveau, il a été mis au point en 2019. De façon plus précise que l'outil Analytics, l'outil Sourcing permet à un recycleur, sur des matières données, de recenser les déchets présents dans une zone. Par exemple les biodéchets dans un rayon de 200 km autour de Schönebeck en Allemagne. Ici son fonctionnement repose sur des tags sectoriels qui sont en fait une version plus précise des profils sectoriels. La précision est nécessaire pour les recycleurs quand il s'agit de fournir des usines existantes ou de nouvelles. Chaque entreprise reçoit alors un tag. Ces tags permettent d'affilier des déchets aux entreprises sans avoir de données réelles. La plus-value de cet outil par rapport à l'outil Analytics pour les recycleurs et non seulement la meilleure précision mais aussi l'ajout de caractéristiques sur les déchets. Ceci permet de les qualifier comme par exemple en donnant le potentiel méthanogène des biodéchets ou leur pourcentage de masse sèche. Ces informations sont cruciales pour les recycleurs.

La plateforme peut ainsi simuler tous les flux de déchets sur une zone données avec des informations cruciales dessus. Le recycleur peut ensuite contacter les entreprises par le biais de ses commerciaux

afin de lui proposer sa solution de valorisation. A l'heure actuelle ce n'est pas moins de 2 000 000 de tonnes de biodéchets qui sont détectées sur 6000 entreprises en Allemagne.

2.4 Perspectives

L'ambition de iNex Circular est de devenir le leader européen de la détection de gisements de déchets et besoins en ressources. D'ici 5 ans, 1 000 000 de tonnes de déchets seront recyclées localement grâce à sa technologie. Les fondateurs visent un CA de 10 M€ en 2023 et pour cela, un financement de 1,5M€ est nécessaire. Pour l'instant un premier fond d'investissement a voté unanimement pour financer une partie ce cette somme et d'autres fonds sont en attente.

2.5 Concurrence

Aujourd'hui le nombre de start-ups dans le domaine de l'économie circulaire explose. Ces start-ups se basent souvent sur des technologies de recyclage dans un domaine particulier (Fabrication de cosmétique à partir de déchets de fruits, recyclage du plastiques …). Ces start-ups sont pour iNex plutôt des partenaires potentiels que des concurrents.

Cependant deux start-ups en France proposent des plateformes d'économie circulaire similaire à iNex: l'outil Actif créé par la CCI et Upcyclea.

Par rapport à ces acteurs, la principale force d'iNex est d'avoir une donnée réelle et statistique déjà présente dans l'outil et donc une utilisation immédiate sans passer par une longue phase de collecte de données. Sa deuxième force est de proposer des substitutions automatiques et innovantes grâce à la base de données de matchs. Un outil comme Actif ne propose que des substitutions directes (ex : déchets de bois match avec bois). Le troisième avantage est d'offrir une mise en œuvre réelle des synergies grâce aux données utilisées dans la plate-forme avec l'organisation d'ateliers et une mise en œuvre logistique de ces synergies. Enfin, l'outil d'iNex offre une expérience utilisateur et une ergonomie de hauts niveaux permettant à des non-experts de créer en quelques clics des synergies complexes.

3 Les travaux effectués et les apports du stage

3.1 Cadre – Contexte

L'équipe Data d'iNex est chargée de la collecte et du traitement de la donnée visible sur les deux plateformes que sont Analytics et Sourcing. Elle travaille avec une entreprise de développement digital qui se charge d'élaborer l'interface et les différentes fonctionnalités de l'outil. En d'autres termes, l'équipe Data s'occupe du « contenu » tandis que le « contenant » est géré par l'entreprise partenaire.

Pour mener à bien sa mission, l'équipe Data est amenée à manipuler de la donnée provenant de différentes sources et tout l'enjeu est d'automatiser le processus de collecte mais aussi de réfléchir à un moyen efficace d'agréger, de stocker et de manipuler cette donnée. C'est le rôle du Data Engineer de l'équipe, mon maitre de stage, et par conséquent le mien. L'implémentation de modèles de prédiction est aussi une prérogative de ce poste car il arrive que pour certains secteurs comme l'agriculture, il soit impossible de trouver les variables d'activités (surface, taille du cheptel) nécessaires à l'estimation des quantités de déchets produits et par conséquent il faut mettre en place un modèle pour estimer ces variables.

3.2 Les différentes étapes d'un projet

Tout projet chez iNex débute par une réunion de cadrage avec le client qui permet d'établir un cahier des charges. Cette phase de lancement permet ainsi de définir la zone géographique, les types de déchets recherchés, les tonnages nécessaires, les secteurs d'activités pertinents et les données devant apparaître dans l'outil.

Une fois le cahier des charges établi, une analyse sectorielle est réalisée par l'expert métier et le but est de déterminer les secteurs d'activités produisant les types de déchets recherchés par le recycleur. Sachant que dans la base de données SIRENE est exhaustive, gratuite et que chaque entreprise contient un code NACE (Nomenclature statistique des Activités économiques) qui correspond à son activité, elle constitue la source principale pour l'identification des entreprises productrices de déchets. De ce fait, l'analyse métier permet d'identifier les NACES pertinents au projet mais aussi les seuils d'employés minimums que devraient avoir les entreprises retenues. En effet, le nombre d'employés est un bon indicateur de la taille d'une entreprise et donc c'est une donnée importante pour estimer la quantité de déchets produites.

Le résultat de l'analyse est une liste de NACEs avec un nombre d'employés minimum pour chacun, qui va permette de lancer des algorithmes d'extraction et de transformation de la donnée issue de SIRENE et Kompass. Une fois cette collecte effectuée, l'étape suivante consiste à vérifier que l'activité d'une entreprise dans SIRENE est bien la bonne. En effet, renseigner son activité dans SIRENE reste à la discrétion de chaque entreprise et compte tenu de la complexité de la nomenclature définie, il arrive

que certaines se trompent (exemple : une entreprise qui se déclare comme boulangerie traditionnelle au lieu d'industriel). Pour cela, des algorithmes pour récupérer les informations complémentaires mais aussi de prédiction sont utilisés.

3.3 Les missions

3.3.1 La Formation

Arrivé en février chez inex les premières missions ont eu pour but ma formation pour la suite des opérations. On peut ressortir plusieurs grands axes :

- Familiarisation avec l'entreprise : présentation générale de l'entreprise, son secteur d'activité, les clients avec qui elle travaille et lecture des différents pitch et documents afin d'assimiler au mieux les valeurs de l'entreprises et ses ambitions.

- Formation sur Excel : apprentissage des fonctions de bases de Excel (« recherchev », conversion des données etc.)

- Prise en main de l'outil Sourcing : cela avait pour but de me faire découvrir les fonctionnalités de l'outil et par conséquent mieux comprendre ce qu'on vendait aux clients.

- Ensuite, j'ai eu une présentation plus technique de la part de mon maitre de stage qui m'a présenté les outils mis en place par l'équipe Data, le déroulement type d'un projet mais aussi les différentes tâches qui me seraient confiées. Il y avait deux types de tâches : opérationnel et R&D. L'opérationnel consistait à manipuler une base de données et à lancer des scripts python, alors que pour la R&D, il s'agissait d'améliorer et d'implémenter de nouveaux modules.

3.3.2 Gestion de base de données

En ce qui concerne la gestion de base de données, l'équipe Data a développé une librairie python permettant d'interagir avec une base de données noSQL. Cette dernière est constituée d'entreprises productrices de déchets avec des informations (nom, adresse, code ape, nombre d'employés …) sur elles provenant de différentes sources. L'utilisation du noSQL s'explique par la flexibilité et la rapidité qu'elle offre dans la gestion des données mais aussi par le fait que la plupart des solutions sur le marché sont en open source. La librairie facilite ainsi les accès en lecture et en écriture à la base de données via des fonctions dont certaines permettent l'envoi de requêtes et la présentation des résultats sous une forme structurée ; et d'autres la mise à jour d'informations ou l'insertion de nouvelles entreprises à partir de fichiers Excel. Cette librairie évolue constamment au fur et à mesure que de nouveaux besoins apparaissent, et l'une de mes missions consistait à participer à son amélioration.

Chez iNex, on utilise une base de données mongoDB dont chaque document représente une entreprise et est structuré comme suit :

- Identifiant : Siret de l'entreprise.
- Created : date de l'ajout dans la base de données.
- Sources : liste de documents avec les informations récoltées de différentes sources (Sirene, Kompass, pages jaunes …).
- Updated_sources : date à laquelle ce champ a été modifié pour la dernière fois.
- Projects : les projets auxquels sont associés cette entreprise. En effet, les clients d'iNex n'ont pas forcément les mêmes demandes concernant les seuils de tonnage et le type de déchet. De ce fait, pour chaque projet un identifiant est créé puis affecté à une entreprise lorsqu'elle remplit les critères du projet.
- Data : champ calculé à partir des sources. On définit les informations que l'on veut voir apparaitre dans ce champs et grâce à un script python, une comparaison est effectuée entre les documents du champ sources afin de récupérer l'information issue de la source la plus pertinente.

Les informations des documents du champ 'sources' sont issues de scrapers implémentés dans le but de récolter des informations complémentaires (sur les entreprises) provenant de différents sites web. Ce qui nous amène à la partie suivante qui concerne l'amélioration et la mise en place de scrapers.

3.3.3 Amélioration et Implémentation de nouveaux scrapers

La récupération d'informations complémentaires est très importante chez iNex car elle permet de vérifier que les informations issues de SIRENE pour une entreprise sont correctes. Or, le web est une mine d'or d'informations qu'iNex a décidé d'exploiter grâce aux scrapers. Parmi ces scrapers, on retrouve :

- Scraper google : l'objectif de ce scraper est de récupérer des informations de contact sur les entreprises (site web, téléphone) que l'on trouve généralement dans un encadré lors d'une recherche. Pour cela, on effectue une recherche avec le nom et l'adresse de l'entreprise en question puis on compare le résultat avec les entrées (nom, adresse, code postal). Le résultat est validé lorsque les trois critères sont au-dessus d'un certain seuil.

- Scraper verif : il vise à récupérer ou valider les variables d'activité d'une entreprise (nombre d'employés, chiffre d'affaire) qui permettent d'estimer la quantité de déchets produite.

- Scraper linkedin : permet de récupérer les pages linkedin de potentiels contacts au sein d'une entreprise. Pour cela, une recherche google est effectuée sur le nom de la personne ainsi que celui de l'entreprise dans laquelle elle travaille. Pour être sûr que le(s) résultat(s) renvoyé(s) sont bien des liens linkedin, une astuce consiste à rajouter *'site:www.pagesjaunes.fr'* lors de la recherche. Cela fonctionne comme la définition d'une expression régulière. Afin de valider ou non le résultat, on effectue une comparaison sur le nom de la personne et celui de

l'entreprise pour le premier résultat renvoyé. L'approche consistant à passer par google a été choisi car il est difficile de scraper directement via linkedin qui impose une limitation sur le nombre de recherches qu'un compte peut effectuer.

- Scraper pages jaunes : le but avec ce scraper était de récupérer des informations susceptibles de nous aider à valider l'activité d'une entreprise. En effet, les pages jaunes des petites et moyennes entreprises sont très bien détaillées en ce qui concerne leur activité et les prestations qu'elles offrent. Ainsi, le scraper fonctionne en deux parties : une première qui permet de récupérer les liens pages jaunes en effectuant une recherche google sur le SIRET puis une seconde qui va consister à la récupération en tant que telle de la donnée.

Un scraper est un script ou programme dont le but est d'extraire le contenu de sites web afin de le transformer pour permettre son utilisation dans un autre contexte. Il existe deux librairies sur python qui sont largement utilisés pour le scraping :

- ***Scrapy*** est un framework Python spécialement conçu pour l'extraction de données sur des pages web. Un framework est une application « semi-complète » qui peut être utilisée pour des applications personnalisées. En d'autres termes, le framework Scrapy fournit un ensemble de scripts Python contenant la plupart du code nécessaire à l'utilisation de Python pour le « web scraping ». Il suffit d'ajouter le morceau de code nécessaire pour indiquer à Python les pages à visiter, les informations à extraire de ces dernières et la marche à suivre. Pour cela, il suffit d'implémenter une classe qui définit le ou les sites à « scraper » lors de l'instanciation de cette dernière mais aussi comment extraire des données structurées de ces pages dans la fonction **collect_data** (voir annexe 5).

- ***Selenium*** est un outil pour tester des applications web. Elle permet d'automatiser les différentes actions qu'un utilisateur peut être amené à réaliser comme remplir un champ de caractères ou cliquer sur des boutons. Même s'il n'était pas prédestiné au « web scraping », ***Selenium*** a fini par devenir incontournable au vu des contraintes qu'il permettait de résoudre. En effet la plupart des sites web se dotent d'outils afin de se prémunir contre le web scraping. Ces outils peuvent consister entre autres à identifier et à bloquer les requêtes issues de scripts de n'importe quel langage informatique, à limiter le nombre de requêtes sur un intervalle de temps défini ou tout simplement à utiliser un framework **Javascript** dont les modules ne seront pas chargés lors d'une requête issue d'un script. Les deux premières peuvent être résolues avec ***scrapy*** car il est possible de masquer la nature d'un script en le faisant passer pour un navigateur web auprès du serveur lors d'une requête. En ce qui concerne les limitations de requêtes, définir des temps d'arrêt lors de l'implémentation du script peut éviter le blocage par le serveur qui ainsi ne répondrait plus aux requêtes issues de notre adresse IP. Cependant, il est impossible pour ***scrapy*** de contourner le troisième car les modules JavaScript sont destinés à être exécutés par un navigateur web. D'où l'intérêt d'utiliser la librairie ***Selenium*** qui a recours à un pilote informatique permettant à python d'interagir avec un navigateur web. Cela permet ainsi de faire des requêtes via un navigateur web qui se chargera d'exécuter les modules JavaScript et donc de disposer de tout le contenu d'une page web (voir annexe 6).

Pour effectuer un scraping efficace et performant en termes de temps, il est nécessaire de mettre en place une infrastructure permettant de contourner les éventuels blocages d'adresse IP. En effet, la plupart des sites web imposent une limite de requêtes aux utilisateurs pour éviter une

surcharge de leurs serveurs. Ainsi, l'équipe data d'iNex possède un serveur avec un pool d'adresse IP permettant de contourner ce type de problèmes.

3.3.4 Le projet Agriculture

Avec la loi relative à la transition énergétique pour la croissance verte votéé en 2015, la France vise une augmentation de sa production d'énergies renouvelables de 70% en 2028 faisant ainsi passer sa part de 15% à 30% de l'énergie totale consommée en 2030. Pour atteindre cet objectif, l'une des ressources renouvelables à plus fort potentiel de développement est le biogaz (mélange gazeux composé principalement de méthane). En effet, les déchets organiques constituent la matière première à la production de biogaz ; or les collectivités, l'industrie et l'agriculture produisent des quantités énormes de déchets dont organiques qui ne demandent qu'à être valorisées. Parmi tous ces secteurs, le biogaz agricole est celui qui connait la plus forte progression depuis l'an 2000 et pour maintenir cette progression dans les années à venir, différents défis doivent être relevés, notamment celui de l'approvisionnement en matières premières. Actuellement, les méthaniseurs agricoles fonctionnent principalement avec des substrats agricoles liquides tels que les lisiers de porcs ou de bovins auxquels sont adjoints des co-substrats à forts potentiels méthanogènes tels que des déchets agro-alimentaires (graisses, huiles, issues de céréales, etc.). Ce modèle de développement, quoique non épuisé, paraît insuffisant pour répondre aux objectifs fixés. Afin d'assurer le développement des unités de méthanisation agricoles, d'autres substrats doivent être mobilisés tels que les résidus de culture et les fumiers. Or, l'évaluation fine des gisements sur le territoire national constitue un enjeu majeur dans la mise en place de nouveaux méthaniseurs du fait des questions logistiques, du procédé et des analyses technico-économiques et environnementales. L'ambition d'iNex est de proposer une solution qui permette de faire gagner du temps sur ces questions et donc accélérer l'ouverture de nouveaux méthaniseurs sur le territoire.

C'est dans ce cadre que s'inscrit le projet agriculture dont l'objectif final était d'estimer pour chaque exploitation Agricole, sa surface Agricole utile ainsi que le nombre de têtes pour les élevages.

3.3.4.1 Données

Pour ce projet, il a fallu d'abord récupérer une liste de toutes les exploitations agricoles en France. Je me suis appuyé sur la base de données SIRENE qui, grâce au code NACE qui définit l'activité d'une entreprise, permet de les distinguer de manière exhaustive. Une fois ce listing obtenu, il était necessaire de trouver des informations sur les agriculteurs permettant d'estimer leur SAU. Cette étape s'est avérée complexe du fait de la faible disponibilité de données à l'échelle individuelle concernant les exploitations agricoles. Pour contourner ce problème, nous nous sommes appuyés sur la politique Agricole commune (PAC) qui encourage les agriculteurs à fournir un grand nombre d'informations sur leurs exploitations. En effet, Pour obtenir leurs subventions, les agriculteurs déclarent certaines informations comme la taille, la nature de l'exploitation, la part de surface d'intérêt écologique ou la nature de l'élevage. S'il n'est pas possible d'accéder directement aux

informations qu'ils donnent, il est en revanche possible de connaître la nature des subventions versées aux différents exploitants.

Grâce à la fusion de la liste d'agriculteurs issue de SIRENE et celle issue de la PAC, on obtient le tableau 'exploitations' dont chaque ligne contient les informations suivantes : code SIRET (unique pour chaque exploitation), adresse, code postal, DPB (paiements découplés à la surface ; il s'agit d'une subvention versée dans le cadre de la PAC), code NACE (indique le secteur d'activité). Les autres colonnes correspondent à d'autres subventions.

Le tableau « exploitations par type d'élevage »

Ce tableau recense le nombre d'exploitations selon le type d'élevage au sein d'un département donné en 2010. Ainsi, chaque ligne correspond à un département et chaque colonne représente le nombre d'exploitations pour un type d'élevage spécifique. Ces données sont issues du dernier recensement agricole de 2010 [3].

Le tableau « SAU par forme juridique »

Ce tableau, comme son nom l'indique, recense la SAU totale des exploitations selon leur forme juridique. Chaque ligne correspond à un département et chaque colonne représente la SAU occupée par les exploitations partageant la même forme juridique. Ces données sont issues du dernier recensement agricole de 2010 [3].

Le tableau « cheptels par département »

Ce tableau est aussi issu du dernier recensement agricole de 2010 [3]. Il recense pour chaque commune, le nombre total de bêtes par catégorie se trouvant dans des exploitations de ce département (par exemple, le nombre de vaches laitières dans un département donné).

Chaque ligne correspond à un département, et les colonnes représentent le type de bête.

Ce tableau a été nettoyé pour être utilisable : en particulier, il avait initialement des cases avec la valeur "s" au lieu d'une valeur numérique dû au secret statistique : certaines valeurs pourraient permettre de trouver des informations sur une exploitation précise. Pour souci de simplicité, ces valeurs sont considérées nulles : la présence d'un "s" pour une commune et une catégorie de bêtes signifie qu'il y a très peu d'exploitations dans cette commune qui possède cette catégorie de bêtes. Il s'agit alors la plupart du temps d'une petite exploitation, donc la valeur réelle de cette quantité est faible.

3.3.4.2 Calcul de la SAU pour chaque exploitation

Ce calcul s'est fait grâce au DPB des exploitations, donnés par le tableau "exploitations". Le DPB est une subvention de la PAC directement reliée à la taille d'une exploitation. La surface peut être retrouvée en divisant le DPB par le DPB/surface de l'exploitation. Ce ratio était initialement dépendant de chaque zone géographique, mais la PAC vise à faire converger les DPB/surface de toutes les exploitations vers le DPB/surface moyen national. En résulte que tous les DPB/surface sont proches de cette valeur moyenne, qui sera celle utilisée par la suite pour toutes les exploitations.

Les résultats sont obtenus grâce à l'équation (1).

$$SAU = \frac{DPB}{DPB_{surfacique}} \qquad (1)$$

Pour les exploitations ne bénéficiant pas du DPB, un autre mode de calcul a été utilisé pour trouver leur surface : on leur a affecté la SAU moyenne par forme juridique au sein du département dans lequel elle se trouve. Ce choix a été fait car il permet de prendre en compte les disparités de surface selon les formes juridiques des exploitations. En effet selon l'INSEE, « Les formes sociétaires concernent près de 7 exploitations sur 10 parmi les grandes exploitations, voire plus de 8 sur 10 parmi les très grandes. Les exploitations agricoles à responsabilité limitée (EARL) sont les formes sociétaires privilégiées, notamment pour les grandes exploitations. Les groupements agricoles d'exploitation en commun (Gaec) sont les autres types de sociétés les plus répandues."

3.3.4.3 Calcul du nombre de têtes pour les élevages

Ce calcul se fait en établissant une proportionnalité entre la surface d'une exploitation et son nombre de bêtes, en fonction du type de bête possédées par l'exploitation. Cela se traduit par l'équation suivante :

$$N_i = \frac{S_i}{\sum_{i=1}^{N} S_i} * N_T * f$$

- N_T : nombre total de têtes selon le type d'élevage dans une commune ou un département donné. En priorité, le calcul est effectué à l'échelle communale (Cette valeur est issue du tableau « cheptels par commune »). Cependant, cette valeur n'est pas forcément disponible pour toutes les communes et dans ce cas on passe au niveau départemental (valeur issue du tableau « cheptels par département »).

- N : nombre d'exploitations dans notre jeu de données selon le type d'élevage au sein d'une commune ou d'un département donné.

- S_i : surface de l'exploitation i

- N_i : taille du cheptel de l'exploitation i

- f : facteur correctif. L'application de ce facteur s'explique par le fait que le dernier recensement date de 2010 et donc le nombre de têtes à répartir N_T aussi. Or, cette valeur a évolué entre temps et il a fallu l'estimer car non disponible (prochain recensement en 2020). Pour ce faire, une simple règle de trois nous a permis d'aboutir à l'équation (2) avec N_{2010} le nombre d'exploitations selon le type d'élevage au sein d'un département donné en 2010 :

$$f = \frac{N}{N_{2010}} \qquad (2)$$

3.3.4.4 Résultats

Dans un premier temps, la méthode décrite précedemment a été implémentée sur quelques départements : Ain, Allier, Côte-d'or, Doubs, Haute-Marne, Nièvre, Jura, Saône-et-Loire, Haute-Saône, Vosges, Yonne, Territoire de Belfort. Ce choix a été dicté par les besoins d'un client intéressé pour une étude de gisement sur cette zone concernant les déchets agricoles.

Calcul de la SAU de chaque exploitation

Ce tableau (page suivante) contient les métriques concernant l'attribution de SAU via notre modèle en comparaison avec les données issues du dernier recensement de l'Agreste en 2010. Chaque ligne correspond à une combinaison (département, forme juridique) avec les informations suivantes:

- NB_exploit_2010: nombre d'exploitations recensés en 2010
- NB_exploit_dataset: nombre d'exploitations dans notre jeu de données
- SAU_2010: SAU recensée en 2010
- SAU_dataset: SAU attribuée via le modèle
- SAU_moyenne_2010: SAU Moyenne des exploitations de la combinaison en 2010
- SAU_moyenne_dataset: SAU Moyenne attribuée via le modèle

L'analyse des résultats d'un point de vue macro révèle le fait que l'on estime une SAU totale supérieure à celle du dernier recensement dans la zone (5812845 contre 3685378). Or la plupart des études suggèrent une légère baisse de la SAU en France. Ce phénomène est observé un peu partout en France et on peut supposer que notre zone n'y échappe pas. En regardant les chiffres dans le détail, on remarque que notre jeu de données contient plus d'exploitations qu'en 2010 (65894 contre 45128) alors que d'après l'INSEE, le nombre d'exploitations agricoles est en baisse depuis 2010. Cette différence s'explique par l'utilisation de SIRENE pour constituer notre liste d'exploitations agricoles. En effet, on sélectionne les entreprises avec un code NACE en rapport avec une activité agricole ; mais dans certains cas l'activité renseignée n'est pas forcément la bonne et un autre problème réside dans le fait que SIRENE ne détecte pas toujours les entreprises qui disparaissent. Cependant, nous n'avons pas réussi à trouver une autre source de données nous permettant de distinguer et d'éliminer ce

surplus d'exploitations de notre jeu de données. Malgré cela, la SAU moyenne estimée grâce à notre approche est de 122.2 ha, ce qui est relativement proche de celle du recensement de 2010 (115.5 ha).

Il aurait été intéressant d'effectuer une comparaison entre les prédictions et les valeurs réels de SAU pour un ensemble d'exploitations, ce qui nous aurait permis d'évaluer l'écart moyen et par conséquent la pertinence de l'approche utilisée. Malheureusement, nous ne possédions pas encore ces données réelles.

département	forme_juridique	NB_exploit_2010	NB_exploit_dataset	SAU_2010	SAU_dataset	SAU_moyenne_2010	SAU_moyenne_dataset
71	['EI']	5411	9998	237368	519666.1065	43.86767695	51.97700605
71	['EARL']	1204	2173	117378	233113.5954	97.49003322	107.2773104
71	['GAEC']	713	1826	143291	358646.4117	200.9691445	196.4109593
71	['PM']	361	761	19416	54251.57279	53.78393352	71.28984598
25	['EI']	2100	2668	83610	110517.443	39.81428571	41.42332947
25	['EARL']	492	405	42445	30732.1938	86.2703252	75.88195999
25	['GAEC']	648	1051	90020	127479.572	138.9197531	121.2935985
25	['PM']	104	178	3535	7046.418936	33.99038462	39.58662324
21	['EI']	2597	3983	143973	254761.3179	55.43819792	63.9621687
21	['EARL']	1225	1394	157950	217545.1455	128.9387755	156.0582105
21	['GAEC']	467	483	125654	132595.7822	269.0663812	274.5254291
21	['PM']	604	889	30102	64577.48101	49.83774834	72.64058607
01	['EI']	2980	4512	117079	196460.4357	39.28825503	43.54176323
01	['EARL']	436	578	43919	68577.86171	100.7316514	118.6468196
01	['GAEC']	466	542	74973	86955.08836	160.8862661	160.4337424
01	['PM']	212	306	11431	19508.12264	53.91981132	63.75203479
88	['EI']	2266	2648	70984	93886.54889	31.32568402	35.45564535

88	['EARL']	329	374	43272	48645.39602	131.5258359	130.0679038
88	['GAEC']	485	677	98655	125882.9215	203.4123711	185.9422769
88	['PM']	118	138	8700	12248.18327	73.72881356	88.75495122
52	['EI']	1235	1624	80124	118920.7537	64.87773279	73.22706511
52	['EARL']	458	545	85313	107682.4422	186.2729258	197.5824627
52	['GAEC']	416	440	121394	127595.8508	291.8125	289.9905701
52	['PM']	136	304	18546	52730.9681	136.3676471	173.4571319
39	['EI']	2165	2575	67230	94109.943	31.05311778	36.54755068
39	['EARL']	457	464	47973	48290.68866	104.9737418	104.07476
39	['GAEC']	419	613	67613	92394.83154	161.3675418	150.7256632
39	['PM']	114	135	5530	7765.104259	48.50877193	57.51929081
70	['EI']	1903	2572	85253	131035.0716	44.79926432	50.94676188
70	['EARL']	396	506	54006	72229.90903	136.3787879	142.7468558
70	['GAEC']	392	554	86293	112914.0182	220.1352041	203.8159173
70	['PM']	101	146	8340	14151.56375	82.57425743	96.92851881
58	['EI']	2449	4290	181319	338843.2769	74.03797468	78.98444683
58	['EARL']	529	1076	89883	186199.8408	169.9111531	173.0481792
58	['GAEC']	258	688	69175	168608.6116	268.120155	245.0706564

58	['PM']	240	550	29825	83690.89538	124.2708333	152.1652643
89	['EI']	2534	3185	160140	244340.5508	63.19652723	76.71602851
89	['EARL']	991	1171	137323	197050.2867	138.5701312	168.2752235
89	['GAEC']	300	232	69914	58513.49977	233.0466667	252.2133611
89	['PM']	449	734	49898	110963.0417	111.1314031	151.1758062
90	['EI']	353	416	9115	12143.69157	25.82152975	29.19156627
90	['EARL']	38	54	3982	5673.027936	104.7894737	105.0560729
90	['GAEC']	39	49	6767	8852.255645	173.5128205	180.6582785
90	['PM']	15	17	351	760.3413913	23.4	44.72596419
03	['EI']	3988	5501	230895	340306.4181	57.89744233	61.86264645
03	['EARL']	724	709	102793	101138.9782	141.9792818	142.6501808
03	['GAEC']	584	834	125707	171687.5366	215.2517123	205.8603556
03	['PM']	227	326	26921	41154.41769	118.5947137	126.2405451

Figure 1 : Tableau avec les métriques sur le calcul de la SAU

Prédiction de la taille des cheptels

dep	moyenne_ tetes_2010	moyenne_ tetes_modele	NB_exploit_2010	NB_exploit_dataset
71	41.91803279	44.54608822	549	369
25	41.32677693	38.29934902	2237	2228
88	48.40368381	53.0182285	1303	941
39	44.75810692	41.71790166	1141	1037
01	48.82886598	46.97584742	970	744
52	58.4757953	58.36504566	723	157
21	49.61980831	52.04772793	313	39
70	48.2032032	54.32452888	999	652
58	34.33333333	41.47470516	105	57
89	50.32802548	61.44405941	314	66
90	44.79661017	65.497657	118	57
03	33.02153846	36.00712919	325	84

Figure 2 : Tableau avec les métriques sur le calcul de la taille de cheptels

Ce tableau contient les métriques concernant la prediction du nombre de têtes (uniquement pour les élevages de vaches laitières) via notre modèle en comparaison avec les données issues du dernier recensement de l'Agreste en 2010. Chaque ligne correspond à un département avec les informations suivantes:

- NB_exploit_2010: nombre d'exploitations recensées en 2010 au sein du département.
- NB_exploit_dataset: nombre d'exploitations au sein du département dans notre jeu de données.
- Moyenne_tetes_2010: nombre de vaches laitières moyen dans le département en 2010.
- Moyenne_tetes _modele: moyenne des nombres de vaches prédites dans le département.

Nous remarquons du point de vue macro que nous avons moins d'élevages de vaches laitières dans notre dataset par rapport au recensement de 2010 (18048 contre 36252), soit une baisse de 50%. Même si on assiste à une baisse du nombre d'exploitations agricoles depuis 2010 selon l'INSEE, ce chiffre ne nous semble pas cohérent. Ce qui nous amène au problème expliqué un peu plus haut concernant l'utilisation de la base de données SIRENE et du NACE. En effet, pour obtenir le listing de tous les élevages de vaches laitières, on sélectionne toutes les entreprises de SIRENE avec le NACE '01.41Z'. Se faisant, nous avons dû passer à coté d'exploitations possédant des vaches laitières mais dont l'activité principale est tout autre (ce qui a pour conséquence que leur NACE est différent de '01.41Z'). Le manque de données réelles (prochain recensement en 2020) ne nous permet pas de quantifier le nombre d'exploitations dans ce cas.

Le manque de données réelles concernant la taille du cheptel pour un ensemble d'exploitations sur la zone ne nous a pas permis aussi d'évaluer les écarts de prédiction et donc de la pertinence du mode de calcul. Cependant, la moyenne des predictions est relativement proche de celle du recensement de 2010 (399 contre 364), ce qui nous permet d'affimer que dans l'ensemble, les prédictions ne sont pas aberrantes.

Conclusion et améliorations possibles

L'implémentation dans une zone de 12 départements nous a permis de voir que sur un échantillon important, l'approche utilisée pour estimer la SAU et la taille du cheptel produit des résultats qui semblent cohérents. Cependant, nous n'avons pas pu confirmer ce sentiment grâce à une comparaison des prédictions avec des valeurs réelles que nous ne possédions pas. Globalement, le manque de données a constitué un obstacle majeur lors de cette mission.

Malgré des résultats encourageants, nombre d'améliorations peuvent être apportées:

- Concernant le jeu de données, il faudrait pouvoir verifier que l'activité (NACE) renseignée est la bonne mais aussi distinguer précisément les différentes activités d'une exploitation (exemple: une exploitation qui cultive des céréales tout en possédant des vaches laitières).

- Pour le calcul des SAU, on part du principe que le DPB/surface moyen national est déja harmonisé. Ce qui n'est pas encore le cas car le processus est en cours et donc il serait intéressant d'avoir le DPB/surface moyen pour chaque région. De plus, pour les exploitations sans DPB dans notre dataset, on aurait pu gagner en précision en prenant en compte l'activité en plus du statut juridique car le type de culture par exemple peut influer sur la taille.

- Pour la prédiction de la taille des cheptels, il serait pertinent d'intégrer le caractère intensif ou non d'une exploitation. De plus, sachant que le modèle se base sur une répartition d'un nombre de bêtes, avoir des chiffres réels s'avèrent pertinents (ce qui sera le cas en 2021 vu que le recensement a lieu en 2020).

4 Conclusion

Ce stage a été très enrichissant pour moi car il m'a permis non seulement de découvrir le concept d'économie circulaire mais aussi de travailler aux côtés de personnes passionnées et engagées en faveur d'une production (de biens et de services) plus responsable et respectueuse de notre environnement via la valorisation des déchets. J'ai particulièrement apprécié leur apporter ma contribution dans leur processus d'automatisation de la collecte, du traitement et le stockage de la donnée. Lors de ces mois passés chez iNex, j'ai aussi aimé la confiance qui m'a été témoignée qui se traduisait par une grande liberté dans mes choix. Cette liberté m'a permis de prendre conscience que même si une bonne planification des tâches reste nécessaire, la capacité d'adaptation est aussi importante car les choses ne se passent pas toujours comme prévues. Les difficultés rencontrées tout au long de ma mission m'ont permis d'approfondir mes connaissances en informatique et plus précisément l'implémentation de scrapers mais aussi d'en acquérir des nouvelles comme l'utilisation du système Linux et de Git, la gestion de flux de travail avec Airflow. J'ai aussi eu l'opportunité de mettre en pratique mes connaissances en termes d'exploitation de données dans l'optique de créer un modèle de prédiction.

Avec la prise de conscience générale que le mode de production actuel n'est pas durable, il devient nécessaire de le changer et la valorisation des déchets peut y contribuer énormément. De plus, notre monde d'aujourd'hui est dominé par les nouvelles technologies et le Big Data qui représente une mine d'informations très importante. Ainsi, l'envie chez iNex d'exploiter ces ressources, de les mettre au service de ce changement représente une initiative à encourager et je suis fier d'y avoir participé.

Malgré le fait qu'iNex soit sur la bonne voie, l'utilisation du Machine Learning n'est pas très développée. Or, l'utilisation du NLP peut apporter de la valeur ajoutée concernant la vérification des activités des entreprises qui est une phase importante pour pouvoir estimer les flux de déchets. Pour cela, ils devront mettre en place une infrastructure leur permettant de pouvoir traiter beaucoup de données de façon rapide et efficace mais aussi d'investir du temps sur des projets de recherche.

Fort de cette expérience qui a renforcé mon goût pour la programmation, le traitement et l'exploitation de données, j'envisage pour la suite une carrière en tant que Data Scientist/Engineer afin de travailler sur des sujets passionnants et stimulants intellectuellement.

Références

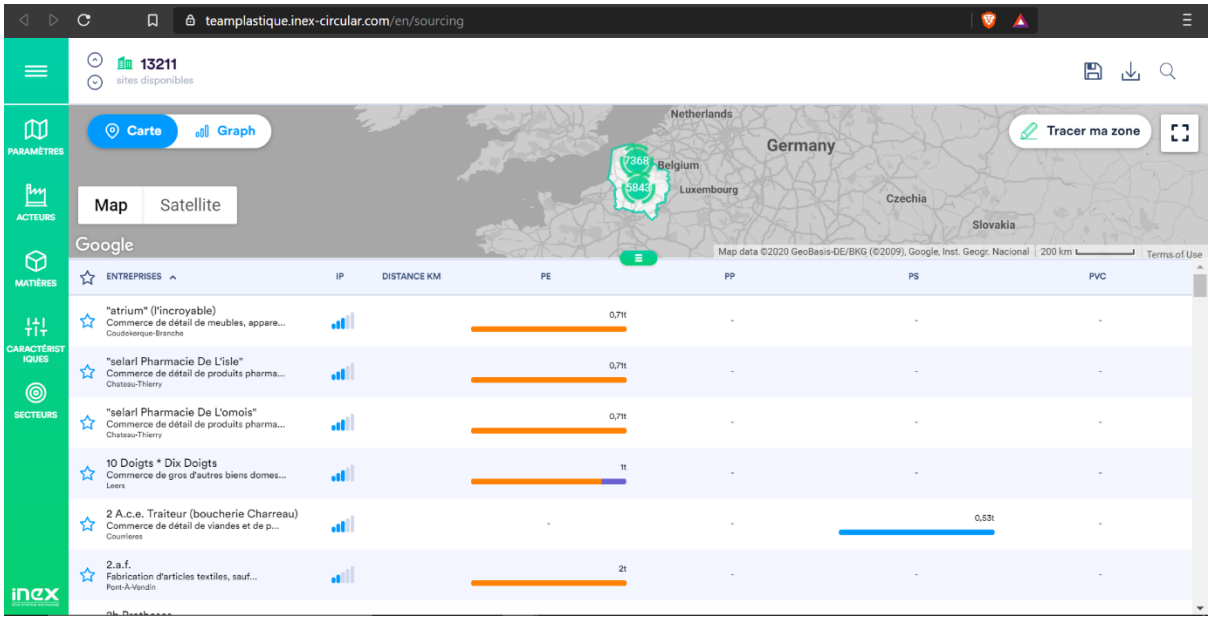
DEGUEURCE, Axelle ; CAPDEVILLE, Jacques ; PERROT, Christophe ; BIOTEAU, Thierry ; MARTINEZ, José ; PEU, Pascal, Fumiers de bovins, une ressource à fort potentiel pour la filière de méthanisation en France ?, *Revue Science Eaux & Territoires*, article hors-série, 9 p., 22/02/2016, disponible en ligne sur <URL : http://www.set-revue.fr/fumiers-de-bovins-une-ressource-fort-potentiel-pour-la-filiere-de-methanisation-en-france> (consulté le 15/07/2020)

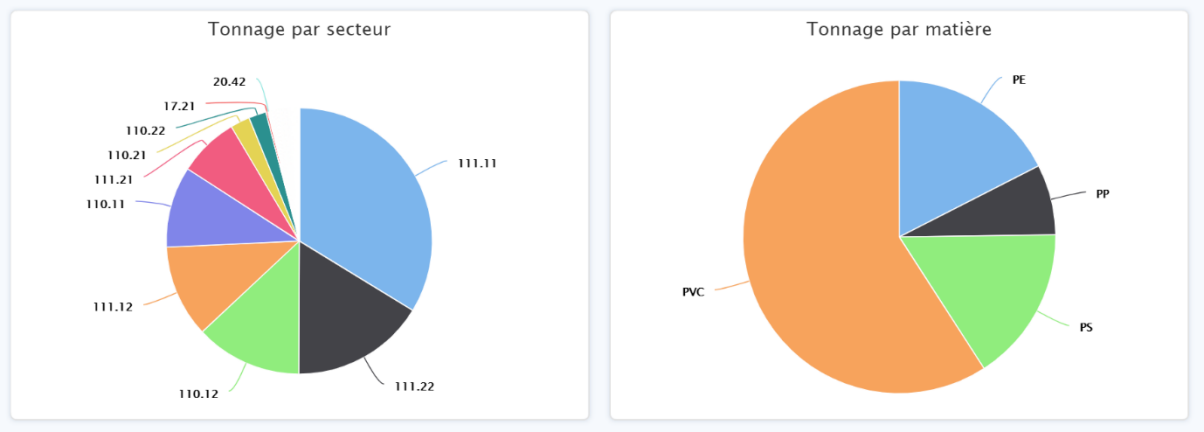
[1] https://www.ademe.fr/sites/default/files/assets/documents/fiche-technique-economie-circulaire-oct-2014.pdf (consulté le 18/07/2020)

[2] https://www.insee.fr/fr/statistiques/3676823?sommaire=3696937 (consulté le 18/07/2020)

[3] https://agreste.agriculture.gouv.fr/agreste-web/ (consulté le 18/07/2020)

Annexe 1: Vues issues de l'outil Sourcing





Annexe 2: Extrait du tableau « exploitations par type d'élevage »

département	Elevages (total hors apiculture)	vaches laitières	autres bovins	Equidés	Caprins	Ovins	porcins	volailles	ovins et caprins
971	4614	0	3390	50	966	94	1250	435	1060
972	1824	15	946	68	311	392	223	219	703
973	496	21	143	49	53	57	115	280	110
974	3752	140	622	102	1082	100	580	2799	1182
77	589	87	133	231	22	92	15	163	114
78	223	17	61	117	12	35	11	40	47
91	67	11	8	29	6	10	0	20	16
93	S	0	0	0	0	0	0	0	0
94	3	0	0	0	0	0	0	0	0
95	122	13	38	57	7	22	4	25	29
18	2130	180	1181	478	230	525	65	695	755
28	1213	162	364	194	25	234	56	613	259
36	3162	289	1666	475	306	862	178	1226	1168
37	1852	354	473	306	209	266	90	660	475
41	1135	223	318	185	87	194	79	366	281
45	1026	207	242	230	64	139	58	487	203
21	2242	313	1393	383	46	558	65	312	604
25	3034	2237	463	770	114	347	175	706	461
39	2139	1141	479	547	110	330	98	541	440
58	2763	105	2113	508	101	756	106	349	857
70	2265	999	901	490	73	538	100	564	611
71	5561	549	3805	942	578	1297	208	1221	1875

Annexe 3: Extrait du tableau « SAU par forme juridique »

	SAU_total	SAU_exploitation_individuelle	SAU_GAEC	SAU_autres	SAU_EARL
FRA	27087794	11651616	5490571	2461894	7483712
971	31401	25269	67	4896	1169
972	24982	12542	54	10235	2151
973	25345	20942	0	4140	67
974	42814	35883	603	5066	1261
75	0	0	0	0	0
77	335860	119703	15777	61932	138448
78	89134	40403	5148	13349	30235
91	84144	35986	6127	9787	32243
92	13	4	0	9	0
93	887	254	0	511	104
94	998	247	0	377	64
95	57800	13275	1691	15086	27747
18	432333	157860	49920	102574	121980
28	450574	201747	29977	60242	158609
36	452691	195754	51459	76082	129396
37	332175	156167	36969	30189	108850
41	288333	133830	31964	28420	94120
45	355267	140620	44142	34161	136346
21	457678	143973	125654	30102	157950
25	219610	83610	90020	3535	42445
39	188346	67230	67613	5530	47973
58	370203	181319	69175	29825	89883
70	233892	85253	86293	8340	54006
71	517453	237368	143291	19416	117378
89	417276	160140	69914	49898	137323
90	20215	9115	6767	351	3982
14	380878	186364	75428	25710	93376
27	376981	128767	44772	30047	173395
50	427119	192119	117059	15547	102394
61	397461	167316	87356	24174	118615
76	397416	139396	97321	42795	117905
02	493330	152312	46551	103496	190971
59	354347	163933	73258	20096	97060
60	368691	83726	33614	65238	186114
62	463513	192036	99012	32566	139899
80	465287	180020	67727	58891	158648
08	302043	97806	59999	26853	117384
10	374639	83824	44262	69458	177095
51	554703	133008	46447	105811	269437
52	305377	80124	121394	18546	85313

Annexe 4: Extrait du tableau « cheptels par département »

département	Total Bovins	Total Vaches	Vaches laitieres	Vaches allaitantes	Chevres	Brebis nourrices	Brebis laitieres	Total Porcins	Poulets de chair et coq
01 - Ain	184 974	70 073	47 364	22 709	5 800	19 272	1 396	141 212	1 181 605
02 - Aisne	204 634	74 522	43 128	31 394	274	25 006	35	62 463	1 147 794
03 - Allier	545 480	209 030	10 732	198 298	7 987	140 782	412	93 007	1 812 300
04 - Alpes-de-Haute-Provence	13 188	5 346	817	4 529	7 002	143 899	1 786	3 190	19 707
05 - Hautes-Alpes	31 064	11 143	4 858	6 285	5 002	174 478	3 012	9 785	20 376
06 - Alpes-Maritimes	1 872	910	428	482	3 822	45 566	2 067	117	1 965
07 - Ardèche	52 554	26 035	12 899	13 136	26 035	65 431	837	6 688	759 051
08 - Ardennes	266 303	86 380	40 613	45 767	232	29 122	s	31 738	764 914
09 - Ariège	86 688	39 692	7 141	32 551	4 931	64 495	2 944	2 487	53 620
10 - Aube	53 671	22 137	10 589	11 548	214	15 731	nd	48 526	932 151
11 - Aude	25 856	12 443	1 944	10 499	2 627	39 402	4 841	18 595	652 935
12 - Aveyron	482 310	221 001	53 665	167 336	48 641	127 120	561 563	185 871	163 301
13 - Bouches-du-Rhône	18 344	5 236	356	4 880	4 383	134 603	141	3 475	s
14 - Calvados	403 419	155 401	103 797	51 604	1 379	20 323	137	71 486	522 333
15 - Cantal	491 175	228 080	76 925	151 155	3 362	33 340	61	42 909	75 543
16 - Charente	159 159	67 759	21 582	46 177	26 160	60 046		118 276	168 947
17 - Charente-Maritime	108 313	47 856	21 762	26 094	19 982	8 856	s	7 493	295 234
18 - Cher	179 412	71 532	7 072	64 460	25 460	40 303	s	56 002	953 212
19 - Corrèze	316 089	149 835	9 434	140 401	3 557	48 173	517	60 091	192 438
21 - Côte-d'Or	224 555	87 082	15 531	71 551	686	41 272	nd	19 034	435 355
22 - Côtes-d'Armor	533 754	222 975	185 612	37 363	1 402	13 867	128	2 685 617	13 049 512
23 - Creuse	446 730	184 379	8 923	175 456	9 166	67 602	562	48 833	174 372
24 - Dordogne	254 848	113 513	27 044	86 469	22 310	50 039	3 870	82 957	1 591 050
25 - Doubs	237 502	98 838	92 448	6 390	1 166	7 752	s	51 401	79 379
26 - Drôme	32 372	11 259	4 644	6 615	27 613	64 253	1 144	26 102	3 302 105

Annexe 5 : Script python s'appuyant sur le framework scrapy pour le scraping des Pages Jaunes

```python
from scrapy import Spider
import scrapy_splash
# import config as cfg
import datetime
# from postal.parser import parse_address
from time import sleep


class PagesJaunesSpider(Spider):
    name = 'scrap_pagesjaunes'

    def __init__(self, *args, **kwargs):
        super(PagesJaunesSpider, self).__init__(*args, **kwargs)
        self.allowed_domains = ['pagesjaunes.fr']
        self.start_urls = kwargs.get('url_list')

    def start_requests(self):
        for i in range(len(self.start_urls)):
            yield scrapy_splash.SplashRequest(
                url=self.start_urls[i],
                callback=self.collect_data,
                dont_filter=True,
                endpoint='render.html',
                args={'wait': 4},
                meta={'url': self.start_urls[i]})


    def collect_data(self, response):
        """Collect the data"""
        url = response.url
        res = dict()
        res['url'] = url
        # res['code_ape'] = self.nace
        res['scrap_date'] = datetime.datetime.now()
        # res['id_source'] = "pages_jaunes"

        data = dict()
        data['url'] = url
        data['nom_etablissement'] = response.xpath('//*[@id="teaser-header"]'
                                                   '/div[1]/div[1]/div/div[1]'
                                                   '/h1/text()').get()
        data['activite'] = response.xpath('//*[@id="teaser-header"]/div[1]'
                                          '/div[1]/div/div[2]/div'
                                          '/span/text()').get()
        tel = response.xpath('//*[@id="teaser-footer"]/div/div'
                             '/div[1]/div/span/span[2]'
                             '/text()').get()
        if tel != '\n':
            data['telephone'] = tel

```

```
49
50          data['adresse_complete'] = ""
51          bloc_adresse = response.xpath('//*[@id="teaser-footer"]/div/div'
52                                        '/div[2]/a[1]/span')
53          if len(bloc_adresse) == 3:
54              adresse_num = bloc_adresse[0].xpath('text()').get()
55              if adresse_num is not None:
56                  data['adresse_num'] = adresse_num
57              adresse_voie = bloc_adresse[0].xpath('span/text()').get()
58              data['adresse_voie'] = adresse_voie
59              adresse_cp = bloc_adresse[1].xpath('text()').get().replace(',', '')
60              data['adresse_cp'] = adresse_cp
61              adresse_ville = bloc_adresse[2].xpath('text()').get()
62              data['adresse_ville'] = adresse_ville
63              try:
64                  data['adresse_complete'] = adresse_num + adresse_voie
65                  + adresse_cp + adresse_ville
66              except TypeError:
67                  if adresse_num is None:
68                      data['adresse_complete'] = adresse_voie + adresse_cp\
69                          + adresse_ville
70                  else:
71                      pass
72
73          print(data['adresse_complete'])
74
75          prestation_bloc = response.xpath('//*[@id="zone-info"]'
76                                           '/div[@class="zone-produits-presta'
77                                           '-services-marques fd-bloc"]'
78                                           '/div[1]/ul/li')
79          if(len(prestation_bloc) == 0):
80              prestation_bloc = response.xpath('//*[@id="zone-info"]'
81                                               '/div[@class="zone-produits-'
82                                               'presta-services-marques fd-bloc"'
83                                               ']/div/div[1]/ul/li')
84          prestations = []
85          if len(prestation_bloc) != 0:
86              for bloc in prestation_bloc:
87                  prestations.append(bloc.xpath('span/text()').get())
88          data['prestations'] = prestations
89
90          activite_bloc = response.xpath('//*[@id="zoneMultiactivite"]'
91                                         '/div/ul/li')
92          activites = []
93          if len(activite_bloc) != 0:
94              for bloc in activite_bloc:
95                  info = bloc.xpath('span/text()').get()
96                  if info is None:
97                      info = bloc.xpath('a/span/text()').get()
98                  activites.append(info)
99          data['activites'] = activites
100
101         try:
102             web = response.xpath('//*[@id="teaser-footer"]/div/div'
103                                  '/div[4]/a/span[2]/text()').get()\
104                                      .strip()
105             if 'http' not in web:
106                 data['web'] = 'http://' + web
107             else:
108                 data['web'] = web
109         except AttributeError:
110             pass
111
112         boutton_insee = response.xpath('//*[@id="ancre-nav"]'
113                                        '/button[@title="Aller à la partie '
114                                        'Infos INSEE"]')
115         boutton_insee_exist = False
116         if(len(boutton_insee) == 1):
117             boutton_insee_exist = True
```

```
        if boutton_insee_exist is True:
            data['siret'] = response.css('li.row.siret').xpath('span/text()')\
                .get().strip()
            res['identifiant'] = data['siret']

            employe_str = response.css('li.row.effectif_entreprise')\
                .xpath('span/text()').get()
            if r'à' in employe_str:
                data['tranche_employe_basse'] = employe_str.split()[0]
                data['tranche_employe_haute'] = employe_str.split()[2]
                data['employes'] = round(
                                        (float(data['tranche_employe_basse']) +
                                         float(data['tranche_employe_haute'])
                                        ) / 2)
            elif r'salarié' in employe_str:
                data['employes'] = int(employe_str.split()[0].strip())

            data['type_etablissement'] = response.css('li.row.siege')\
                .xpath('span/text()').get()
            try:
                data['adresse_siege'] = response.css('li.row.adresse_siege')\
                    .xpath('span/text()').get()
            except AttributeError:
                pass

            forme_juridique = response.css('li.row.forme_juridique')\
                .xpath('span/text()').get()
            if forme_juridique is not None:
                data['forme_juridique'] = forme_juridique

            data['dirigeants'] = response.css('li.row.dirigeants')\
                .xpath('div/span/text()').extract()
```

```
            ca = response.css('li.row.chiffre_affaire')\
                .xpath('span/text()').get()
            if ca is not None:
                data['ca'] = response.css('li.row.chiffre_affaire')\
                    .xpath('span/text()').get()
                if r'à' in data['ca']:
                    data['tranche_ca_basse'] = \
                        data['ca'].split()[0].replace(',', '.')\
                                                .replace(' .', '.')
                    data['tranche_ca_haute'] = \
                        data['ca'].split()[2].replace(',', '.')

                    data['ca'] = round((float(data['tranche_ca_basse']) +
                                        float(data['tranche_ca_haute'])
                                        ) / 2)

        # we suppose that if the title is missing it's
        # because page jaunes blocked us
        if data['nom_etablissement'] is None:
            sleep(15)
            print('nom_eta_none')

        else:
            for key, value in data.items():
                print(key, value)
                if isinstance(value, str):
                    data[key] = value.strip().replace('\n', '')
            res['data'] = data

        yield res
```

Annexe 6 : Script python s'appuyant sur la librairie Selenium pour la première partie du scraping des Pages Jaunes

```python
import pandas as pd
from bs4 import BeautifulSoup
import time
import random
from bson import ObjectId
import config as cfg
from fuzzywuzzy import fuzz
from selenium import webdriver
from selenium.common.exceptions import TimeoutException
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC
from selenium.webdriver.common.by import By
from selenium.webdriver.common.keys import Keys
import db_credentials as db_cred

from inex_tools.mongo.entreprise_class import EntrepriseConnection
# from connect_to_db import ConnectToMariaDB


class ScrapPagesJaunes():

    def __init__(self, project, selenium_instance,
                 nace_list=None, force_scrap=False):

        self._project = project
        self._nace_list = nace_list
        self._force_scrap = force_scrap

        self._options = webdriver.ChromeOptions()
        self._options.add_argument('--no-sandbox')
        # Launch webdriver
        self._driver = webdriver.Remote(
            command_executor=selenium_instance,
            desired_capabilities=self._options.to_capabilities())

```

```python
    def _get_companies(self):
        """collect contacts to search"""
        self._mongo = EntrepriseConnection()
        self._mongo.connect_to_db(
            host=db_cred.mongodb['host'],
            port=db_cred.mongodb['port'],
            db=db_cred.mongodb['database'],
            collection=db_cred.mongodb['collection'])

        self._companies = self._mongo.get_companies_by_project(
                self._project,
                nace_codes=self._nace_list,
                df=True)

        # Retrieving people who was previously scraped
        if self._force_scrap is False:
            companies_with_pj_source =\
                self._mongo._collection.find({'projects':self._project,
                                              'sources.id_source':'pages_jaunes'})
            datas = []
            for company in companies_with_pj_source:
                try:
                    data_temp = dict(company['data'])
                except KeyError:
                    # Company has no data field
                    data_temp = {}
                data_temp['identifiant'] = company['identifiant']
                datas.append(data_temp)
            was_scraped = pd.DataFrame(datas)

            # Deleting those people
            if len(was_scraped) != 0:
                self._companies = self._companies[
                    ~self._companies['identifiant'].isin(
                        was_scraped['identifiant'].tolist())].reset_index(drop=True)

```

```
        print(self._companies)

    def _wait_and_select_elem_by_xpath(self, xpath, waiting_time, text=False):
        """ Select an element by its xpath and return text"""
        try:
            wait = WebDriverWait(self._driver, waiting_time)
            wait.until(EC.presence_of_element_located((By.XPATH, xpath)))
            if text:
                return self._driver.find_element_by_xpath(xpath).text
            else:
                return self._driver.find_element_by_xpath(xpath)
        except TimeoutException:
            print('balise not found')
        except AttributeError:
            print('no result')

    def _search_on_google(self, search):
        """ Search on google and look if there is a connection failure"""
        # Going on the main page
        self._driver.get("https://www.google.com")
        # Selection the xpath of the search bar
        elem = self._wait_and_select_elem_by_xpath(
            cfg.XPATH_SEARCH_BAR,
            60,
            text=False)
        # Type search in the bar
        print(search.encode('ascii', 'ignore'))
        elem.send_keys(search)
        # Launch search
        elem.send_keys(Keys.RETURN)

    def _collect_data(self, index):

        res = self._companies.loc[index,:].to_dict()
        data = dict()
        data['recherche_google'] = \
            self._companies.loc[index, 'recherche_google']
        try:
            wait = WebDriverWait(self._driver, 10)
            wait.until(EC.presence_of_element_located((By.XPATH, '//*[@id="rso"]')))

            # we turn the page content into bs4 object
            response = BeautifulSoup(self._driver.page_source, 'lxml')
            result_list = response.find('div',attrs={'id':'rso'}) \
                .find_all('div', attrs={'class':'g'})
            data['link_first_result'] = result_list[0].find('a')['href']
            data['name_first_result'] = result_list[0].find('h3').text
            data['description_first_result'] = \
                result_list[0].find('span',attrs={'class':'st'}).text

            """name_contact_linkedin = \
                    res['name_first_result'].split(' - ')[0].strip()
            if fuzz.WRatio(name_contact_linkedin, res['contact']) >= 90:
                res['matched'] = True"""

            print(data)

        except TimeoutException:
            print('no result')
            data["commentaire"] = "timeout (no result)"

        except AttributeError:
            data["commentaire"] = "pas de resultat"

        self._mongo.append_source_to_company(res['identifiant'], 'pages_jaunes',
                                             data)
```

```python
    def _search_pj_link(self, i):
        """search contact of index i"""

        print(str(i + 1) + " / " + str(len(self._companies)))
        siret = self._companies.loc[i, 'identifiant']

        try:
            time.sleep(random.uniform(4, 8))
            to_search = 'site:www.pagesjaunes.fr {}'.format(siret)
            self._companies.loc[i, 'recherche_google'] = to_search
            self._search_on_google(to_search)

        except:
            time.sleep(15)
            self._search_on_google(to_search)

        self._collect_data(i)

    def _test_detection(self):
        """Test if google has detected the scraper"""
        text = "Our systems have detected unusual traffic from your computer"
        if text in self._driver.page_source:
            return True
        else:
            return False

    def run_scrap(self):
        print("""start searching pages jaunes links...""")
        self._get_companies()
        self._companies = self._companies.reset_index(drop=True)
        for i in range(len(self._companies)):
            self._search_pj_link(i)
            # Test if google has detected the bot and wait a bit if so
            if self._test_detection():
                print("Google detected us, waiting few minutes ...")
                time.sleep(300)
```