



Rapport du Stage Ingénieur

XIN YAN

OPTION : DATASIM

3 avril 2023 - 18 août 2023

Tuteur(s) université :
SAÏD MOUSSAOUI
said.moussaoui@ec-nantes.fr

Tuteur(s) entreprise :
CÉLINE COLIN
CAline.Colin@ge.com



Remerciement

Je tiens à exprimer ma sincère gratitude envers vous tous. Durant cette période de stage, j'ai acquis de précieuses compétences et souvenirs, et cela n'aurait pas été possible sans votre soutien et votre aide.

Tout d'abord, je tiens à remercier les enseignants de datasim pour leur enseignement diligent. Merci pour les cours soigneusement planifiés qui m'ont non seulement enrichi en connaissances, mais m'ont également permis de mener à bien mon stage plus facilement. Les enseignants de l'École Centrale de Nantes m'ont également profondément impressionné. Votre professionnalisme, votre dévouement et votre patience ont été d'une grande valeur pour moi. Je tiens également à remercier le programme de formation de l'École Centrale de Nantes qui m'a donné l'opportunité d'effectuer un stage d'ingénieur en France avant mon retour en Chine, me permettant ainsi de vivre l'expérience du travail en France.

Ensuite, je souhaite exprimer ma gratitude envers ma superviseure qui a découvert mon CV et m'a offert cette opportunité de stage. Plus important encore, pendant le stage, elle m'a prodigué des conseils patients qui m'ont permis d'acquérir de précieuses connaissances et compétences.

De plus, je tiens à remercier mes formidables collègues. Vous êtes tous exceptionnels ! Nos moments de pause café quotidiens et nos efforts conjoints pour résoudre les problèmes sont des moments que je chérirai toujours. Votre collaboration et votre soutien ont été une motivation importante pour moi pendant mon stage.

Enfin, je tiens à exprimer ma gratitude pour l'expérience de vie précieuse que j'ai acquise en étudiant en France. Bien que j'aie rencontré des défis, ces expériences m'ont également permis de grandir et de devenir plus fort et plus confiant.

Encore une fois, merci à vous tous pour votre soutien et vos encouragements. Je continuerai à travailler dur et à mettre en pratique ce que j'ai appris dans mes futurs travaux et études. J'espère avoir l'opportunité de collaborer à nouveau avec vous à l'avenir.

Table des matières

1 Introduction	4
1.1 Présentation de l'entreprise	4
1.2 Contexte du sujet	5
2 Connaissances préalables du projet	6
2.1 La vie d'une pièce	6
2.2 Voyages des pièces	7
2.3 Problème de FOA	8
2.3.1 Définition de FOA	8
2.3.2 Etat actuel de FOA chez GEHC	9
2.3.3 Pourquoi c'est intéressant	10
3 Base de données utilisées	11
3.1 FBI	11
3.2 Service Suite	11
3.3 GSPO	12
3.4 One Model Explorer	12
4 Méthodologie du projet	13
4.1 Choix des critères	13
4.2 Construction de la base de données	14
4.2.1 Collection des données de Service Suite	14
4.2.2 Tracing des pièces	14
4.3 Calcule des critères	16
4.4 Encoder les données	16
4.5 Choisir l'algorithme utilisé	17
4.5.1 Méthodes non-supervisées	18
4.5.2 Méthodes supervisées(Forêt Aléatoire)	21
4.6 Analyse de haut niveau	23
4.6.1 Tous features	23
4.6.2 Feature Engineering	25
4.7 Analyse par référence	26
4.8 Analyse par catégorie/composant	29
4.9 Analyse thématique	30
4.9.1 Analyse sur les Centres de Réparation	30
4.9.2 Analyse des pièces primées	31
5 Outils créés	33
5.1 Module de collection	33
5.2 Module d'analyse	34
5.3 Module de prédiction	35
5.4 Documenter les modules	35
6 Conclusions	39
6.1 Conclusions générales	39
6.2 Conclusions individuelles :	41

7 Conseils pour la future	48
7.1 Contraints des données	48
7.2 Comment retirer les contraints	49
7.2.1 RFID Technologie	49
7.2.2 QR code ou bar code	51
8 Résume du stage	52
9 Annexe	53
9.1 Abréviations utilisées	53

1 Introduction

Ce rapport expose les travaux réalisés dans le cadre d'un stage d'ingénieur au sein de GE Healthcare. Le sujet de stage consiste à identifier les causes racines potentielles des défaillances de pièces dès leur arrivée (Failure On Arrival (FOA)) en utilisant des méthodes non-supervisé et supervisé de Machine Learning, notamment les clusterings et les Forêts Aléatoires. En effectuant cette analyse, on pourrait améliorer les processus de traitement des pièces.

Ce rapport présente en détail la méthodologie utilisée pour mettre en œuvre ces techniques de Machine Learning, ainsi que les résultats obtenus. De plus, nous discuterons des implications pratiques de ces résultats et proposerons des recommandations pour améliorer la qualité et la fiabilité des produits.

1.1 Présentation de l'entreprise

GE Healthcare est un des leaders mondiaux dans les ventes et services des systèmes médicaux, fournissant des solutions innovantes pour améliorer les soins de santé et la qualité de vie des patients. Présent en France depuis 1987, il emploie aujourd'hui 2800 collaborateurs, dont 600 ingénieurs R&D dans son site d'excellence internationale à Buc dans les Yvelines. GE Healthcare a noué de solides partenariats de recherche avec des PME et des centres de recherche français pour développer des technologies et des services médicaux révolutionnaires qui ouvrent une nouvelle ère pour les soins apportés aux patients.



FIGURE 2 – Porte d'entreprise à Buc

1.2 Contexte du sujet

La fiabilité des produits revêt une importance cruciale dans le secteur médical, car toute défaillance potentielle peut entraîner des conséquences graves pour les patients et les professionnels de la santé. Compte tenu de la criticité de ses produits (appareils médicaux) General Electric Healthcare (GEHC) propose un service de maintenance pour ses clients. L'objectif principal de ce service est d'assurer un temps maximum de disponibilité du produit tout en veillant à réduire l'ensemble des coûts associés à l'entretien des machines grâce une grande maîtrise de l'ensemble des processus de maintenance.

En particulier, GE Healthcare fournit des pièces de remplacement pour le dépannage des équipements (pour un montant d'environ 1 milliard de dollars par an). Parmi ces pièces, les pièces réparées et recyclées (représentant 10% du coût pour 30% du volume d'activité) sont clefs pour une maintenance efficace de la base installée. Ces pièces permettent d'assurer une supply chain sur le long terme avec une équation économique et environnementale intéressante. Les pièces réparées sont en général les plus chères et les plus complexes. La chaîne logistique inverse rajoute à cette complexité. L'amélioration de la qualité de ces pièces est un axe constant d'amélioration. Par conséquent, il est essentiel de comprendre les causes racines des défaillances de pièces dès leur arrivée afin de prendre des mesures préventives pour améliorer la qualité et la fiabilité des produits.

Ainsi, mon principal objectif lors de ce stage était d'exploiter les données fournies par l'entreprise, comprenant des informations détaillées sur les pièces, les transactions et les réparations, etc., et de faire parler ces données.

2 Connaissances préalables du projet

Dans cette section, j'aborde les connaissances préalables nécessaires pour comprendre le contexte du projet. Tout d'abord, je présente ce qui se passe avant l'installation d'une pièce, puis j'explique de manière plus détaillée le problème de la FOA.

2.1 La vie d'une pièce

Un système médical est composé de plusieurs pièces différentes. La plupart de ces pièces sont fabriquées par nos fournisseurs, mais aussi une partie est fabriquée en interne. Il y a plusieurs paramètres liés avec ces pièces, par exemple la référence de la pièce, qui indique le type et la version de pièce. Normalement la référence est une série de caractères, parfois suivie d'un tiret '-X' ('X' signifie une ou plusieurs chiffres ou lettres). '-chiffre' indique une version différente de cette référence, '-lettre' indique l'expérience de vie de la pièce. Dans ce cas là, '-R' signifie que cette pièce est réparable et il a été réparé au moins une fois. '-H' signifie que cette pièce est Harvest, c'est à dire elle est récupérée d'une autre machine mise au rebut. Généralement pour une pièce chère ou une pièce complexe à fabriquer, la vie est longue.

Une fois une pièce est fabriqué par une fournisseur ou nous-même, il sera envoyé aux clients pour la première installation. Malheureusement, on n'a aucune information de fournisseur à la première installation de la pièce. Après envoyé chez un client, cette pièce sera installé par un Feild Engineer (FE) et commencer sa carrière dès maintenant. La période où la pièce travaille est appelée le temps de opération(Operating Time).

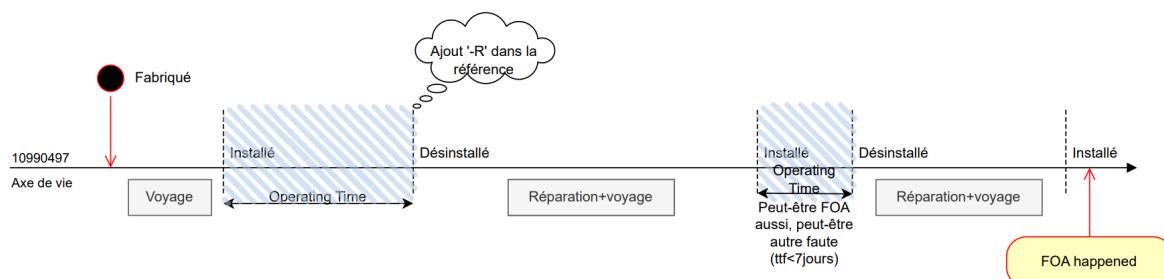


FIGURE 3 – La vie d'une pièce

Quand elle est tombé en panne une fois, un FE a été appelé pour réparer la machine. La pièce sera été démonté et été renvoyé à l'un des centre de réparation pour réparer si ça vaut le coût de réparer(Comme la figure montre). A ce moment là, un '-R' est ajouté dans la référence.

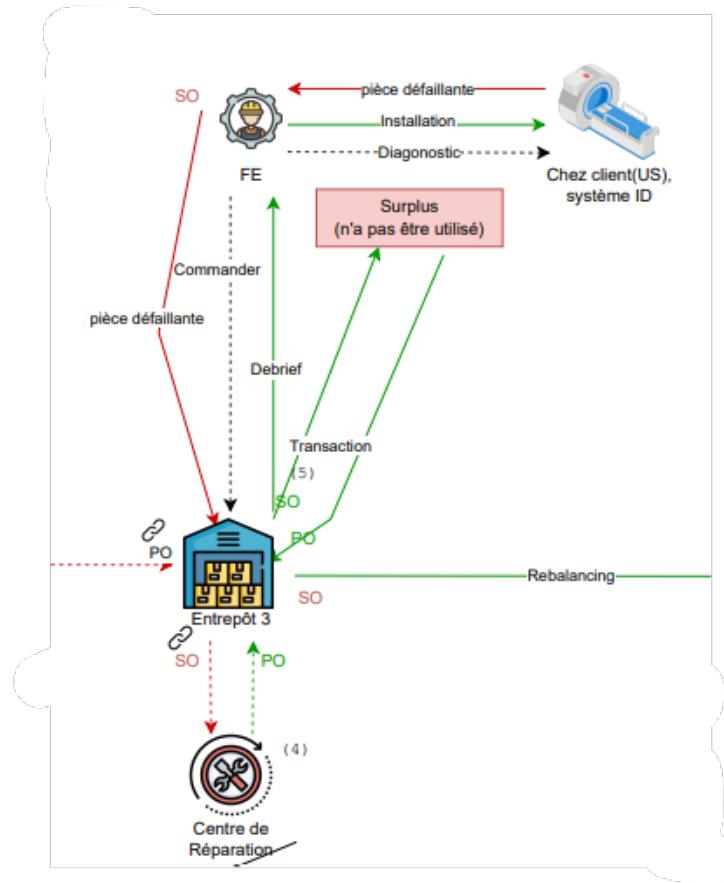


FIGURE 4 – Quand une pièce est tombé en panne..

Après la réparation, lorsque cette pièce est commandée par un autre client, elle est à nouveau envoyée et recommence son travail jusqu'à ce qu'elle casse à nouveau.

2.2 Voyages des pièces

Dans cette sous-section, nous allons explorer les transactions des pièces lorsqu'elles ne sont pas en service, c'est-à-dire ce qui s'est passé après leur sortie du centre de réparation. Comprendre ces transactions est essentiel pour évaluer l'historique des pièces et sélectionner nos caractéristiques pour l'analyse. Nous examinerons les différentes opérations telles que le prélèvement, le transfert, le retour et l'échange de pièces, qui contribuent à la gestion efficace des stocks. La figure 5 montre les flux et les interactions entre les pièces tout au long de leur cycle de vie.

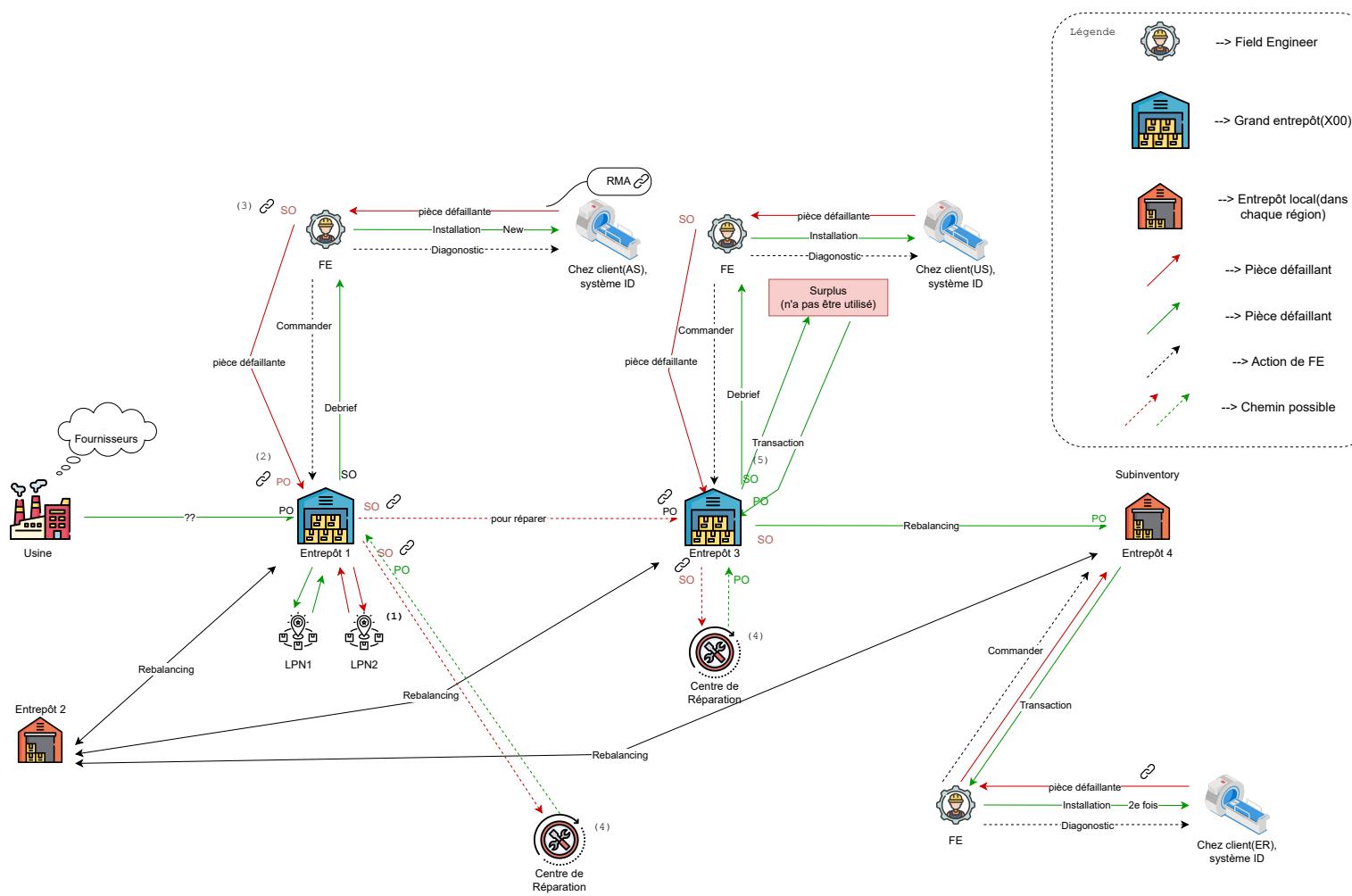


FIGURE 5 – Transaction des pièces

2.3 Problème de FOA

Je tiens à rappeler que ma mission consiste à identifier les causes des défaillances dès leur arrivée (FOA) et à proposer des démarches d'amélioration visant à réduire ce taux, avec pour objectif final la réduction des coûts.

2.3.1 Définition de FOA

Mais c'est quoi exactement la pièce FOA ? FOA est l'abréviation de 'Failure On Arrival', ça se réfère à une pièce qui présente une défaillance ou un dysfonctionnement dès son arrivée.

2.3.2 Etat actuel de FOA chez GEHC

En mettant l'accent sur l'analyse des données européennes de l'année 2022, nous avons constaté que certaines références de pièces présentaient des taux de FOA assez élevés, atteignant parfois 60 %, 80 %, voire même 100 %. Cette tendance peut s'expliquer par le fait que ces pièces sont peu consommées, ce qui signifie que même un petit nombre de pièces FOA peut entraîner un ratio FOA / Non-FOA élevé. Cependant, il convient de noter que ces types de pièces représentent une petite proportion du total.

En général, nous avons observé que les pièces ayant un taux de FOA inférieur à 30 % représentaient 98,1 % de l'ensemble de toutes les pièces consommées. De plus, les pièces affichant un taux FOA inférieur à 20 % représentaient 94,6 % du total, tandis que celles avec un taux FOA inférieur à 10 % représentaient 86,2 % du total. Ces chiffres démontrent que la majorité des pièces présentent des taux FOA relativement faibles.

De plus, nous avons constaté une variation significative des probabilités de FOA en fonction des régions (voir figure 6), ce qui est lié à la référence de la pièce et est plus évident pour certaines références, comme le montre la figure suivante (voir figure 7). Cela suggère que des facteurs géographiques ou des variations dans les processus de manipulation des pièces peuvent influencer les taux de défaillance à l'arrivée. Cette constatation souligne l'importance d'une analyse plus approfondie et d'une compréhension des spécificités régionales pour développer des stratégies efficaces de prévention et de gestion des pièces FOA.

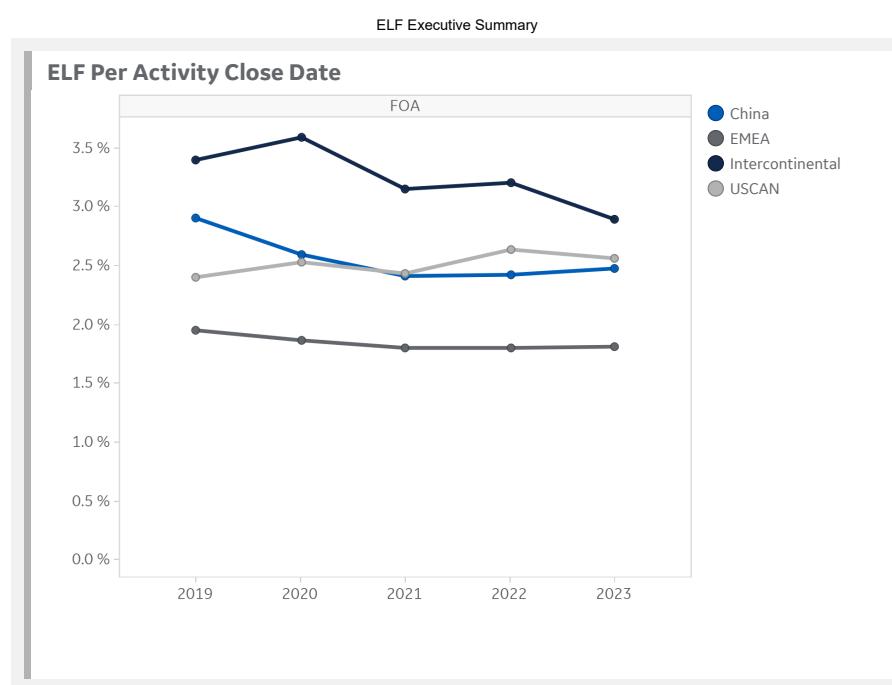


FIGURE 6 – Taux de FOA moyenne

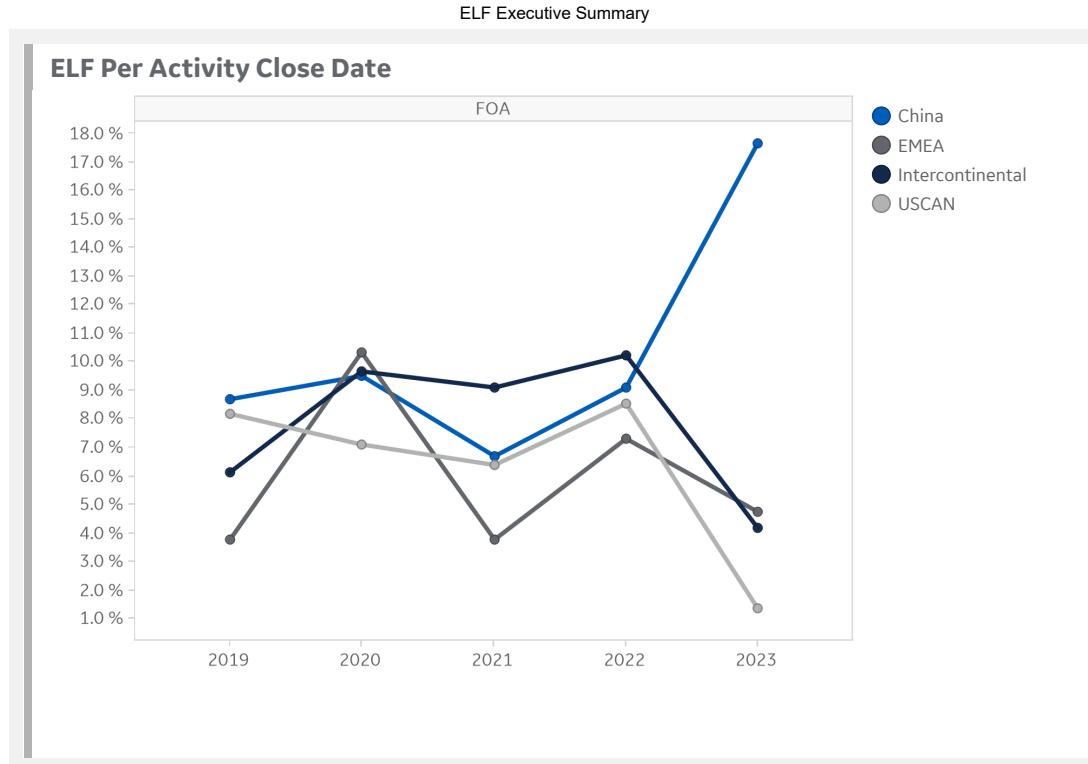


FIGURE 7 – Taux de FOA pour ref. 2351573

Ces résultats mettent en évidence la nécessité de poursuivre nos investigations pour identifier les causes racines des défaillances des pièces dès leur arrivée. Une meilleure compréhension de ces causes permettra de mettre en place des mesures préventives ciblées et d'améliorer ainsi la qualité globale des produits.

2.3.3 Pourquoi c'est intéressant

Pourquoi c'est un sujet intéressant à creuser ? Parce que les FOAs sont coûteuses. En effet, les défaillances dès l'arrivée des pièces engendrent des coûts importants à différents niveaux.

Tout d'abord, il y a le coût lié à la main-d'œuvre. Les ingénieurs mobilisés pour réparer ou remplacer les pièces défaillantes entraînent des dépenses supplémentaires en termes de salaires, de temps et de ressources dédiées à cette tâche.

De plus, les FOAs entraînent des retards dans les processus de soins médicaux, ce qui peut avoir un impact négatif sur les patients. Le temps d'attente accru, les rendez-vous annulés ou reportés peuvent non seulement causer des désagréments pour les patients, mais également compromettre leur santé et leur bien-être.

Les coûts de transport sont également à prendre en compte. Lorsqu'une pièce FOA est identifiée, il peut être nécessaire de la renvoyer au fournisseur pour réparation ou remplacement. Cela implique des frais d'expédition supplémentaires, ainsi que des émissions de gaz à effet de serre dues aux transports, contribuant ainsi à l'empreinte environnementale de l'entreprise.

Enfin, les pièces FOA ont un impact direct sur les coûts de fabrication. Les ressources utilisées pour produire ces pièces défectueuses sont gaspillées, ce qui entraîne une perte financière pour l'entreprise. De plus, cela peut également perturber la chaîne d'approvisionnement et la planification de la production, affectant ainsi l'efficacité globale de l'entreprise.

En conclusion, l'analyse des pièces FOA revêt une grande importance en raison des coûts élevés qu'elles engendrent. Comprendre les causes racines de ces défaillances dès l'arrivée permettra de mettre en œuvre des mesures préventives ciblées, réduisant ainsi les dépenses liées à la main-d'œuvre, le temps d'attente des patients, le transport et la fabrication. Cette approche contribuera à améliorer la rentabilité, la satisfaction des patients, la durabilité environnementale et la qualité globale des produits de l'entreprise.

3 Base de données utilisées

Dans cette partie, je vais présenter les bases de données chez GEHC qui serve à la suite à mon projet.

3.1 FBI

La base de données du FBI contient plusieurs tableaux différents. Nous avons utilisé les tableaux "transactions" et "Repair Tracker (eRT)" pour rechercher l'historique d'une pièce. Le tableau "transactions" contient toutes les transactions des pièces depuis 2013, tandis que le tableau "eRT" contient les enregistrements des pièces réparées. En construisant l'historique d'une pièce, nous utilisons les numéros de commande d'achat (PO - Purchase Order Number) et de commande de vente (SO - Sale Order Number) à partir de ces deux tableaux.

De plus, nous avons également utilisé le tableau "XCarrier" dans cette base de données pour trouver les dimensions et le poids des différentes pièces. De plus, il existe un tableau "DimItem" pour rechercher les changements de catégorie des pièces. À l'aide de ces tableaux, nous avons construit une référence qui répertorie toutes les pièces, leurs pièces parentes, leurs pièces parentes ultimes, leurs catégories, leurs poids et leurs dimensions. Cette référence peut être mise à jour automatiquement en utilisant les sources de données disponibles.

Tous ces tableaux sont accessibles via des requêtes SQL, mais ils peuvent également être consultés à partir d'un site web en créant une connexion.

3.2 Service Suite

SERVICE SUITE est une plate-forme d'analyse de données de services qui consolide et normalise les métriques stratégiques créées à l'aide d'ensembles de données certifiés pour Toutes les régions GEHC et toutes les modalités. Il est soutenu par une équipe d'experts techniques et opérationnels dédiés pour apporter les meilleures données et visualisations. Voici un exemple de visualisation dans Service Suite. Au part de ça, il y a aussi des données brutes qui peut télécharger sous forme .csv pour analyser.

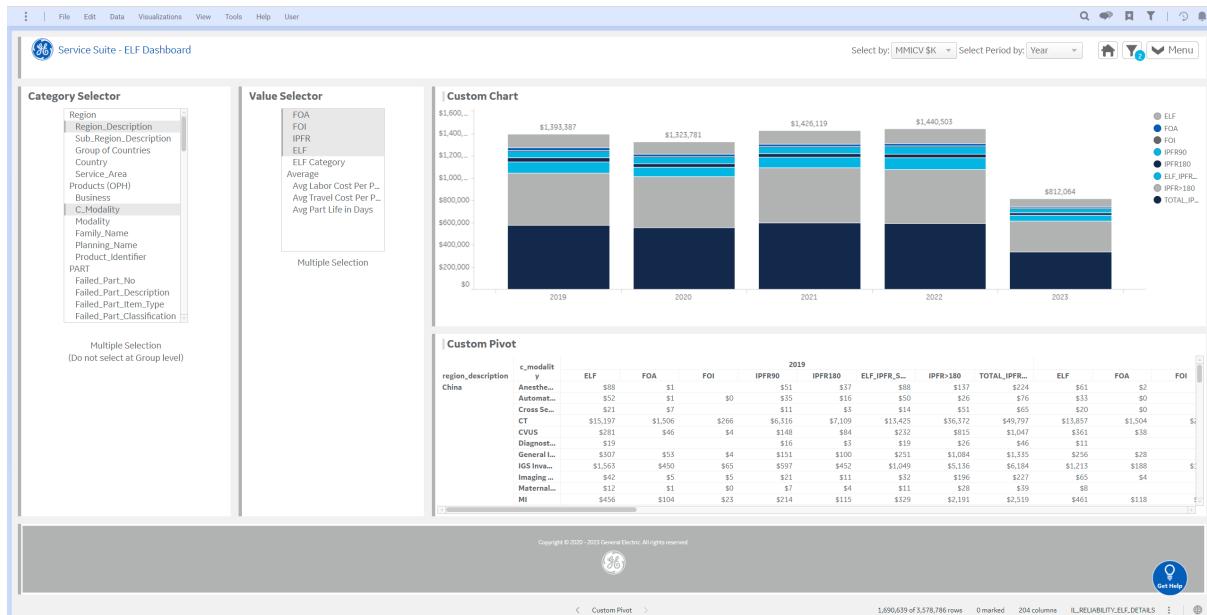


FIGURE 8 – Un exemple de visualisation dans Service Suite

Pendant mon stage, j'ai principalement utilisé ELF. Le tableau ELF dans Service Suite est un outil qui nous permet d'obtenir toutes les informations sur les pièces cassées depuis 2019, y compris leur type, leur date, leur fournisseur, leur catégorie, etc. Il se trouve sous la grande colonne de la Fiabilité. Les données sont divisées en cinq grandes parties en fonction des types de défaillances : FOA (Failure on Arrival), FOI (Failure on Installation), IPFR90 (Tombe en panne sous 90 jours), IPFR180 (Tombe en panne sous 180 jours) et IPFR>180 (Tombe en panne après 180 jours).

3.3 GSPO

J'utilise cette base de données pour rechercher les poids des pièces et créer un tableau de référence qui sera utile pour la suite de mon stage. Comme tous les autres outils de données, elle contient plusieurs tableaux. Mon principal point d'intérêt est le tableau "Ship on time", qui regroupe toutes les transactions des pièces sur une période choisie, ainsi que les détails des colis, notamment le poids, la quantité, les transporteurs, les modes de transport, les dimensions, les délais, etc.

3.4 One Model Explorer

J'utilise cette base de données pour rechercher les composants de chaque pièce. Ainsi, je peux identifier les références de pièces ayant des composants similaires, ce qui me permet de segmenter les pièces en groupes pour effectuer des analyses ultérieures.

4 Méthodologie du projet

Jusqu'à présent, nous avons présenté les objectifs, le contexte et les données. La prochaine étape consiste à laisser les données parler d'elles-mêmes et à identifier les raisons des défaillances dès l'arrivée des pièces (FOA). Pour cette analyse, nous suivrons la méthodologie suivante :

4.1 Choix des critères

On commence par comprendre le processus et sélectionner les critères pertinents à partir des données fournies par l'entreprise. Ces critères peuvent inclure des informations telles que les caractéristiques des pièces, les fournisseurs, les régions géographiques, les données de transaction, les dates d'acquisition, etc. Le choix judicieux de ces critères nous permettra d'effectuer une analyse approfondie.

Après avoir échangé avec plusieurs collègues, j'ai choisi ces critères suivantes comme les features de Machine Learning :

1. Cycle de vie de la pièce :
 - Km de transaction : La distance parcourue par la pièce lors des transactions.
 - Nombre de transactions : Le nombre total de transactions impliquant la pièce, y compris les retours et les surplus.
 - Entrepôts visités : Le nombre d'entrepôts par lesquels la pièce est passée.
 - FE visités : Le nombre de Field Engineers (techniciens sur site) impliqués dans les interactions avec la pièce.
 - Expérience moyenne des FE visités : La moyenne jours d'embauche des FEs.
 - Centre de réparation : Centres de réparation visités et le nombre de fois où la pièce y a été réparée.
 - Jours passés dans l'entrepôt(max, min, median) : Calculer la durée de stockage dans chaque entrepôt, enregistrer les jours les plus longs dans l'entrepôt et l'entrepôt correspondant.
 - Région et sous-région : La description de la région où la pièce a été installée, permettant de comprendre la relation avec les lieux d'installation tels que le climat et la fréquence d'utilisation.
 - Activity_Month_Close_Date : la date à laquelle la pièce a connu une panne ou une défaillance, ce qui peut être lié à la résistance à des températures élevées.
 - Livreurs utilisés : Les livreurs utilisés pendant les transactions des pièces
2. Informations sur le système :
 - Système ID : L'identifiant du système dans lequel la pièce est installée.
 - Système Modality : La modalité du système médical dans lequel la pièce est utilisée.
 - Système age : l'âge du système depuis la date d'installation.
 - Goldseal_flag : D'après ce que j'ai compris, si Goldseal_flag==True, c'est à dire cette machine est acheté en occasion. Parfois certaines hôpitaux riches voudraient changer leurs machines à la nouvelle version, mais l'ancienne machine reste bonne, dans ce cas là, on vend cette ancienne machine avec Goldseal_flag==True.
3. Fiabilité :
 - Nombre de réparations : Le nombre total de réparations effectuées sur la pièce.

- Nombre de FOA : le nombre de fois où la pièce a été signalée comme ayant une défaillance dès son arrivée (FOA)
4. Propriétés de la pièce :
- Poids et dimensions : Les caractéristiques physiques de la pièce.
 - Classification : La classification de la pièce (Prime, Repairable, Harvest) et le service auquel elle est destinée.
 - Catégorie : La catégorie à laquelle la pièce appartient.
 - Product_identifier : la version spécifique de la pièce.
 - Family_name : le nom de la famille à laquelle la pièce appartient, même si la référence est différente.

4.2 Construction de la base de données

Une fois que nous avons défini nos caractéristiques (features), la prochaine étape consiste à collecter les données qui contiennent toutes ces caractéristiques. Cela nous permettra d'appliquer des méthodes de Machine Learning, telles que le clustering et les forêts aléatoires, pour extraire des connaissances précieuses et établir des modèles prédictifs.

La collecte des données se divise en deux parties. Dans la première partie, nous collections directement les données depuis Service Suite. Dans la seconde partie, nous effectuons des manipulations pour obtenir les données nécessaires.

Il est important de noter qu'en raison de certaines limitations, nous ne pouvons actuellement suivre que les pièces qui ont été réparées. Autrement dit, une pièce n'entre dans notre système informatique et ne peut être suivie qu'après avoir été réparée au moins une fois.

4.2.1 Collection des données de Service Suite

La collection des données de Service Suite est simple, simplement définir les conditions de filtrages, choisir les colonnes(features) et puis exporter le tableau ça suffit.

4.2.2 Tracing des pièces

Au part des critères collectées de Service Suite, on a beaucoup d'autres critères qui demande de manipulation. Par exemple, calculer les séjours dans chaque entrepôts, combien de fois les livreurs sont utilisés, calculer l'age du système etc. Avant de faire ces manipulations, il est important de définir le point de départ pour retrouver les pièces. On a décidé de partir de l'événement. Le paramètre lié avec l'événement est les RMAs. La logique de tracing par RMAs sont illustré dans la figure 9 :

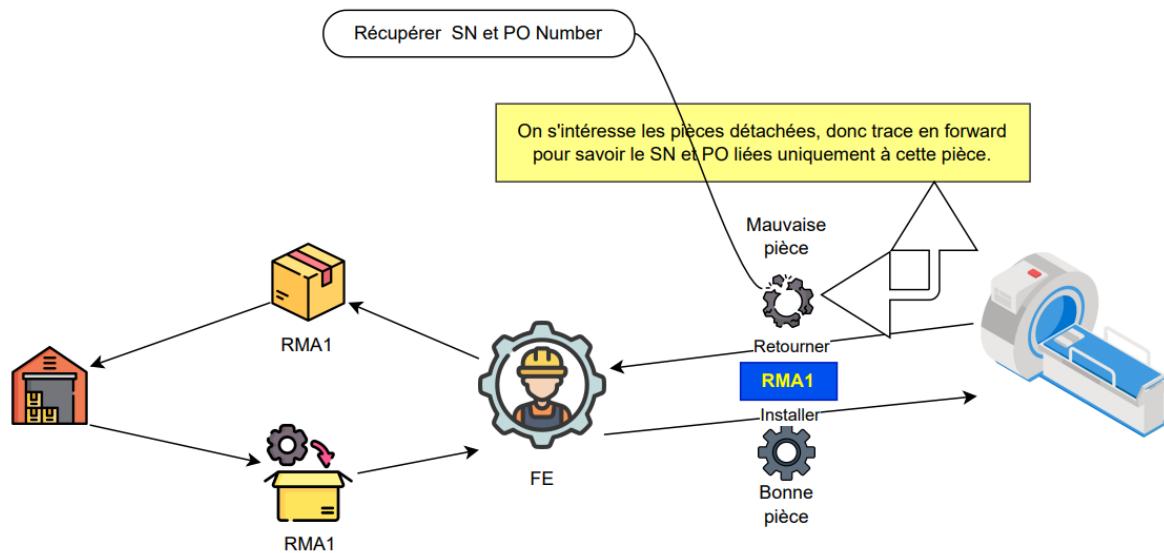


FIGURE 9 – Tracing par RMAs

Les Return Merchandise Authorizations (RMA) sont une série de chiffres que nous définissons lors du remplacement des pièces. C'est ce numéro qui lie la pièce en bon état à la pièce cassée. Pour illustrer la notion de RMA, imaginez-la comme une boîte en carton. Lorsque les pièces sortent de l'entrepôt, elles sont placées dans cette boîte, puis envoyées chez le FE pour réparation de la machine. Le FE installe les pièces en bon état sur le système, puis remplace les pièces défectueuses, les place à nouveau dans la même boîte et les renvoie à l'entrepôt. Ensuite, en fonction de l'état des pièces, elles sont soit envoyées au centre de réparation, soit mises au rebut. Ainsi, pour obtenir des informations sur les pièces défectueuses, nous devons rechercher les RMAs en suivant le flux en avant.

Cependant, ce qui nous intéresse particulièrement, c'est de comparer les différences entre les pièces qui présentent des défaillances dès leur arrivée (FOA) et celles qui n'en présentent pas, en termes d'événements qu'elles ont connus avant leur installation dans le système. Par conséquent, pour une pièce réparée, nous nous concentrerons principalement sur la période entre sa sortie du centre de réparation et son installation dans la machine.

Pour obtenir ces informations, nous partons de l'événement du remplacement des pièces. Après avoir collecté toutes les RMAs liées aux défaillances dans Service Suite, nous traçons ces RMAs en avant pour obtenir les numéros de commande d'achat (Purchase Order (PO)) correspondants, puis nous traçons ces PO numbers pour savoir ce qui s'est passé après la sortie du centre de réparation. Heureusement, au sein de notre équipe, nous comptons sur un doctorant qui a déjà développé un outil de traçabilité. En entrant différents paramètres (RMA, PO number, SO(Sale Order (SO)) number, etc.), nous pouvons retracer l'historique d'une pièce, soit en avant (transactions après le paramètre entré), soit en arrière (transactions avant le paramètre entré), et générer automatiquement un fichier qui enregistre l'historique de la pièce.

Cependant, il est important de noter que toutes les pièces ne sont pas traçables. Les données existent parce qu'elles sont enregistrées par quelqu'un quelque part. La traçabilité d'une pièce nécessite plusieurs étapes, et toute absence de données à l'une de ces étapes ou toute erreur dans les enregistrements peut entraîner l'impossibilité de suivre le parcours de la pièce. Malheureusement, nous ne pouvons pas changer cette situation. Nous devons donc faire face à cette contrainte et agir dans les limites de nos moyens. Comme on dit,

nous devons "danser avec des chaînes aux pieds". Notre objectif sera donc de chercher des tendances dans les données que nous pouvons suivre. Malgré les limitations, en exploitant les informations disponibles, nous pourrons toujours tirer des enseignements précieux et identifier des pistes pour comprendre les défaillances dès l'arrivée (FOA)."

4.3 Calcule des critères

Maintenant on a tracé l'histoire d'une pièce comme un fichier, c'est le moment de commencer à manipuler ces données. Ce processus s'est avéré être un exercice très enrichissant dans le traitement des données avec Python. J'ai appris à nettoyer les données en éliminant les valeurs bizarres ou manquantes, ainsi qu'à uniformiser les types de données pour faciliter l'analyse et utiliser les expressions régulières pour reformuler les données etc. Concernant les détails spécifiques sur le calcul de chaque critère, je ne vais pas rentrer trop dans les détails. Cependant, si vous êtes intéressés, vous pouvez trouver mon code et les explications correspondantes dans ce lien github : https://github.com/Anais-Y/FOA_root_cause

Après cette étape, on obtient les données comme ça 10 :

	AF2	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA
1	ces	B_distances	A_distances	tot_distances	nb_inout	adj_Serial_Number	FE_SSO	Repair_center	Repair_site_code	Repair_qty	failure_type	nb_FOA	SystemAge	Carrier	sejourWHS	FE visited	
2	[0]	13856	11525	25381	2	240667K6	[10003722] 'GE Healthcare' [GPO ASIA 511]			1	["Debrief"]	0	["NEVO6849"]	2093	["MRW"]	["I":00'..30', 'R1	1
3	6698.0]	8826	6698	15524	3	205057K6	[50305651] 'GE Healthcare' [GPO ASIA 511]			1	["Debrief"]	0	["246037K6"]	3582	["FEDEX"]	["I":00'..72", 'SU	1
4	,1068.0, 1068.0, 1068,	18116	0	18116	6	238342K6	[99998101] 'GE Healthcare' [GPO ASIA 511]			2	["Debrief"]	0	["8303784549"]	294	["Saam"]	["I":00'..14", 'A3	1
5	6698.0]	8826	6698	15524	3	255621K6	[50305651] 'GE Healthcare' [GPO ASIA 511]			1	["Debrief"]	0	["23897K6"]	106	["FEDEX"]	["I":00'..66", 'SU	1
6		1023	468	1491	2	201609K9	[21256777] 'GE Healthcare' [GPO ASIA 501', 'i			5	["[Blank]", "Debr	0	["013471U670"]	1426	["BYBOD"]	["I":00'..105", 'R	1
7	6698.0]	8826	6698	15524	3	SN-243297K9R	[50305651] 'GE Healthcare' [GPO ASIA 511]			1	["Debrief"]	0	["026626K9R"]	1527	["FEDEX"]	["I":00'..90", 'SU	1
8	,1064.0, 6698.0]	20479	6698	27177	5	248605K8	[50305651] 'GE Healthcare' [GPO ASIA 511]			3	["[Blank]", "Debr	0	["233411K8R"]	2567	["FEDEX"]	["I":00'..88", 'R0	2
9		562	0	562	2	231687K8	[22301985] 'GE Healthcare' [GPO ASIA 511]			1	["Debrief"]	0	["0486589901"]	1856	["CIBLEX"]	["I":00'..61", 'SFI	1
10	707.0, 2707.0, 2707.0,	18267	741	19008	6	210501K07	[40301205] 'GE Healthcare' [GPO ASIA 501', 'i			2	["[Blank]", "Debr	0	["709651L9E9"]	3743	["USPROT"]	["I":00'..6", 'SCA	1
11	,1068.0, 1068.0, 1068,	18116	13844	31960	6	248276K8R	[50268837] 'GE Healthcare' [GPO ASIA 511]			2	["Debrief"]	0	["8303780201E"]	1693	["Saam"]	["I":00'..62", 'A3	1
12		1218	0	1218	2	246004K6	[10004096] 'GE Healthcare' [GPO ASIA 511]			1	["Debrief"]	0	["121555JU04"]	729	["TNT"]	["I":00'..57", 'SD	1
13	6698.0]	8826	6698	15524	3	24571K7R4	[50305651] 'GE Healthcare' [GPO ASIA 511]			2	["Debrief"]	0	["98842K8R"]	2971	["FEDEX"]	["I":00'..50", 'SU	1
14		0	0	0	[]	1	239599K4	[GPO ASIA 501', 'i		3	["Debrief"]	0	["I"]	["I":00'..6], [0		
15	0,420.0,643.0]	25436	643	26079	4	207976K62	[21273601] 'GE Healthcare' [GPO ASIA 501', 'i			2	["Debrief"]	0	["AS355501"]	3898	["TNT"]	["I":00'..21", 'R1	2
16	703.0, 703.0, 390Z.0,:	16130	14616	30746	5	27796K6	[21251712] 'GE Healthcare' [GPO ASIA 511]			1	["Debrief"]	0	["83037240512"]	1606	["Local Co"]	["I":00'..41", 'A3	1
17	6698.0,6698.0]	8826	13396	22222	3	221050K7	[50305651] 'GE Healthcare' [GPO ASIA 511]			1	["Debrief"]	0	["0277865K2"]	663	["FEDEX"]	["I":00'..49", 'SU	1
18		9665	0	9665	2	257385K8R	[]			1	["Debrief"]	0	["I"]	["I":00'..6], [0		
19		562	0	562	2	207560K6	[21256775] 'GE Healthcare' [GPO ASIA 501', 'i			2	["Debrief"]	0	["0503111100"]	651	["CIBLEX"]	["I":00'..24", 'SI	1
20	1064.0]	9890	0	9890	5	275151K9R	[21204206] 'GE Healthcare' [GPO ASIA 511]			1	["[Blank]", "Debr	0	["I"]	["I":00'..6,"R00	0		
21	1064.0]	9890	0	9890	5	275151K9R	[21204206] 'GE Healthcare' [GPO ASIA 511]			1	["[Blank]", "Debr	0	["I"]	["I":00'..3,"SUS	1		
22	,3130.0, 1068.0, 1068,	26512	13844	40356	10	230975K9R	[50162405] 'GE Healthcare' [GPO ASIA 511]			1	["Debrief"]	0	["83037847487C"]	579	["FEDEX"]	["I":00'..3, 'SUS	1
23	6698.0]	8826	6698	15524	3	217958K8R	[50301117] 'GE Healthcare' [GPO ASIA 511]			1	["Debrief"]	0	["I"]	["I":00'..50,"SUS	1		
24		0	0	0	[]	1	262056K5R	[GPO ASIA 511]		1	["Debrief"]	0	["I"]	["I":00'..59,"SUS	1		
25	,1886.0, 1886.0, 4416	27089	10714	37803	6	2504901K2R	[503020131] 'GE Healthcare' [GPO ASIA 511]			1	["Debrief"]	0	["I"]	["I":00'..3], [0		
26		6698	0	6698	2	210655K6R	[]			1	["Debrief"]	0	["I"]	["I":00'..32,"A0	1		
27		8826	0	8826	3	189446K6	[21280645] 'GE Healthcare' [GPO ASIA 511]			1	["Debrief"]	0	["I"]	["I":00'..8,"U00	0		
28	1064.0]	9890	0	9890	5	24396K6R	[50305651] 'GE Healthcare' [GPO ASIA 511]			2	["Debrief"]	0	["I"]	["I":00'..6,"SUS	1		
29	,3130.0, 1068.0, 1068.0, 10	30544	13844	53388	8	268922K2R	[50301741] 'GE Healthcare' [GPO ASIA 511]			1	["Debrief"]	0	["I"]	["I":00'..25,"A3	1		
30	6698.0]	8826	6698	15524	3	228389K63	[50305651] 'GE Healthcare' [GPO ASIA 511]			1	["Debrief"]	0	["I"]	["I":00'..22,"T2R9P	0		
31	,1886.0, 1886.0, 1739	24412	16769	41181	6	192792K9R	[22302295] 'GE Healthcare' [GPO ASIA 501', 'i			2	["Debrief"]	0	["I"]	["I":00'..25,"A3	1		
32	6698.0]	8826	6698	15524	3	208671K6R	[50305651] 'GE Healthcare' [GPO ASIA 511]			2	["Debrief"]	0	["I"]	["I":00'..8,"SU	1		
33	6698.0]	8826	6698	15524	3	240671K8R	[50305651] 'GE Healthcare' [GPO ASIA 511]			1	["Debrief"]	0	["I"]	["I":00'..76,"SU	1		
34	,1068.0, 1068.0,	17048	0	17048	6	261067K93	[]			2	["Debrief"]	0	["I"]	["I":00'..28,"A3	0		

FIGURE 10 – Données après manipulations

4.4 Encoder les données

Comme vous le voyez, certains critères ne sont pas numériques, il est nécessaire d'encoder les données pour faciliter l'analyse ultérieure.

Pour les données caractères, on utilise les fonctions dans `sklearn.preprocessing` [1] pour encoder. Après comparer les différentes méthodes soigneusement, on décide d'utiliser label encoder en même temps générer une table de correspondance pour le label encoder.

Pour les données sous forme de liste, chaque situation nécessite une analyse au cas par cas. Pour les livreurs et centres de réparation, on compte le nombre d'occurrences de chaque élément dans la liste, puis on ajoute une nouvelle colonne pour enregistrer

combien de fois cet élément est utilisé. Pour la liste des FEs passé par cette pièce, on calcule la moyenne des expériences de ces FEs. Pour la dictionnaire des entrepôts et leurs séjours, on ajoute des colonnes ***Max/Median/Min jours dans Warehouse*** et ***Max/Median/Min Warehouse*** correspondants.

A la fin on arrive à des données comme montrer dans la figure 11 :

Customer	Country	Customer	Pdcr	of	Warenber	of ret	Number	of Surplus	Number	of Pole	B_distan	Repaire_qty	nb_FOA	SystemAge	FE visited	adjusted_earlylife_failure	elf_days	failed_part_description	Product	identif	failur
0		4	2	2	1	0	1	13856	1	0	2093	1	2	895	1	14	36				
1		22	0	2	1	0	3	8826	1	0	3582	1	2	3525	0	7	24				
2		9	1	3	1	1	2	18116	2	0	294	1	2	414	1	28	55				
4		6	2	2	1	0	1	1023	5	0	1426	1	2	568	0	2	65				
5		22	0	2	1	0	3	8826	1	0	1257	1	2	479	1	16	39				
6		22	0	3	2	1	4	20479	3	0	2567	2	2	800	2	2	61				
7		5	2	1	1	0	1	562	1	0	1856	1	2	1384	1	18	42				
8		3	0	3	1	0	2	18267	2	0	3743	1	2	1736	0	4	12				
9		9	1	3	1	0	3	18116	2	0	1693	1	2	447	1	21	44				
11		22	0	2	1	0	3	8826	2	0	2971	1	2	1076	0	4	10				
14		9	1	4	1	0	3	16130	1	0	1606	1	2	814	0	7	32				
15		22	0	2	1	0	4	8826	1	0	663	1	2	535	1	19	43				
17		5	2	1	1	0	1	562	2	0	651	1	2	383	0	4	12				
18		22	0	2	1	0	2	9890	1	0	934	1	2	1035	1	26	53				
19		22	0	2	1	0	2	9890	1	0	579	1	2	559	1	16	39				
20		9	1	3	1	0	3	26512	1	0	958	1	2	943	1	26	53				
21		22	0	2	1	0	3	8826	1	0	1639	1	2	604	0	2	62				
23		0	1	3	1	0	3	27089	1	0	1786	1	0	91	0	3	67				
25		22	0	2	1	0	2	8826	1	0	1216	1	2	184	0	2	65				
26		22	0	2	1	0	2	9890	2	0	1210	1	2	952	1	16	39				
27		9	1	3	1	0	5	39544	1	0	1078	1	2	984	1	21	44				
29		0	1	4	1	0	3	24412	2	0	1647	1	0	114	0	4	14				
30		22	0	2	1	0	3	8826	2	0	627	1	2	598	0	2	64				
31		22	0	2	1	0	3	8826	1	0	1821	1	2	1674	1	23	50				
34		22	0	2	1	0	3	8826	2	0	1635	1	2	482	1	15	38				
35		9	1	4	1	0	3	17751	1	0	1134	1	2	1112	1	21	44				
37		22	0	2	1	0	3	8826	2	0	551	1	2	378	0	3	68				
38		9	1	5	2	1	3	18255	1	0	204	2	2	1093	1	19	43				
39		5	2	1	1	0	1	988	1	0	1270	1	2	909	1	25	52				
42		22	0	2	1	0	3	8826	1	0	1424	1	2	1417	1	13	35				
45		5	2	1	1	0	1	562	1	0	1563	1	2	1904	0	4	12				
46		22	0	2	1	0	3	8826	3	0	25	1	0	14	1	19	43				
48		9	1	4	1	0	3	16130	1	0	1826	1	0	2	1	21	44				

FIGURE 11 – Données après encoder

4.5 Choisir l'algorithme utilisé

Suite à la préparation des données, on est désormais prêt à explorer différents algorithmes et à choisir celui ou ceux qui seront les plus appropriés pour l'analyse. Pour ce faire, j'ai collecté des données spécifiques pour la pièce de référence *2351573-R*, afin de réaliser des tests avec différents algorithmes.

Parmi les 414 lignes de données collectées, on a identifié les occurrences suivantes pour chaque label de défaillance :

- 88 FOA (label : 0) - 12 FOI (label : 1) - 5 ELF90 (label : 2) - 28 ELF180 (label : 3)

- 281 IPFR>180 (label : 4)

4.5.1 Méthodes non-supervisées

On commence l'analyse en se concentrant sur les algorithmes de clustering mentionnés dans le sujet de stage.

Le clustering est une technique d'apprentissage non supervisé qui permet de regrouper les données similaires dans des ensembles appelés "clusters". En explorant les résultats de différents algorithmes de clustering, on cherche à identifier des schémas et des structures dans les données, ce qui pourrait révéler des groupes de pièces partageant des caractéristiques communes liées à leur défaillance dès l'arrivée (FOA). En outre, on a les vrais labels pour vérifier si l'algorithme a bien fait son job et regrouper les pièces avec différentes défaillances. [2]

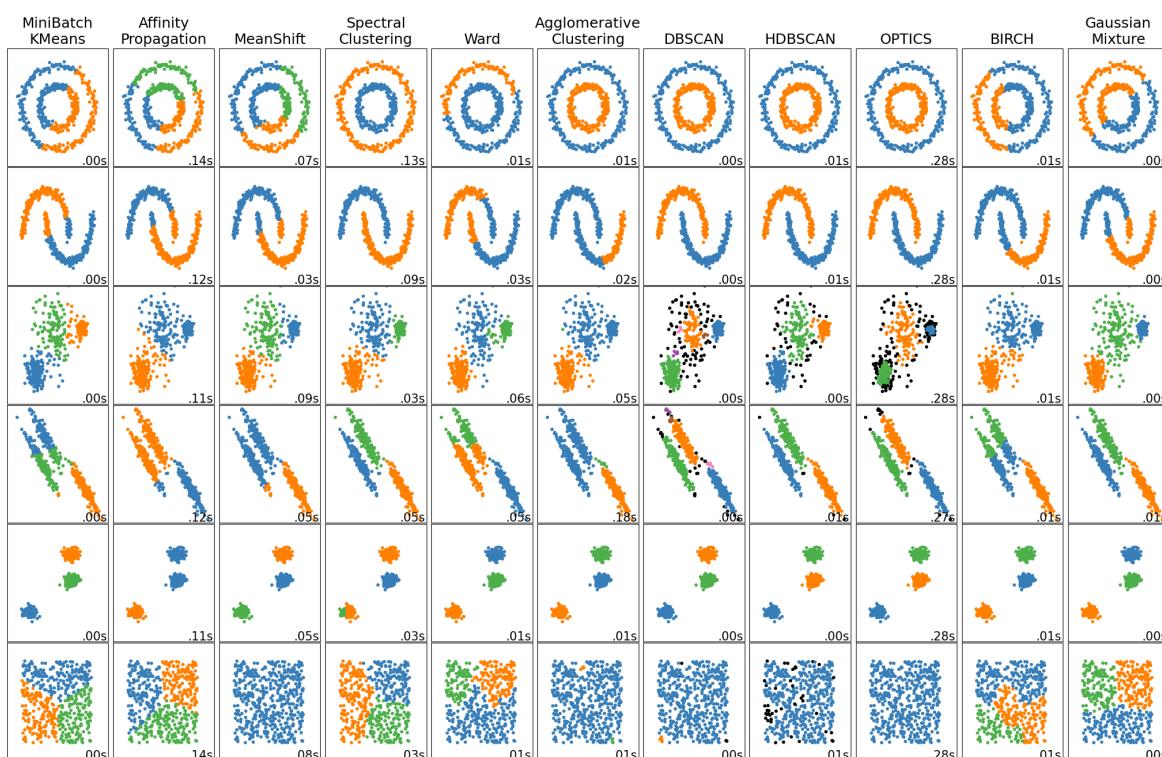


FIGURE 12 – Comparaison des méthodes dans sklearn

1. D'abord on essaie l'algorithme le plus connu : **K-means**[3]

K-means vise à diviser les données en K clusters, où K est un nombre prédéfini par l'utilisateur. L'algorithme commence par initialiser K centres de cluster de manière aléatoire. Ensuite, il attribue chaque point de données au cluster dont le centre est le plus proche, et met à jour les centres des clusters en calculant la moyenne des points de chaque cluster. Ce processus est répété jusqu'à ce que les centres convergent vers une position stable.

Voici les résultats de K-means figure 13, la figure à gauche est les labels obtenus par K-means. Dans la figure à droite, j'ai fait visualisé ces points dans un espace de 3D en réduisant la dimension par Analyse Composante Principale et aussi visualisé les cluster centers. Car notre objectif est de distinguer les FOAs, j'affiche seulement les pièces FOA est pièces de très bonne qualité(IPFR>180).

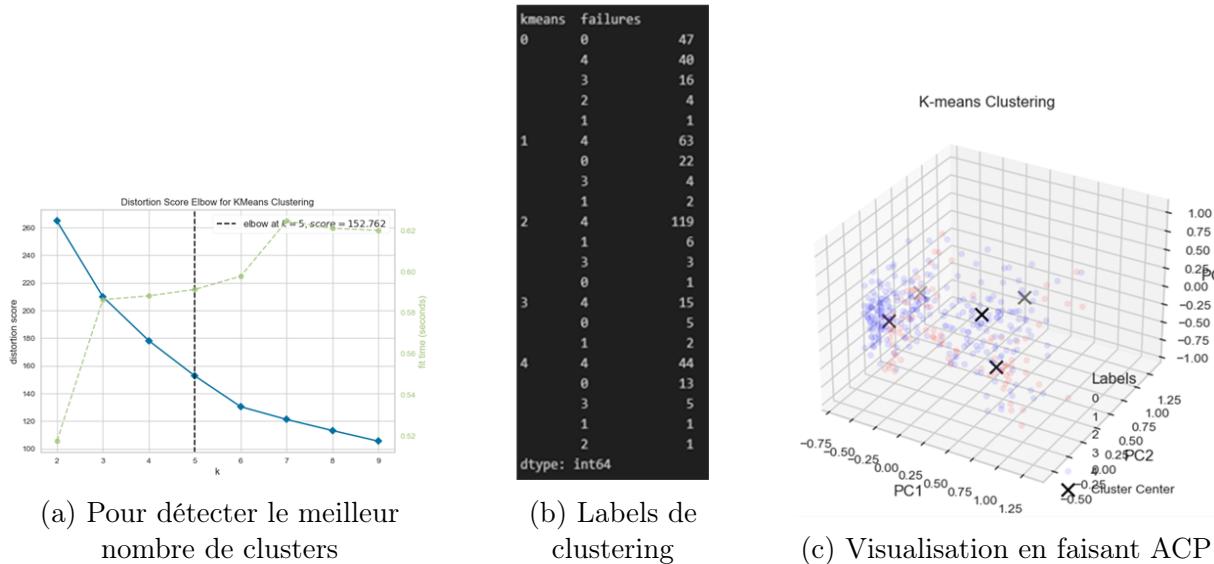


FIGURE 13 – K-means

On peut voir dans les figures, ça sépare un peu les FOAs et IPFR>180. Le cluster 2 concentre surtout les labels 4 et cluster0 concentre une moitié des labels 0.

2. Les résultats ne sont pas si satisfaisants, on voudrait essayer encore des méthodes : **Clustering Hiérarchique[4]**

Le clustering hiérarchique est une méthode de clustering agglomérative, qui commence par considérer chaque point de données comme un cluster individuel et fusionne progressivement les clusters similaires pour former une hiérarchie de clusters. Il existe deux approches pour le clustering hiérarchique : le regroupement ascendante (agglomerative) et le regroupement descendant (divisif). Le premier commence avec des clusters individuels et les fusionne progressivement pour former des clusters plus grands, tandis que le second commence avec un cluster global et le divise en sous-clusters. Après avoir essayé les deux méthodes, j'ai pas vu grande différence, donc ici on choisi clustering agglomérative.

Voici les résultats de cette méthode :

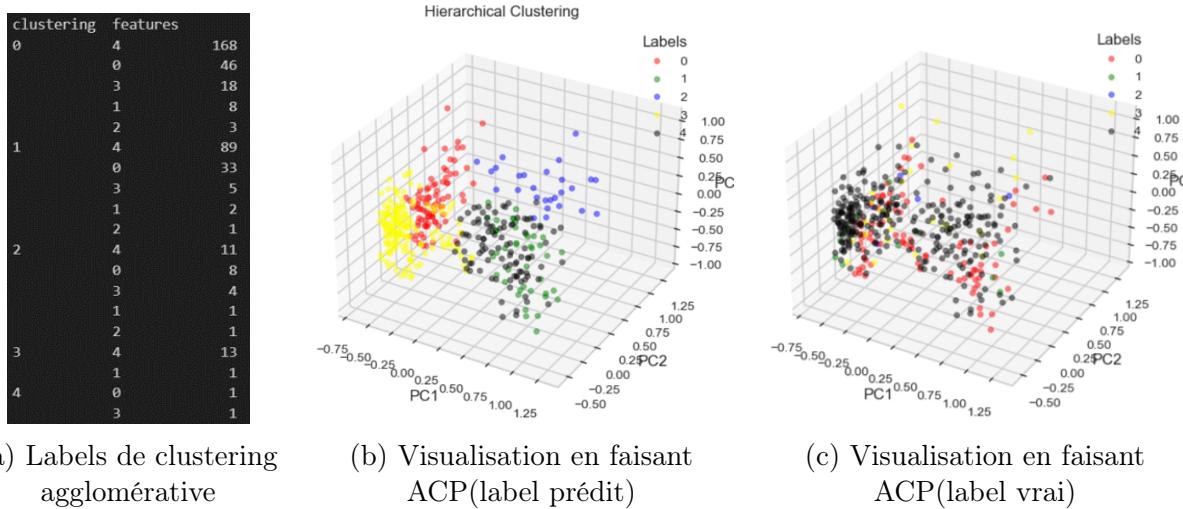


FIGURE 14 – Clustering Hiérarchique

Malheureusement cette méthode est encore pire que K-means. Elle ne sépare pas du tout les différentes pièces.

3. DBSCAN (Density-Based Spatial Clustering of Applications with Noise)[5] :

DBSCAN est un algorithme de clustering basé sur la densité qui peut détecter des clusters de forme arbitraire, tout en étant robuste aux points aberrants. L'algorithme attribue à chaque point de données une étiquette de cluster en fonction de la densité des points dans son voisinage. Les points appartenant à des régions denses du jeu de données sont attribués à un cluster, tandis que les points isolés et les points situés dans des régions peu denses sont considérés comme du bruit.

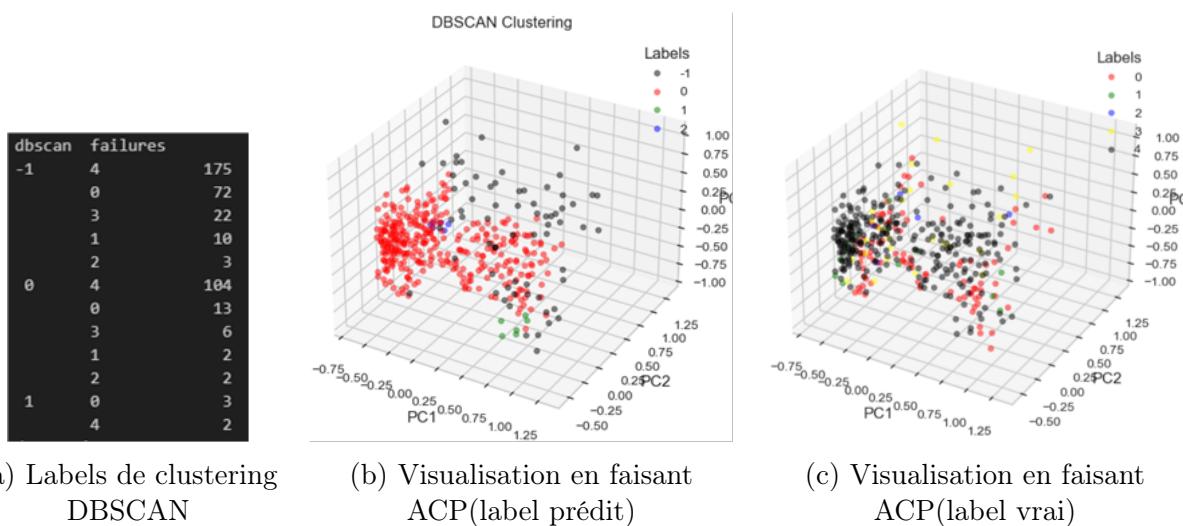


FIGURE 15 – Clustering DBSCAN

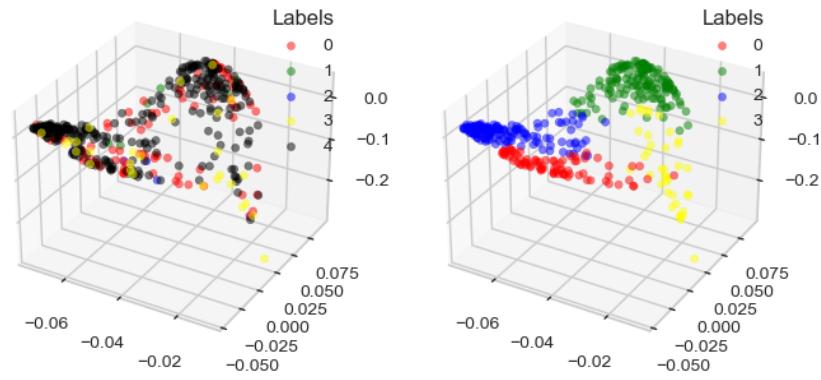
DBSCAN aussi, ne peut pas bien séparer les différentes pièces. On cherche encore d'autre méthode pour mieux classifier. Dans le cours de DATASIM, on fait souvent les transformations spectrales, donc je voudrais essayer de passer par l'espace spectral pour ces données.

4. Spectral Clustering[6]

Guidé par vidéo YouTube et cette référence [7], j'ai calculé la matrice laplacienne des données et faire les compositions des valeurs propres. En gardant 3 premiers eigenvalues et utiliser encore le clustering k-means, on arrive aux résultats comme ça :

	Spectral Clustering	failures
0	4	106
	0	36
	1	5
	3	5
1	4	127
	0	25
	3	9
	1	6
2	2	1
	4	48
	0	27
	3	14
4	2	4
	1	1

(a) Labels de clustering Spectral



(b) Visualisation en faisant ACP(label vrai à gauche, label prédit à droite)

FIGURE 16 – Clustering spectral

Maintenant, il semble que même en utilisant le clustering dans l'espace de fréquences, les résultats ne se sont pas améliorés. Donc il est temps de considérer l'utilisation d'algorithmes de Machine Learning supervisés pour la classification des pièces en fonction de leurs défaillances.

4.5.2 Méthodes supervisées(Forêt Aléatoire)

Forêt aléatoire[8] est un algorithme d'apprentissage supervisé utilisé pour la classification et la régression. Elle se compose d'un ensemble d'arbres de décision, construits à partir de sous-ensembles aléatoires des données d'entraînement. Chaque arbre vote pour la classe majoritaire (classification) ou fournit une prédition moyenne (régression). Voici les explications de Forêt Aléatoire[9] :

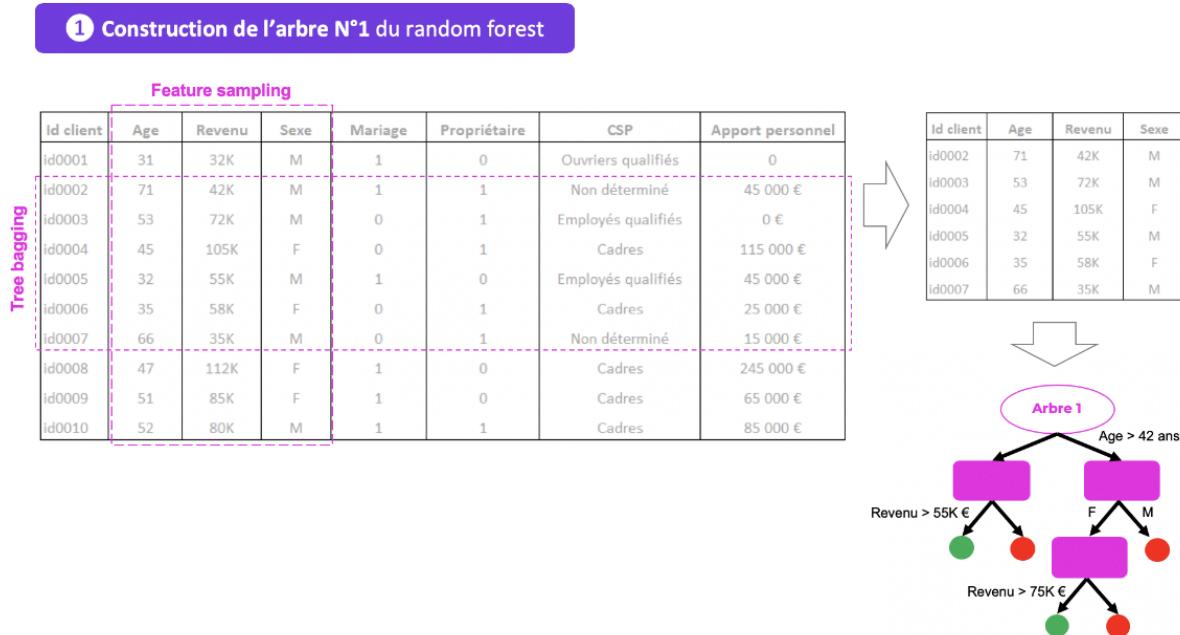


FIGURE 17 – Forêt Aléatoire - étape 1

2 Mise en œuvre d'une forêt d'arbres

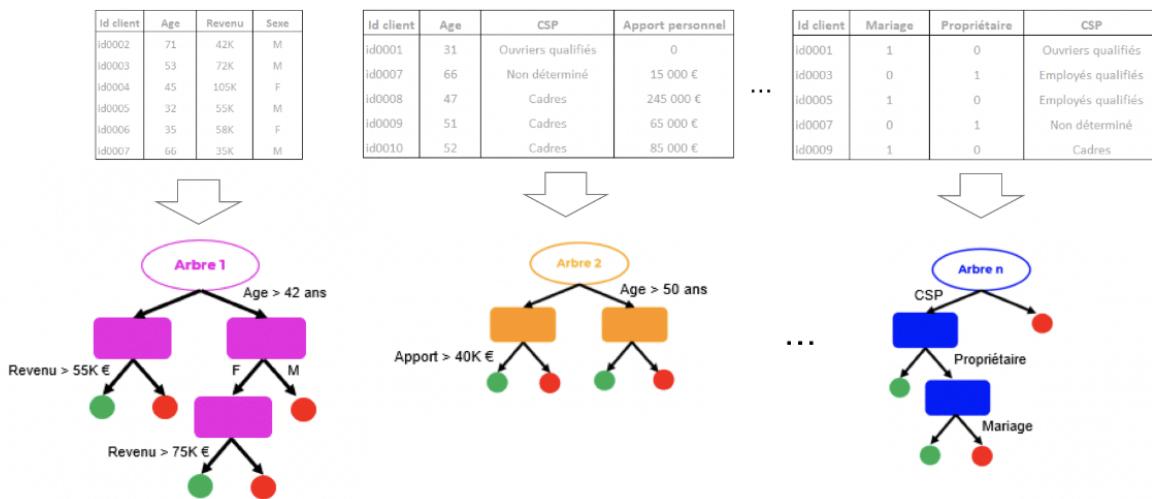


FIGURE 18 – Forêt Aléatoire - étape 2

3 Prédiction du random forest

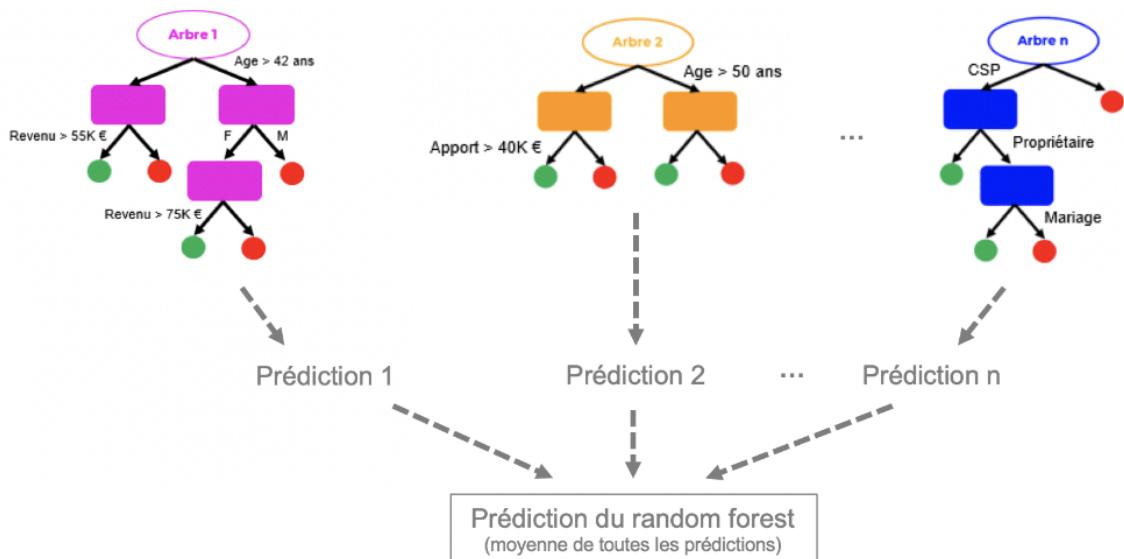


FIGURE 19 – Forêt Aléatoire - étape 3

Étant donné que la random forest peut identifier les caractéristiques importantes en utilisant la méthode "*feature importance*" dans *sklearn*, cette capacité nous semble extrêmement utile. Par conséquent, on a décidé d'expérimenter l'utilisation de la random forest pour classifier ces pièces. Toujours commencer de faire l'expérimentation avec une référence, on obtient un accuracy de prédiction assez élevé : pour distinguer 2 classes(ELF et IPFR>180, la vie d'une pièce dure moins 180 jours ou plus de 180 jours), on atteint un accuracy de 92.3%, qui implique l'efficacité de cette méthode.

4.6 Analyse de haut niveau

Après testé plusieurs méthodes, trouve que K-means et Forêt Aléatoire sont les plus performantes, donc on utilise principalement ces deux méthodes pour étapes suivantes.

Ensuite on va segmenter les pièces et faire nos analyses. Voici les pistes qu'on a essayé pour faire les analyses :

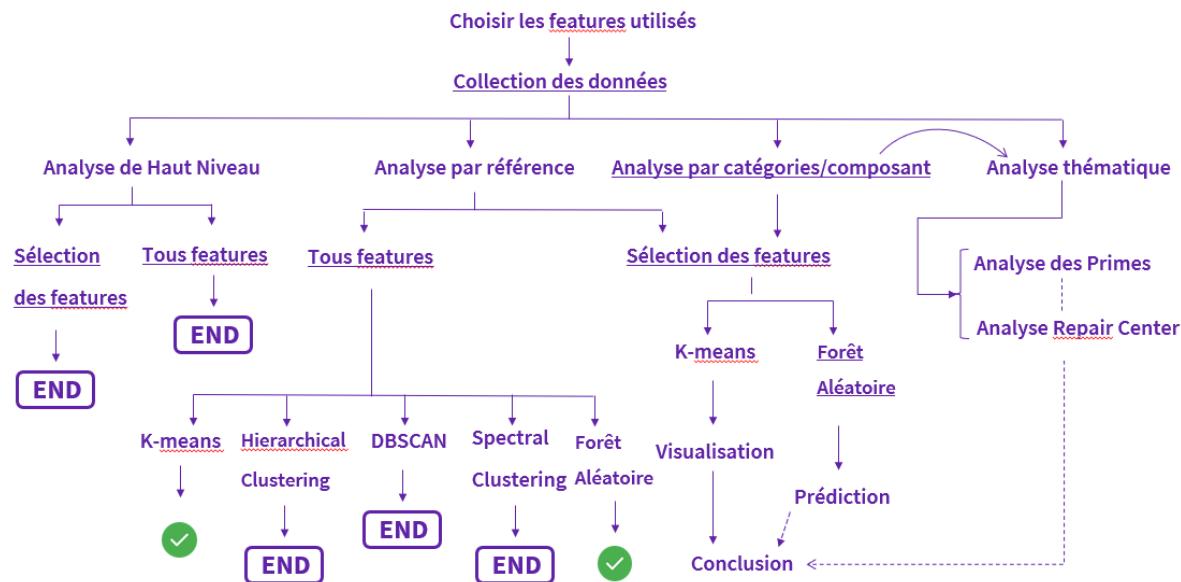


FIGURE 20 – Plan d’analyses

On commence par analyse de haut niveau, c'est à dire on analyse toutes les pièces de différentes catégories pour obtenir une grande direction qui pourrait aider dans les suivantes.

4.6.1 Tous features

Au tout début, j'ai inclus toutes les caractéristiques sans effectuer feature engineering. En filtrant par la modalité MR, j'ai extrait toutes les pièces en panne depuis 2019, ce qui représente environ 500 000 pièces. En calculant toutes les caractéristiques possibles et en examinant les corrélations, voici les résultats obtenus :

	adjusted_earlylife_failure	1	1
adjusted_earlylife_failure			
system_age	0,148885461	0,148885	
planning_name	-0,114634874	0,114635	
key_group_desc	0,09079073	0,090791	
lifecycle	0,08946734	0,089467	•
supplier_failed_flag	0,083587756	0,083588	
failed_part_classification	-0,081653987	0,081654	
product_group	-0,053695385	0,053695	
current_fe_return_rate	-0,049889644	0,04989	
product_identifier_description	-0,048695528	0,048696	
equipment	0,035625192	0,035625	
goldseal_flag	-0,034426211	0,034426	
family_name	-0,033350293	0,03335	
current_xelus_last_12month_demand	0,030055632	0,030056	
region_description	0,024056394	0,024056	
sub_region_description	0,02325309	0,023253	
current_xelus_last_12month_consumption	0,022987705	0,022988	
country_description	0,020611444	0,020611	
grp_of_countries_description	0,01773791	0,017738	
zone_description	0,015415138	0,015415	
Exp	0,011703972	0,011704	
asset_system_id_location	-0,011143631	0,011144	
asset_system_id	-0,009315012	0,009315	■
asset_channel_partner	0,007710555	0,007711	
product_identifier	0,006338075	0,006338	
segment	-0,005405944	0,005406	
market_segment	0,003491743	0,003492	
Activity_Month_Close_Date	-0,001614292	0,001614	
failed_part_supplier_flag		0	

FIGURE 21 – Corrélation de type de défaillances et les features pour toutes les pièces

En examinant les résultats de corrélation, aucune tendance significative n'a été observée. Mais seulement la corrélation n'est pas suffisant, j'ai aussi essayé de faire les clusterings pour ces données. Pour faciliter les clusterings, chaque labels on a choisi par hasard 1000 pièces.

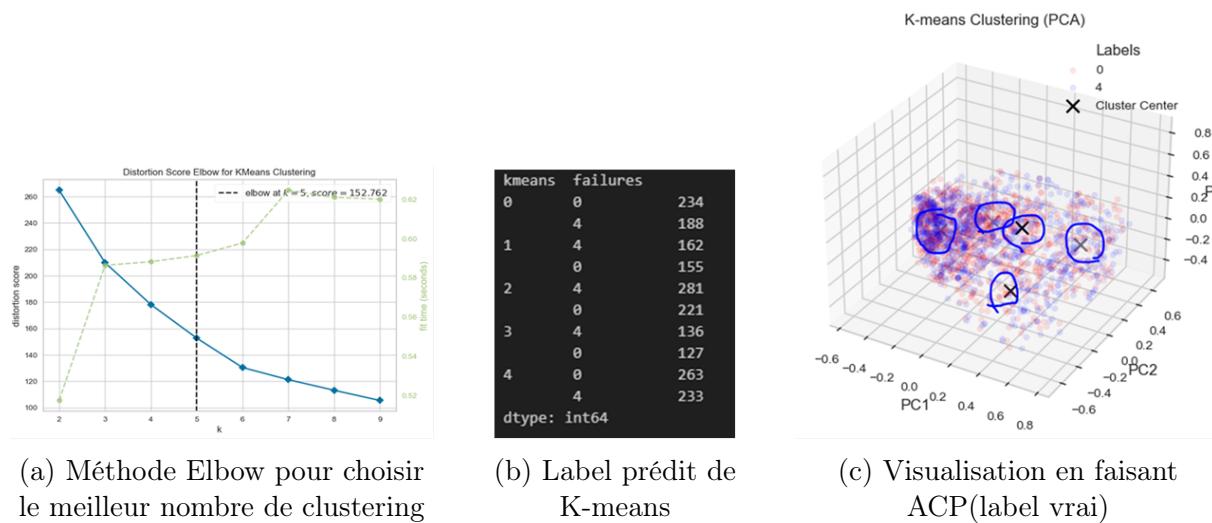


FIGURE 22 – Clustering K-means _ ANALYSE HAUT NIVEAU

On voit toujours rien. Peut-être changer encore une fois les méthodes :

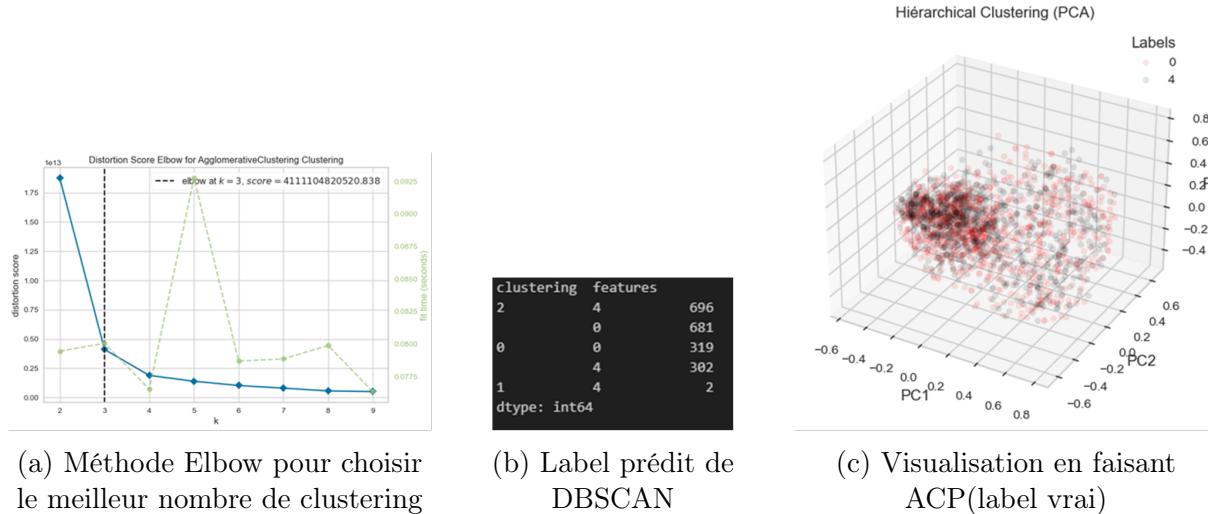


FIGURE 23 – Clustering DBSCAN _ ANALYSE HAUT NIVEAU

Il semble que cette approche ne soit pas efficace, ce qui est normal car nous avons examiné toutes les pièces et toutes les features. En fait parmi ces 30 features, il y en a beaucoup qui sont corrélé entre eux. Donc peut être choisir les features plustôt indépendants nous donnerait les résultats un peu mieux.

4.6.2 Feature Engineering

Donc on a sélectionné 10 features plutôt indépendantes et faire la même chose[10] [11] :

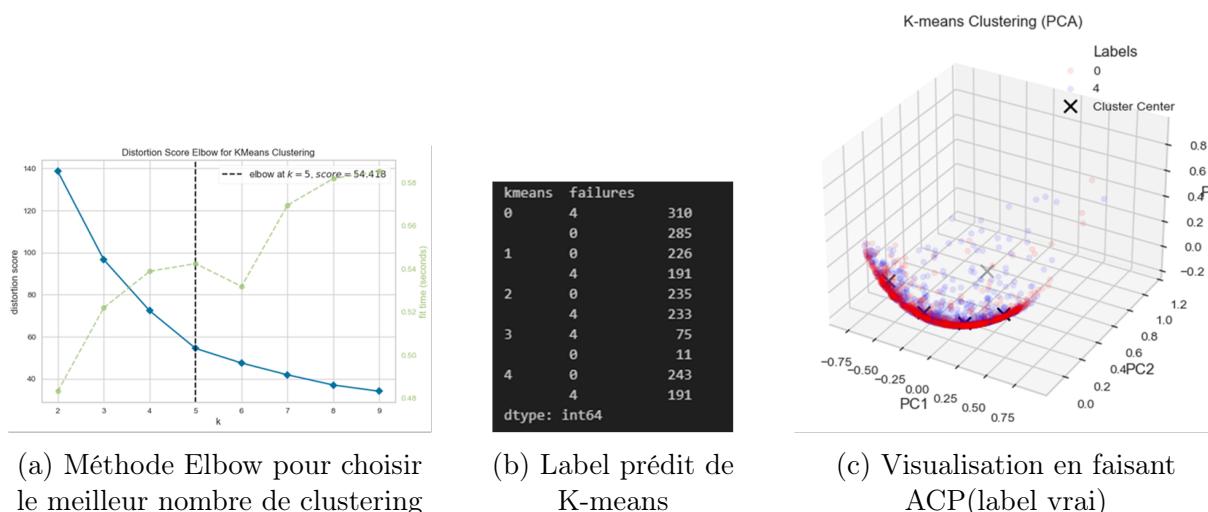


FIGURE 24 – Clustering K-means _ ANALYSE HAUT NIVEAU en sélectionnant les features

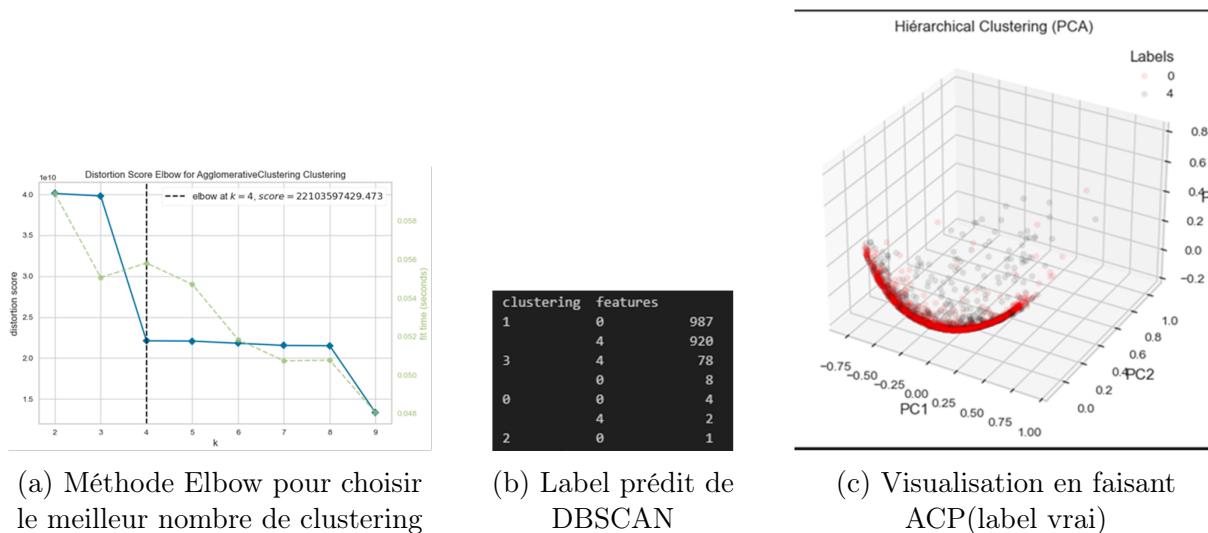


FIGURE 25 – Clustering DBSCAN_ANALYSE HAUT NIVEAU en sélectionnant les features

4.7 Analyse par référence

Dans la section 4.5, lors de l'exploration des méthodes appropriées avec la référence 2351573-R, on a séparé les pièces en cinq classes en fonction de type de défaillance (adjusted_failed_type) dans service suite. Cependant, j'ai remarqué que ces labels ne sont parfois pas précises. Par exemple, certaines pièces ont une durée de vie de 0 jours, mais sont étiquetées comme IPFR90 au lieu de FOA ou FOI. De plus, il sépare FOA et FOI comme deux différentes classes qui ne correspond pas à notre intention. Par conséquent, j'ai décidé de classifier les pièces en trois catégories en fonction de leur durée de vie : FOA pour une durée de vie de 0 jour, ELF pour une durée de vie de 0 à 180 jours, et IPFR180 pour une durée de vie de plus de 180 jours. D'ailleurs, quand on suivre des pièces FOA, certaines pièces jamais réparées peuvent également être suivies car pour obtenir l'histoire de pièces FOA, il n'y a pas besoins de Repair PO tandis que pour les pièces non-FOA, nous pouvons seulement suivre celles qui ont été réparées au moins une fois en utilisant le dernier repair PO. Pour éliminer ce biais, on concentre uniquement sur les pièces FOA qui ont été réparées une fois.

Après ce traitement, pour la référence 2351573-R, nous avons 358 pièces traçables avec une confiance totale, dont 52 sont des FOA, 25 sont des ELF et 281 sont des pièces de très bonne qualité.

Même si le clustering peut séparer légèrement les différentes pièces, il est difficile de retirer des tendances significatives, et ces tendances ne sont pas forcément fiables. C'est pourquoi nous avons décidé de nous concentrer sur les modèles de prédiction pour comprendre les facteurs qui contribuent aux résultats de prédiction et étudier notamment la distribution de chaque feature des pièces.

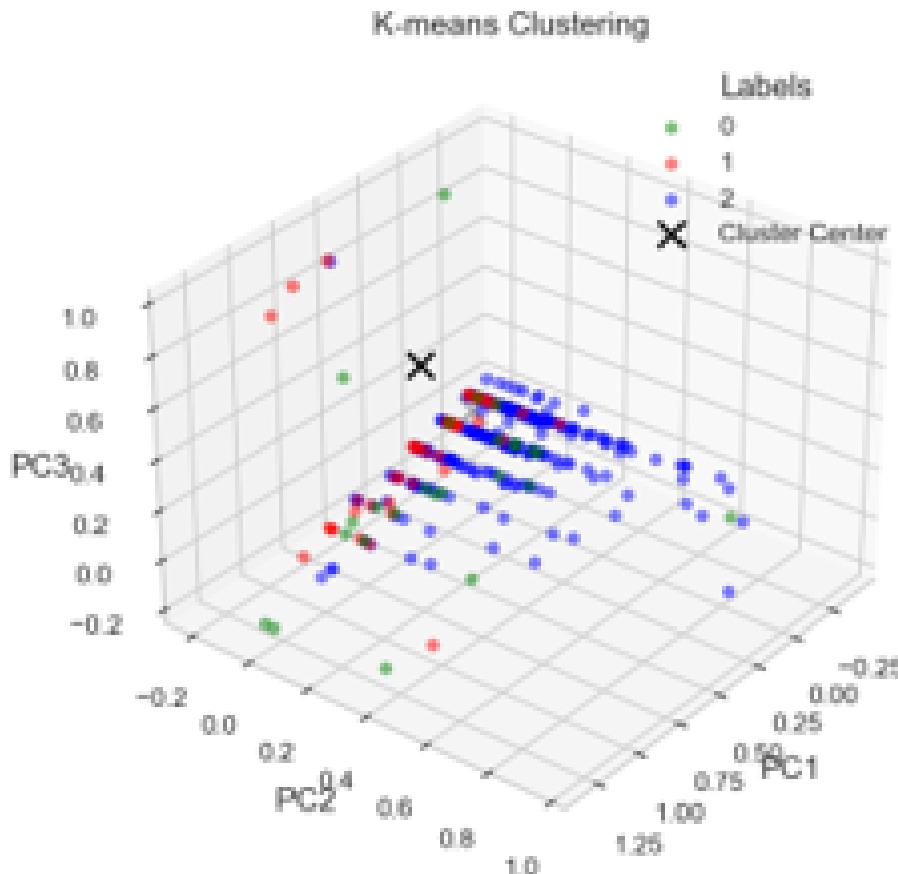


FIGURE 26 – Visualisation de clustering

En raison du déséquilibre entre le nombre de pièces de chaque catégorie, nous avons un grand nombre de pièces de très bonne qualité, tandis que les deux autres catégories ont un nombre plus restreint de pièces. Si on inclure toutes ces pièces dans le modèle de prédiction, celui-ci pourrait opter pour une approche simpliste en prédisant toutes ces pièces comme étant de très bonne qualité, ce qui conduirait à une précision élevée. Pour éviter cela, il est nécessaire de diviser judicieusement l'ensemble de données en ensembles de entraîner (train set) et de tester (test set). Cela permettra d'assurer une répartition équilibrée des différentes catégories de pièces et de garantir des prédictions plus précises et équitables.

Effectivement, la répartition de l'ensemble de formation (train set) et de l'ensemble de test (test set) est également un processus aléatoire, tout comme les prédictions du modèle. Ainsi, la précision de prédiction du modèle est influencée par ces variations. Cependant, pour cette référence, en général, la précision se situe toujours au-dessus de 70 %.

Voici les feature importances quand on fait la prédiction :

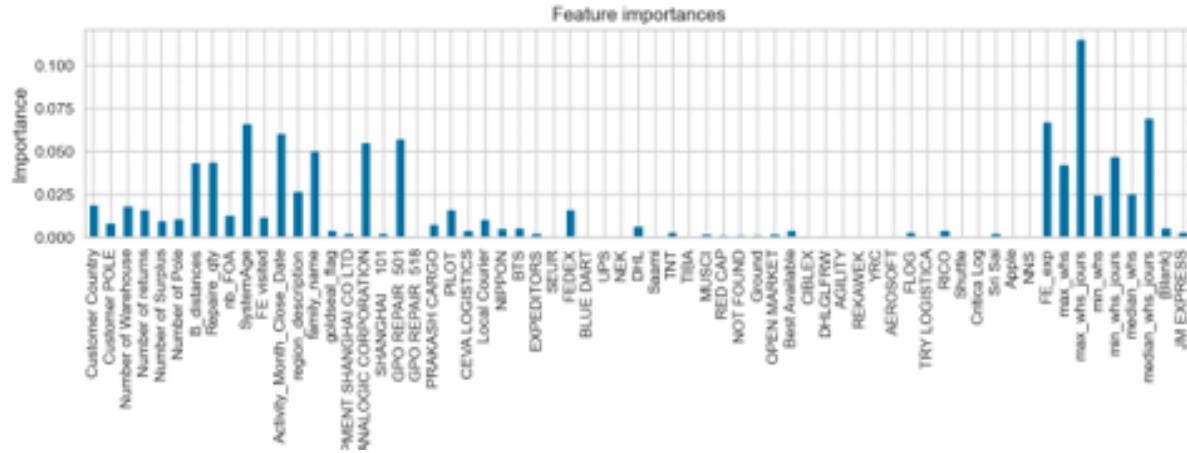


FIGURE 27 – Feature importances pour faire la prédiction

En examinant la distribution des features importantes, nous avons remarqué des différences dans le taux de FOA entre les différentes sociétés de transport. Cependant, par la suite, nous avons constaté une forte corrélation entre les livreurs et les régions où se trouvent les pièces. Il devient donc difficile de déterminer si les dommages aux pièces sont causés par le transport ou par d'autres facteurs dans la région.

En observant la distribution des features essentielles, nous avons noté des variations dans le taux de défaillance dès l'arrivée (FOA) selon les différentes sociétés de transport. Cependant, cette relation a été nuancée par la découverte d'une forte corrélation entre les sociétés de transport et les zones géographiques où se trouvent les pièces. Ainsi, il est devenu complexe d'attribuer les dommages aux pièces exclusivement au transport, car d'autres facteurs liés à la région peuvent également jouer un rôle significatif.

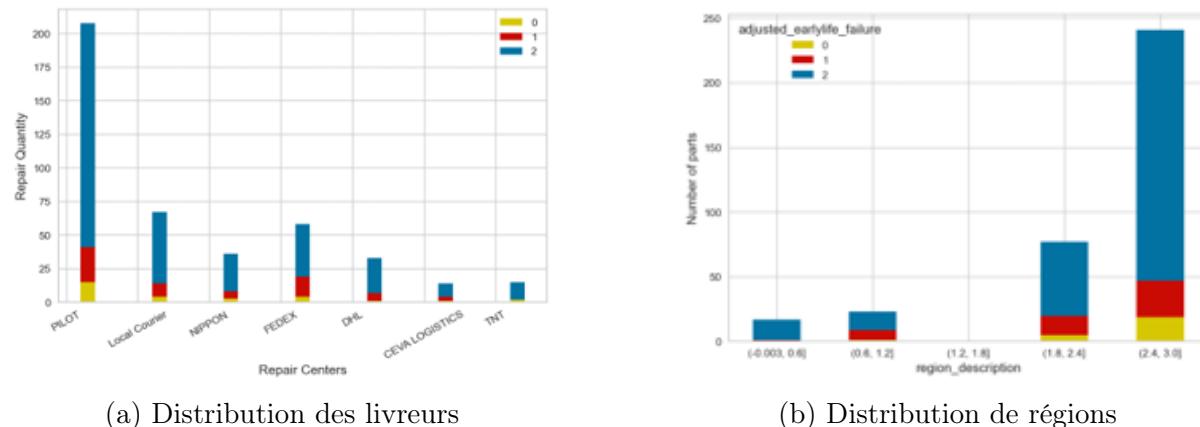


FIGURE 28 – Analyse par référence

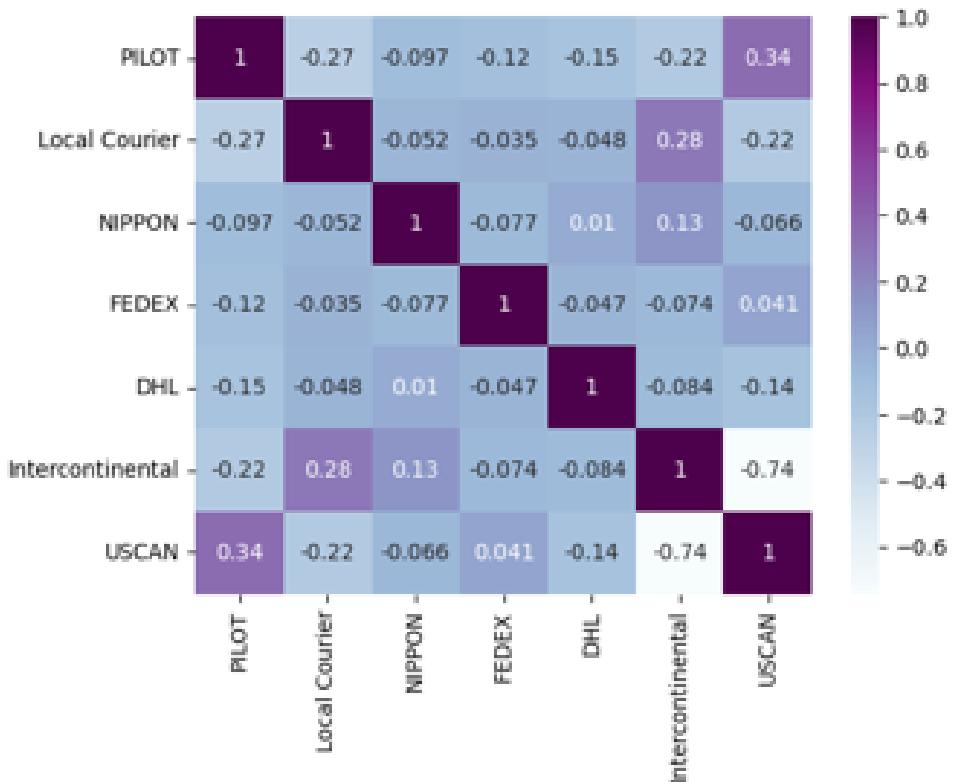


FIGURE 29 – Corrélation des livreurs et régions

4.8 Analyse par catégorie/composant

Néanmoins, les pièces de même type ou ayant les mêmes composants ont souvent des causes de défaillance similaires. Par conséquent, on a procédé à une segmentation des pièces pour les regrouper. Dans cette section, je vais présenter qu'est-ce qu'on a trouvé en regroupant les pièces.

On se concentre principalement sur l'analyse de ces trois catégories : Coil, Coldhead et Amplifier-Gradient, d'ailleurs, en filtrant par composant, on choisit aussi les pièces avec probe.



FIGURE 30 – Probe de Ultra sonore

4.9 Analyse thématique

Comme j'avais présenté précédemment, le taux de FOA dans cette région est corrélé avec le taux de FOA du centre de réparation dans cette région et le taux de FOA des livreurs dans cette région. La logique la plus probable est que les différences de qualité de réparation des pièces entre les centres de réparation entraînent des variations dans le taux de FOA de la région. C'est pourquoi nous souhaitons poursuivre avec une analyse thématique approfondie.

4.9.1 Analyse sur les Centres de Réparation

On a sélectionné plusieurs références dans la catégorie "Coil", collecté leurs informations sur les dommages et le centre de réparation par lequel elles sont passées avant les défaillances, et comparons la différence de qualité des pièces de réparation dans chaque centre de réparation.

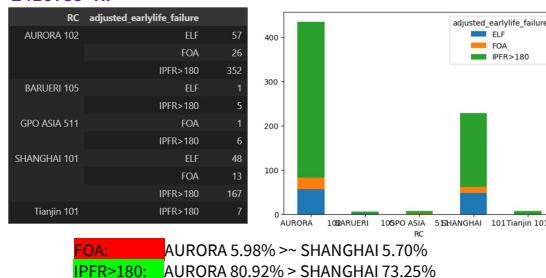
En faisant ça, on a constaté que parmi les pièces choisies qui sont pas mal consommées et réparées dans plusieurs centres de réparation, le centre de réparation Aurora aux États-Unis a un taux de FOA plus élevé que le centre de réparation à Shanghai. De même, la proportion de pièces réparées IPFR>180 (bonne qualité) est aussi plus élevée que Shanghai.

Analyse thématique- RC

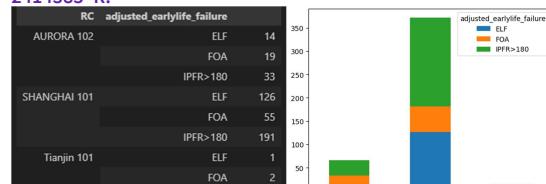
Suite à l'analyse de la catégorie Coil:

- Collecter toutes les données dans tableau ELF (RMA, liée avec l'événement de changer la pièce) depuis 2019
- Chercher les RMAs en forward, collecter les PO numbers liés avec cette pièce
- Retrouver 'Vendor name' dans 'repair history', joindre avec les failures types, calculer le taux et visualiser la portion.

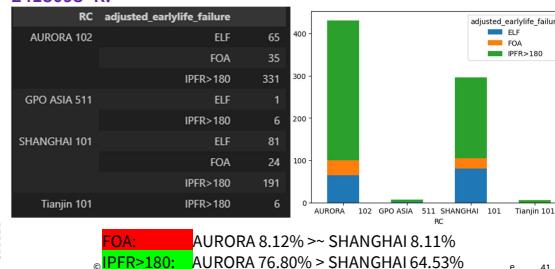
2416759-R:



2414383-R:



2418093-R:



 GE HealthCare

e. 41

FIGURE 31 – Comparaison des Centres de réparation

On a deux hypothèses pour ce phénomènes : La première est qu'il y a un problème avec le processus de diagnostic au centre de réparation Aurora, certains facteurs cachés qui peuvent causer FOA ne sont pas vérifiés. Il est également possible que les FEs dans

certaines régions n'ait pas assez d'expérience et qu'il y ait des erreurs d'installation, ce qui entraîne le retour de certaines pièces qui ne sont pas mauvaises en tant que FOA.

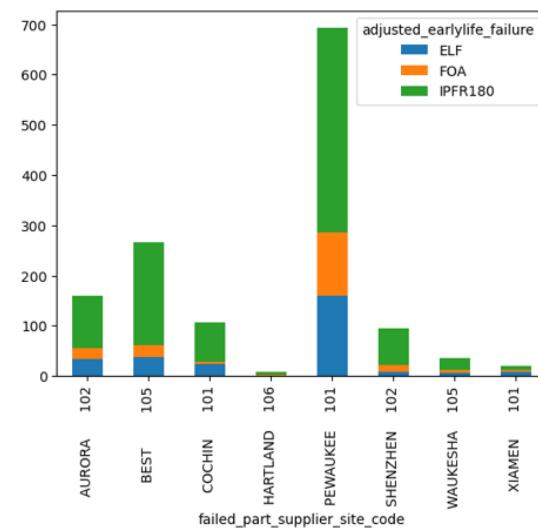
Personnellement, je préfère la première supposition, car la différence de taux de FOA est principalement reflétée dans une référence(ref.2414383-R), et la différence de taux de FOA des autres parties est de deux décimales, il est donc probable qu'Aurora ait des problèmes avec le processus lors de la réparation de cette pièce.

4.9.2 Analyse des pièces primées

En effet, comme j'avais dit avant, mon analyse a un grand contraint. On ne peut qu'analyser les pièces traçables et bien enregistrer chaque étape. Ce sont les pièces réparées au moins une fois, de cette manière, il pourra entrer dans notre système et être suivi. De cette façon, notre analyse n'est pas tout à fait fiable car on manque une grande partie de pièces. Pour compenser un peu, on voudrait aussi faire les analyses seulement pour les pièces primées. Inspiré par la comparaison des différents centres de réparations, on voudrait d'abord regarder les différences entre fournisseurs. Voici la distribution des pièces coil chez différents fournisseurs :

adjusted_earlylife_failure	ELF	FOA	IPFR180	ratio
failed_part_supplier_site_code				
AURORA 102	33.0	22.0	105.0	0.137500
BEST 105	37.0	24.0	206.0	0.089888
COCHIN 101	24.0	4.0	79.0	0.037383
HARTLAND 106	1.0	2.0	6.0	0.222222
PEWAUKEE 101	159.0	127.0	407.0	0.183261
SHENZHEN 102	9.0	13.0	72.0	0.138298
WAUKESHA 105	6.0	6.0	24.0	0.166667
XIAMEN 101	8.0	5.0	7.0	0.250000

(a) Comparaison des fournisseurs



(b) Visualisation de comparaison

FIGURE 32 – Analyse des pièces primées

A part de ça, on fait aussi la même chose qu'avant pour ces pièces nouvelles. Mais car les features sont beaucoup moins que les pièces réparées, le résultat de clustering et prédiction ne sont pas aussi bien qu'avant. Au lieu de faire la classification, on regarde directement les distribution de chaque features.

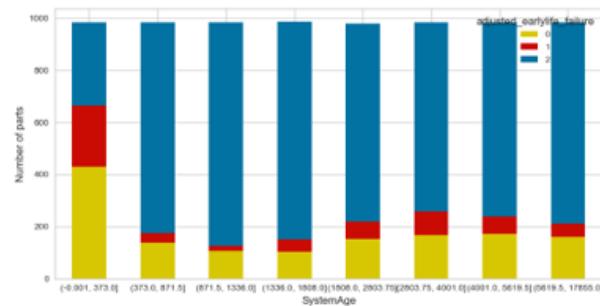
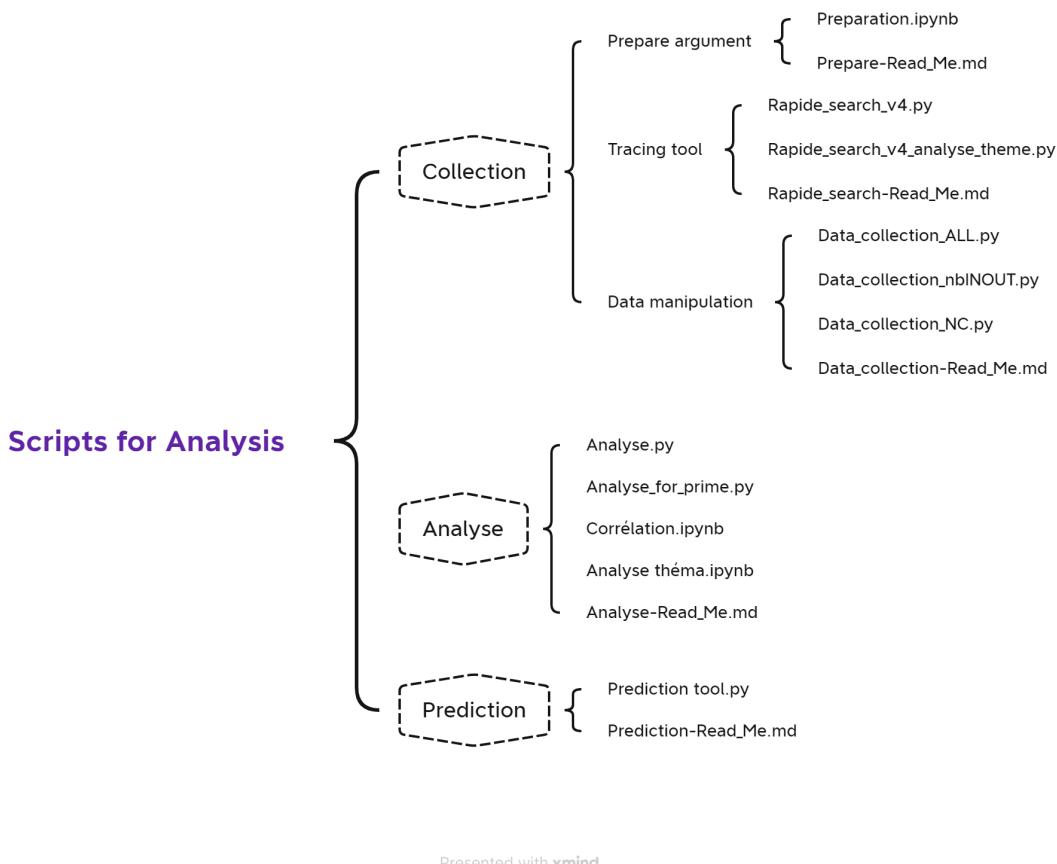


FIGURE 33 – Distribution de System Age pour les pièces primées

À partir de cette distribution, nous pouvons constater que pour ce genre des pièces, si elles sont installées sur un système plus jeune (< 1 an), la pièce est plus susceptible de se tomber en FOA. De plus, il est à noter que sur ce graphique, nous nous concentrons uniquement sur la distribution de FOA, la distribution d'ELF ne dit rien. Car c'est impossible d'avoir une pièce en servis plus de 1 an dans un système installé il y a moins un an. Quant à savoir pourquoi ce phénomène se produit, je ne l'ai pas encore compris..

5 Outils créés

En effet je n'ai pas assez de temps d'analyser toutes les catégories. Et puis il y a trop de façon pour regrouper les pièces, c'est impossible de tout analyser. Donc j'ai créé les outils automatiques pour le faire, comme ça on peut faire l'analyse soi-même même s'il ne sais pas trop les techniques informatiques.



Presented with xmind

FIGURE 34 – Scripts créés

5.1 Module de collection

La première module utile est la module de collection, ça pourrait collecter et calculer toutes les features et engendrer le fichier de toutes les données besoins pour l'analyse.

Sachant que notre logique de collecte de données est sophistiquée, nous ne pouvons donc pas automatiser entièrement la collecte de données. Nous utilisons donc le fichier Preparation.ipynb comme un pont entre différentes étapes.

Dans ce script intermédiaire, on peut

- unifier les labels. Étant donné que les données sur Service Suite divisent le type de défaillance des pièces en cinq classes (FOA, FOI, IPFR90, IPFR180 et IPFR>180), nous ne voulons pas prêter attention à autant de catégories. Nous divisons donc ici les pièces en trois catégories en fonction de la durée de vie des pièces et les stockons dans le fichier Excel.

- séparer les RMAs différents Pour les pièces FOA, on cherche les RMAs en backward et pour les pièces non FOA, nous les suivons en avant. Nous devons donc les séparer. Dans ce notebook, nous les stockons séparément dans deux fichiers Excel différents. Ensuite, vous devez exécuter le script Recherche rapide pour obtenir les fichiers de suivi.
- collecter PO number En fait on n'a pas de source pour récupérer directement les PO numbers, mais on peut les capter dans le fichier de tracing. La fonction dans ce fichier nous permet de faire ça.
- créer un lien entre RMA et PO

En stockant les fichiers suivis par recherche rapide dans un dossier appelé key_group, nous pouvons utiliser ce module pour obtenir et calculer les features dont nous avons besoin.

Pour la collection, juste suivre les instructions et il faut changer les colonnes à encoder si vous ajoutez les nouvelles colonnes.

5.2 Module d'analyse

Ce script regroupe toutes les fonctions liées à l'analyse, et est utilisé pour effectuer diverses analyses sur les scripts collectés. Après avoir entré le chemin où enregistre le résultat de Data_collection, on peut effectuer le clustering K-means, clustering DBSCAN et la prédiction de forêt aléatoire à ces données. On peut également afficher les distributions des différentes colonnes.

```
please set your threshold for feature selection(generally 0.05):0.05
We've selected these features: ['Repaire_qty' 'failed_part_classification' 'MKS INSTRUMENTS INC 107541'
 'GPO REPAIR 501' 'GPO REPAIR 518']
----- end of selection -----
Choose your analyse mode:
 1. K-means clustering
 2. DBSCAN clustering
 3. Prediction(Random Forest)
 4. Distribution of data
 5. Exit
 1
----- K-means Clustering -----
Do you want our suggestion about nb of cluster centers?■
```

FIGURE 35 – Interface du script

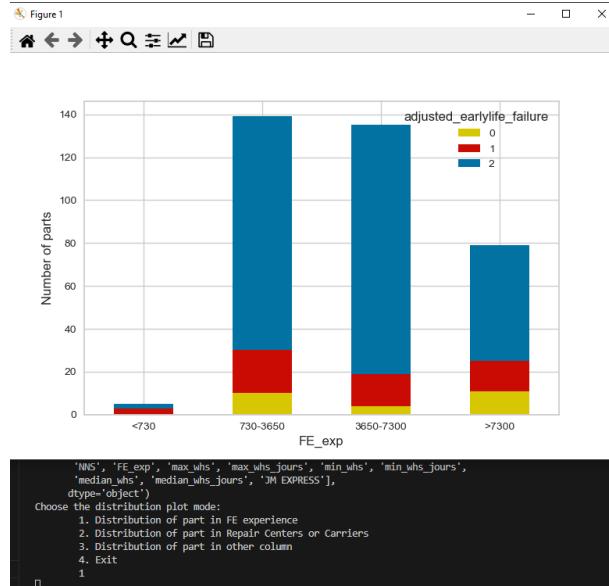


FIGURE 36 – Exemple d'utilisation pour afficher la distribution

5.3 Module de prédiction

Avant de définir le scénario d'utilisation, j'ai créé un outil de prédiction basé sur mon imagination des scénarios de travail FE. Ce pendant, il s'est avéré que mon imagination de la scène de travail des FEs était complètement différente de la réalité, de sorte que le scénario d'utilisation de cet outil est encore très vague. C'était aussi l'une des choses stupides que j'ai faites pendant mon stage. Mais je présente cet outil quand même, peut-être un jour je trouverais la situation d'utiliser ce script.

Cet outil est conçu pour prédire la durée de vie des pièces.

Pour les pièces qui ont été réparées et qui peuvent être tracées, nous devons fournir son dernier repair PO et les propriétés de cette pièce (par exemple, de quelle famille elle appartient), à la fin nous puissions obtenir une bonne précision de prédiction (précision de 90% pour séparer FOA et non FOA). Le résultat dépend aussi le regroupement des pièces.

Pour les pièces qui n'ont jamais été réparées, nous devons entrer leurs propriétés une par une comme les features. Dans ce cas là, la précision de prédiction n'est pas très élevée.

5.4 Documenter les modules

Jusqu'aux moments, j'ai plein de scripts à la main et personne ne sais pas comment les utiliser sauf moi [12]. Donc ça me donne envie de bien documenter ces fichiers. Dans la réunion hebdomadaire des stagiaires de notre équipe, j'ai été initié à l'utilisation de sphinx.

Sphinx [13] est une bibliothèque de documentation pour Python qui permet aux développeurs de générer facilement des documents de documentation clairs et complets pour leurs projets. En utilisant le format de marquage reStructuredText, Sphinx extrait automatiquement les commentaires docstrings du code source Python pour les intégrer dans la documentation générée, sous forme de HTML (y compris Aide HTML Windows), LaTeX (pour les versions imprimables PDF), ePub, Texinfo, pages de manuel ou texte brut. J'ai

choisi le format de HTML.

Avec Sphinx, la génération de la documentation devient un processus automatisé, offrant ainsi une meilleure cohérence et une mise à jour rapide de la documentation en cas de modifications dans le code source. La personnalisation est également possible grâce à la prise en charge de thèmes personnalisés et d'extensions, permettant d'ajouter des fonctionnalités supplémentaires à la documentation, telles que la coloration syntaxique, les diagrammes, les mathématiques, etc.

Grâce à sa facilité d'utilisation, Sphinx est devenu un choix incontournable pour les développeurs souhaitant offrir une documentation exhaustive et professionnelle à leurs utilisateurs et collaborateurs. Il y a beaucoup de bibliothèques très connus qui utilisent sphinx pour générer la documentation par exemple numpy, pandas et scikit-learn.

Pour faciliter la rédaction des commentaires, j'ai choisi d'utiliser l'extension autodoc de VScode, qui m'a permis d'écrire des docstrings pour chaque fonction. Cette extension peut engendrer la forme de docstrings automatiquement, ça m'aide pour rédiger les commentaires.

```
def get_coordinates(country,city):
    """Obtenir la position de ville, latitude et longitude

    Parameters
    -----
    country : string
        nom du pays
    city : string
        nom de la ville

    Returns
    -----
    float
        latitude et longitude
    """

    geolocator = Nominatim(user_agent='xin.yan@ge.com')
    if city=='nan':
        location = geolocator.geocode(f'{country}')
    else:
        location = geolocator.geocode(f'{city} {country}')
    if location!=None:
        return location.latitude, location.longitude
    else:
        print('echec', country,city)
        return None, None
```

FIGURE 37 – Exemplaire de docstring

Après avoir terminé la rédaction des docstrings, j'ai procédé à l'installation de Sphinx sur mon environnement[14]. Une fois Sphinx installé, j'ai utilisé la ligne de commande pour générer la documentation au format HTML. La documentation générée par Sphinx était

bien structurée et facile à naviguer. Les docstrings que j'ai rédigés ont été automatiquement intégrés à la documentation, permettant aux utilisateurs de comprendre facilement le fonctionnement de chaque fonction.

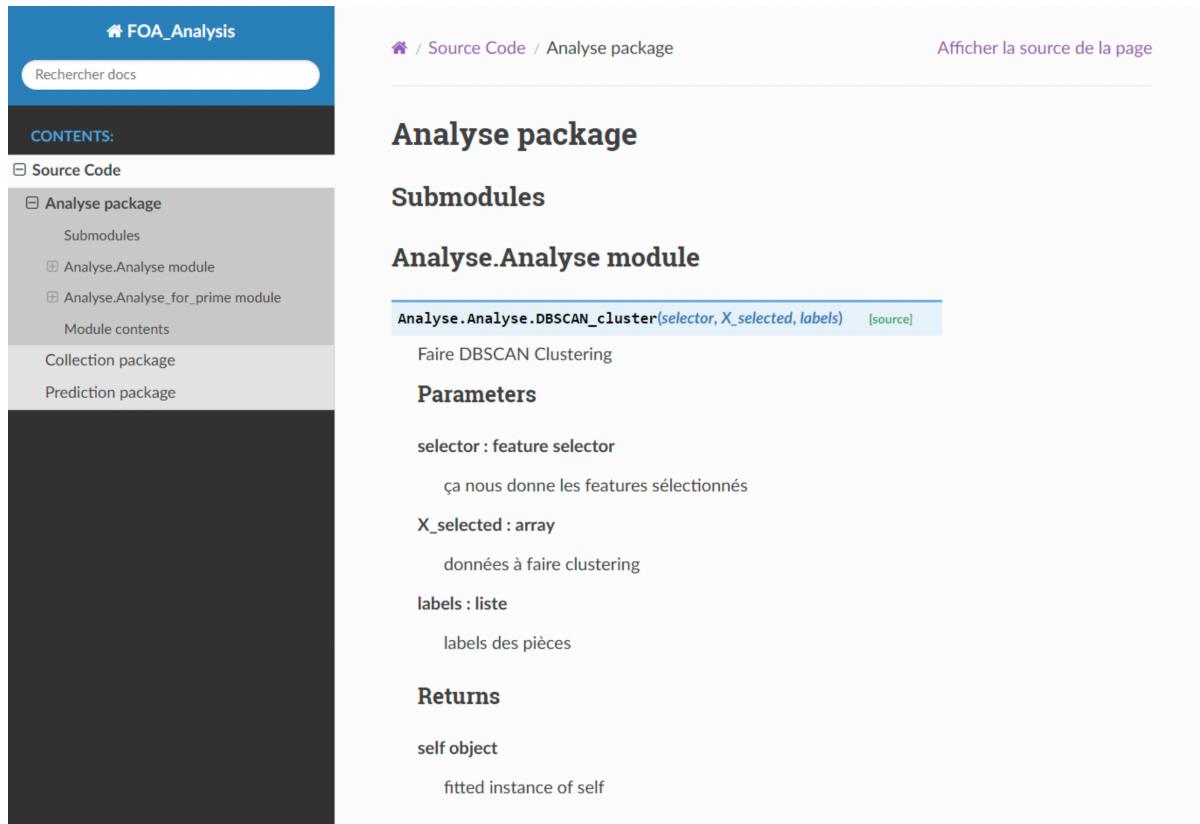
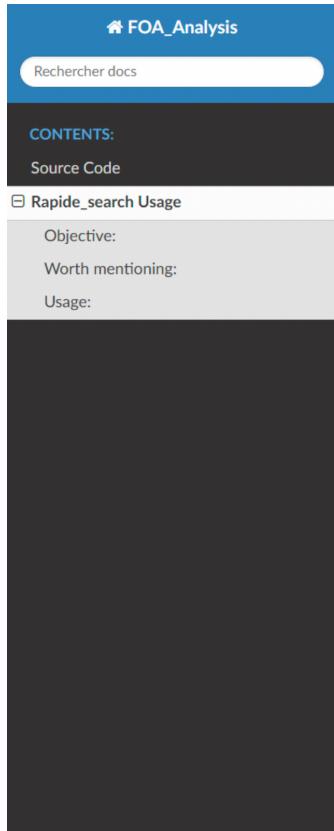


FIGURE 38 – Explication des fonctions dans module

D'ailleurs, non seulement on peut documenter les scripts comme modules, mais aussi écrire ***Read_Me*** pour chaque module. Afin de faciliter l'adoption de mon projet par d'autres ou de permettre à d'autres stagiaires de poursuivre mon travail, j'ai documenté chaque module en détaillant son objectif, sa méthodologie d'utilisation et les innovations qu'il apporte comme suivant :



Rapide_search Usage

This script is adapted from *Module_tracing* tool of **Abdelhamid**

Objective:

The purpose of this script is to speed up the tracing of parts by copying the data on the FBI to local.

Worth mentioning:

Because the search time of the tracking tool will become quite long (2-3h) when encountering prime parts, this script adds a timeout interrupt function. If the tracking time of a part is longer than one minute, the tracking of this part will be abandoned automatically.

```

def timeout(timeout):
    def deco(func):
        @functools.wraps(func)
        def wrapper(*args, **kwargs):
            res = [Exception('function [%s] timeout [%s seconds] exceeded!' % (func.__name__, timeout))]
            def newFunc():
                try:
                    res[0] = func(*args, **kwargs)
                except Exception as je:
                    print ('error starting thread')
                    raise je
                ret = res[0]
                if isinstance(ret, BaseException):

```

FIGURE 39 – Exemplaire de Read_me pour chaque module

6 Conclusions

Avant de donner les conclusions, il est important de préciser que toutes nos conclusions sont basé sur les phénomènes observés et obtenues par raisonnement inductif, ça évolue en fonction de données. En changeant des données, les conclusions pourraient ne plus être valables. Voici les processus de raisonnement inductif[15] :

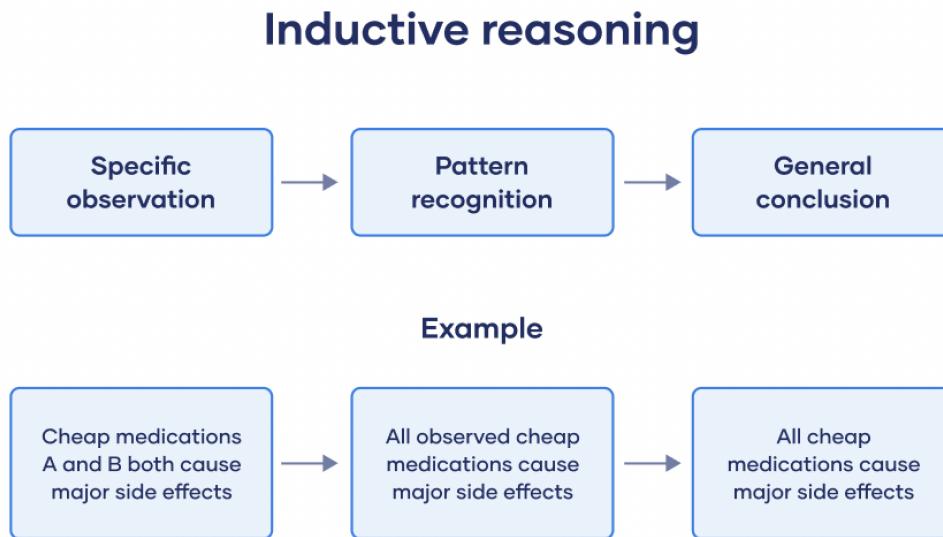


FIGURE 40 – Raisonnement Inductif

Dans notre analyse, on a observé que les phénomènes et les conclusions varient d'un groupe de pièces à l'autre. Pour rendre compte de cette diversité, on a divisé nos conclusions en deux parties distinctes. Une partie de conclusions générales, ce sont les phénomènes présentées dans toutes les segmentations qu'on a analysé, une autre partie de conclusions individuelles.

6.1 Conclusions générales

- **La distance parcourue longue n'entraîne pas FOA.** Dans tous nos analyse, l'impact de distance est assez faible. Que ce soit en examinant la répartition des pièces à différentes distances ou en regardant l'importance des features lors de la classification, nous n'avons pas trouvé de tendance particulière.

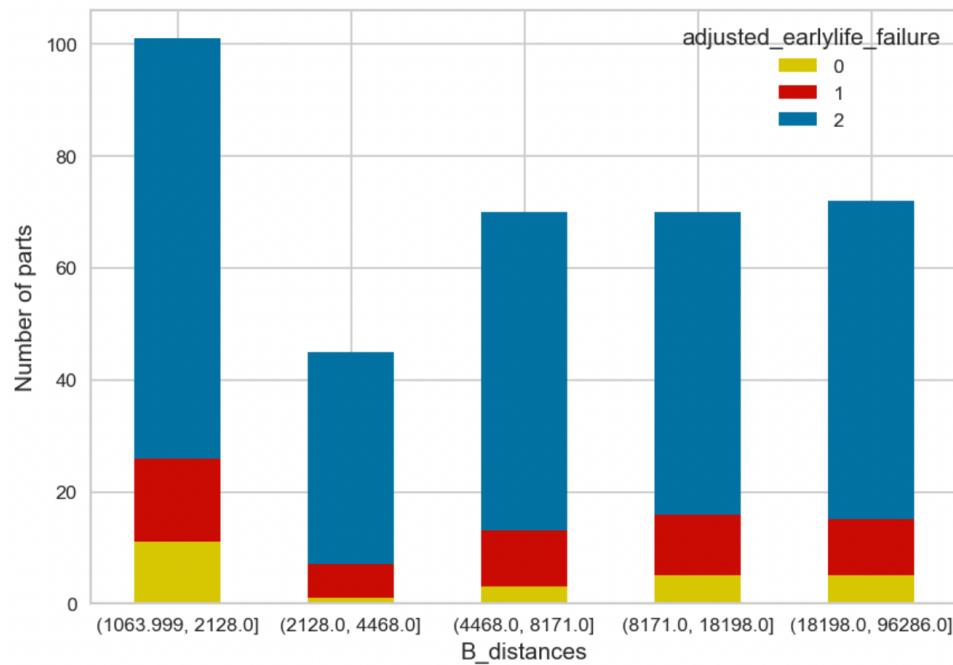


FIGURE 41 – Distribution de distance

- **Système jeune ou vieux entraîne FOA.** En effet on a constaté dans de nombreux types de pièces que le taux de FOA des pièces est plus élevé quand le système est très vieux ou très jeune. Par exemple,

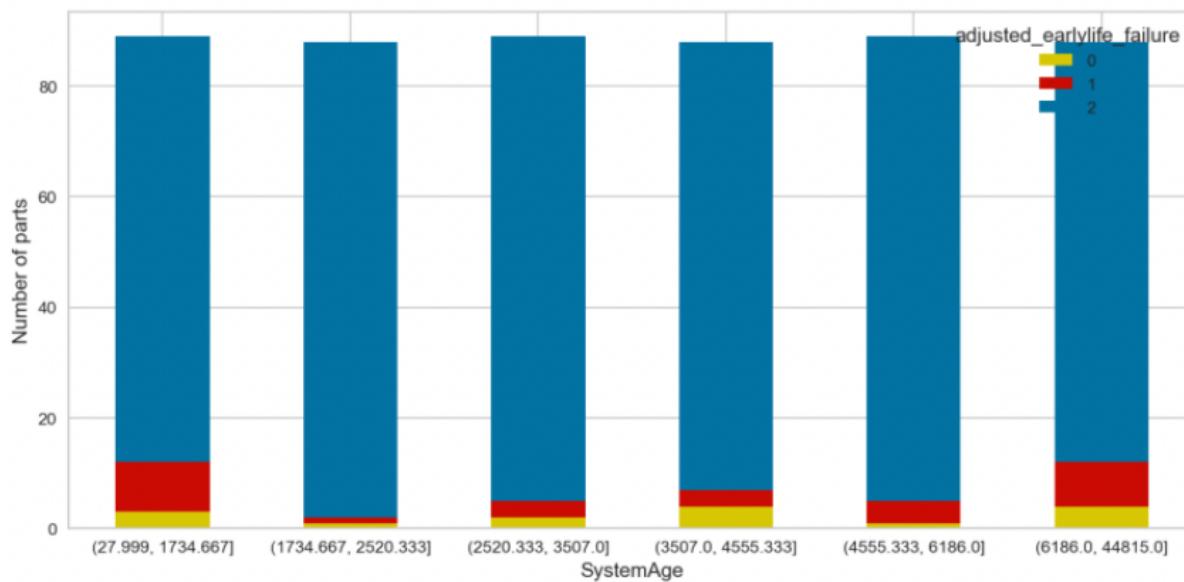


FIGURE 42 – Distribution de System Age

- **Séjour longue dans l'entrepôt n'entraîne pas FOA, mais flux plus intense entraîne FOA.**

On a constaté un phénomène bizarre dans plusieurs segmentations des pièces, c'est à dire une pièce reste plus long temps à l'étagère, elle a une probabilité plus élevée

de FOA. Intuitivement, les pièces restent longtemps à l'étagère sont plus possibles tomber en panne, mais les données sont différentes de ce que nous pensons. Pourquoi c'est comme ça ?

Pour creuser ce problème, je me suis dit, peut-être les pièces restent moins longtemps dans l'entrepôt ce sont les pièces transaction plusieurs fois ? Donc j'ajoute encore une fois la distribution de nombre d'entrée et sortie de Warehouse, mais je n'ai pas vu de tendance significative.

J'en peux plus, du coup je demande aide d'un autre data scientiste. D'après lui, Comme on avait analysé, lorsque les pièces restent moins de temps dans l'entrepôt, le taux de FOA est plus élevé. Cela pourrait être dû au fait que séjour courte représente un flux plus intense de la chaîne d'approvisionnement des pièces.

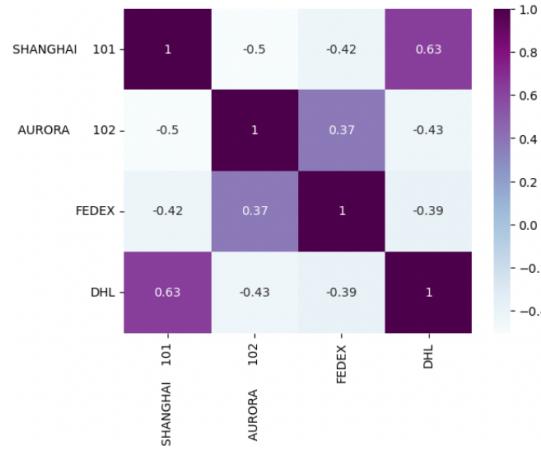
6.2 Conclusions individuelles :

Sauf les conclusions générales, pour certaines pièces, on peut également tirer les conclusions inter individuelles fiables.

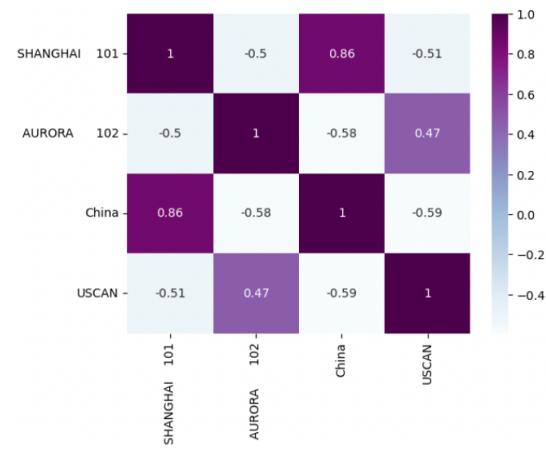
- **Coil** : Nous avons analysé un échantillon de toutes les pièces de Coil défectueuses enregistrées sur tableau ELF dans Service Suite depuis 2022, en se concentrant sur les parties qu'on peut suivre avec précision.

- **Taux de FOA varie en fonction de Centre de réparation.**

En analysant les pièces de cette échantillon, on trouve que « region description », « repair center » sont tous les features importants quand on faire la classification. De plus, le taux de FOA change aussi en fonction des livreurs. Mais en fait, ces trois critères sont très corrélés. Ils s'influencent mutuellement, donc c'est difficile de dire exactement quel facteur entraîne les différences de taux de FOA. La chaîne logique le plus possible est : différente qualité de cause la différence FOA des livreurs et ça entraîne la différence FOA dans chaque région.



(a) Corrélation-régions et livreurs



(b) Corrélation-région et Centre de répare

FIGURE 43 – Analyse corrélation

Pour confirmer notre conclusion, on fait l'analyse de différents Centre de réparation. On a choisi les pièces réparées en même temps à l'Aurora des États Unis et à Shanghai : Référence 2414383-R, Référence 2416759-R et Référence

2418093-R, ce sont toutes les pièces de catégorie Coil. Et puis on collecte toutes les données de défaillance depuis 2019 et comparer les taux de défaillance.

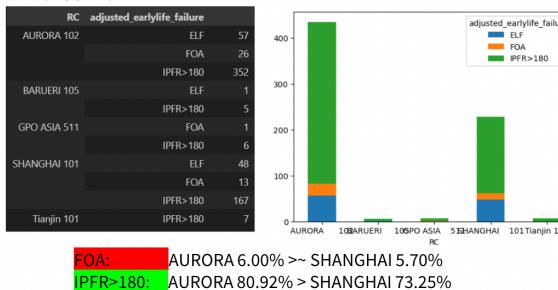
On trouve que le taux de FOA est toujours plus élevé dans l'Aurora, surtout pour la référence 2414383-R. Mais taux de très bonne qualité dans l'Aurora est aussi plus élevé qu'à Shanghai. En fait seulement à partir les données, on ne peut pas tout comprendre les raisons de cette différence. Peut-être il faut aller personnellement à ces centres et trouver les différences de processus.

Analyse thématique- RC

Suite à l'analyse de la catégorie Coil:

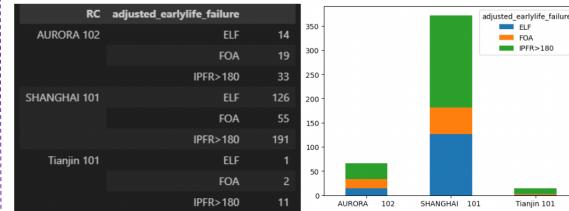
- Collecter toutes les données dans tableau ELF (RMA, liée avec l'événement de changer la pièce) depuis 2019
- Chercher les RMAs en forward, collecter les PO numbers liés avec cette pièces
- Retrouver 'Vendor name' dans 'repair history', joindre avec les failures types, calculer le taux et visualiser la portion.

2416759-R:

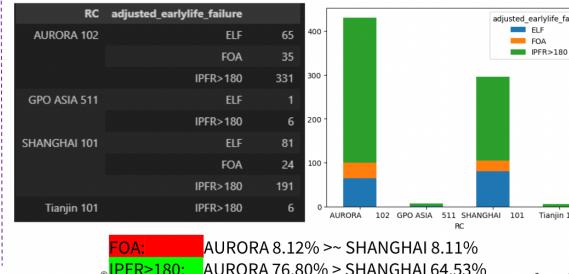


 GE HealthCare

2414383-R:



2418093-R:



e. 41

FIGURE 44 – Différence entre Aurora et Shanghai

— **Amplifier RF :** On a analysé un échantillon de toutes les pièces d'Amplifier RF défectueuses enregistrées sur tableau ELF dans Service Suite depuis 2019, en se concentrant sur les parties qu'on peut suivre avec précision.

- **Taux de FOA est plus élevé dans certaines familles.**

Parmi les pièces que on a étudiées, ces familles d'Amplifier-RF ont contribué à la majeure partie des pièces FOA, ils sont : 3.0T Signa Architect, 3.0T Signa Excite, 3.0T Signa HDx, 3.0T Signa HDxT et 3.0T Signa PET/MR.

— **Probe :** On a analysé un échantillon de pièces contient le composant *Probe* et cassé depuis 2022, en se concentrant sur les parties qu'on peut suivre avec précision. En regardant en même temps les features importants et les distributions des features, on n'a pas trouvé les tendances spéciales pour les probes. Ce sont plutôt les conclusions vues en toutes les pièces :

- **Taux de FOA est plus élevé dans les systèmes plus jeunes.**

À partir de ce graphe 45, nous pouvons constater que l'âge du système est un facteur très important qui influence les performances de classification. Donc on affiche aussi la distribution d'âge du système sur graphe 46

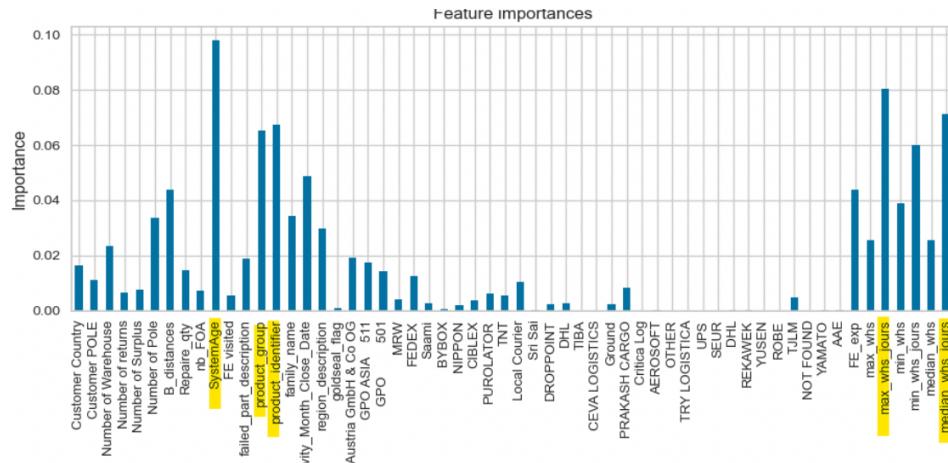


FIGURE 45 – Feature Importance

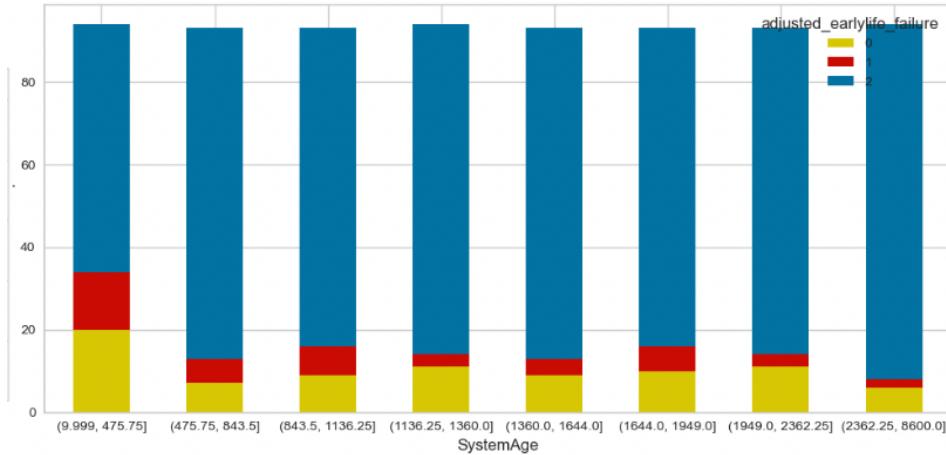


FIGURE 46 – Distribution de System Age

Ça pourrait parce que les systèmes plus jeunes peuvent comporter des pièces nouvellement fabriquées et installées, ce qui signifie qu'elles n'ont pas encore eu le temps de subir les épreuves du temps ou de l'usage. Les défauts de fabrication ou de conception initiaux peuvent se manifester plus rapidement dans ces pièces.

- **Taux de FOA plus élevé quand les pièces tournent plus vite.** Comme toutes les autres pièces, on a aussi constaté que quand les pièces tournent vite(rester moins long temps dans l'entrepôt), le taux de FOA est plus élevé. Ça pourrait parce qu'une forte demande de pièces entraîne une pression accrue sur les centres de réparation, ce qui a un impact sur la qualité.

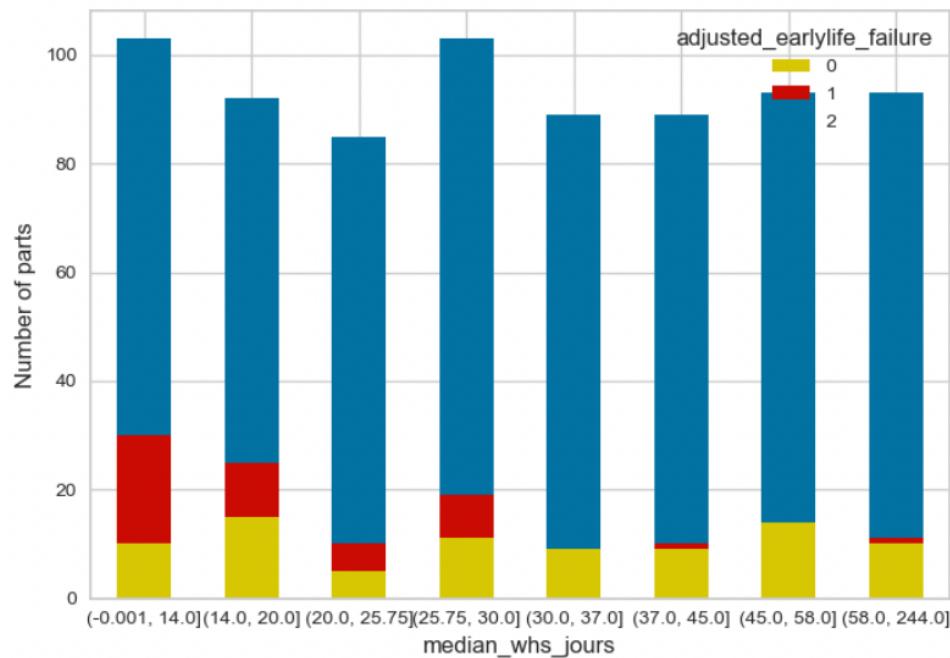


FIGURE 47 – Distribution de séjour dans les entrepôts

- **Coldhead :** On a analysé un échantillon de pièces contient le composant *Coldhead* et cassé depuis 2022, en se concentrant sur les parties qu'on peut suivre avec précision. Les conclusions générales sont bien présentées dans cette catégorie :
 - Taux de FOA plus élevé quand les pièces tournent plus vite

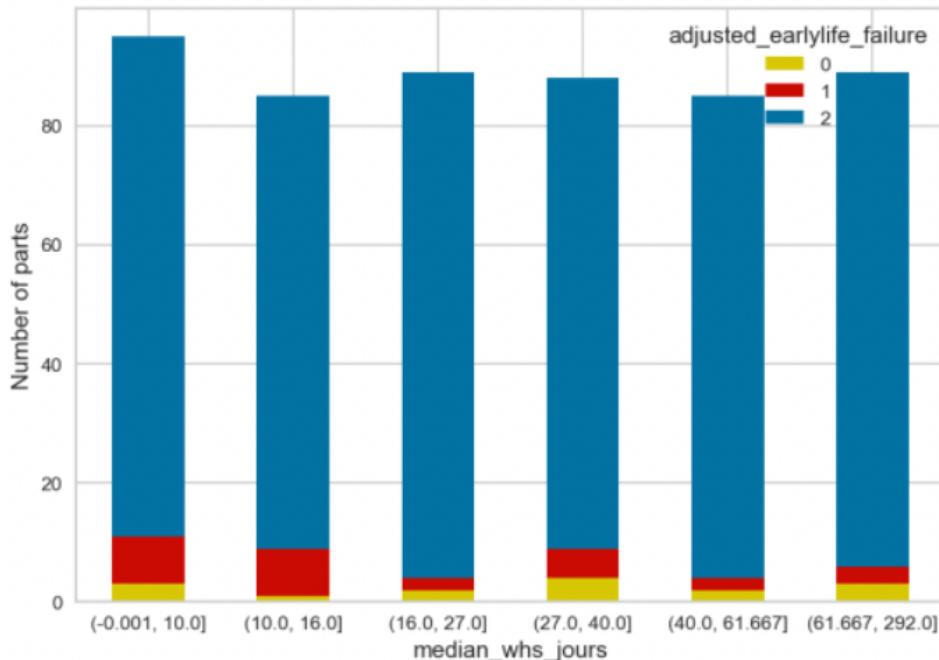


FIGURE 48 – Distribution de séjour dans les entrepôts

- Taux de FOA plus élevé quand elle installé dans un système très

jeune ou très vieux

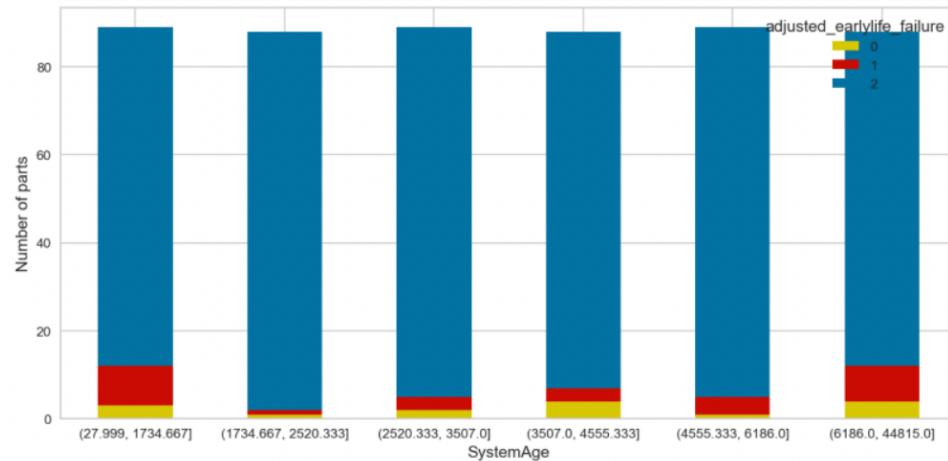


FIGURE 49 – Distribution de SystemAge pour coldhead

- sauf les conclusions générales, on a aussi trouvé une conclusion présente spécialement dans cette segmentation des pièces comme montré dans le graphe suivant.

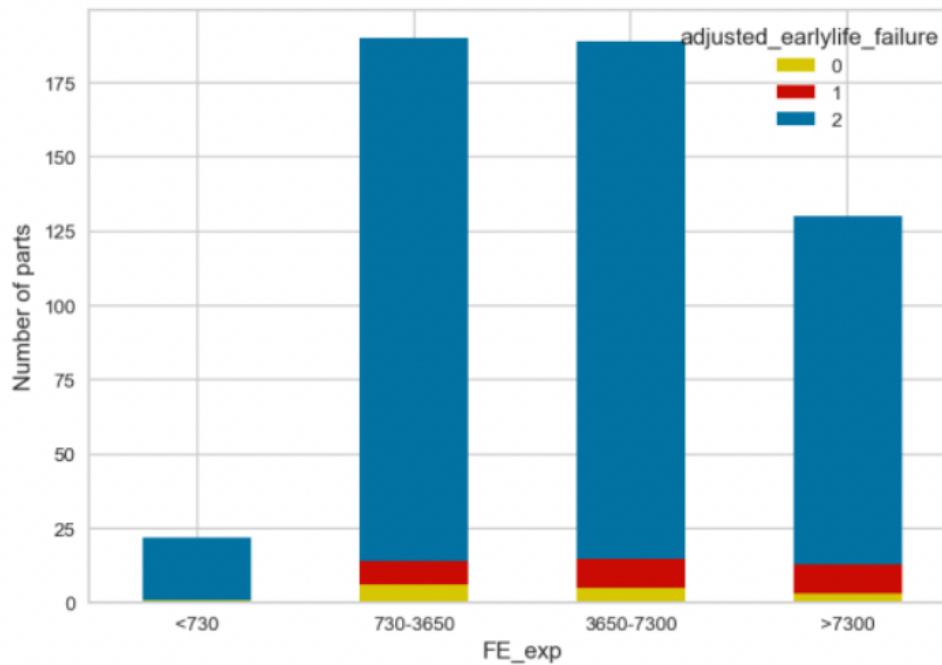


FIGURE 50 – Distribution d'expérience des FEs

En séparant tous les FE par ses jours d'embauche en quatre parties : FE d'expérience de moins deux ans, FE d'expérience de 2-10 ans, FE d'expérience de 10-20 ans et FE d'expérience plus de 20 ans. On trouve que **plus un ingénieur(FE) est employé pendant une longue période, plus le taux de FOA des pièces qu'il installe est élevé.**

- Référence-2351573-R :** On a analysé un échantillon de pièces Référence 2351573-R et cassé depuis 2019, en se concentrant sur les parties qu'on peut suivre avec

précision. C'est en analysant les informations sur cette pièce que on a trouvé les raisons pour laquelle séjour courte dans l'entrepôt entraîne FOA. Les conclusions générales sont aussi présentées pour cette pièce.

- Impact de séjour dans l'entrepôt est le plus évident ici :

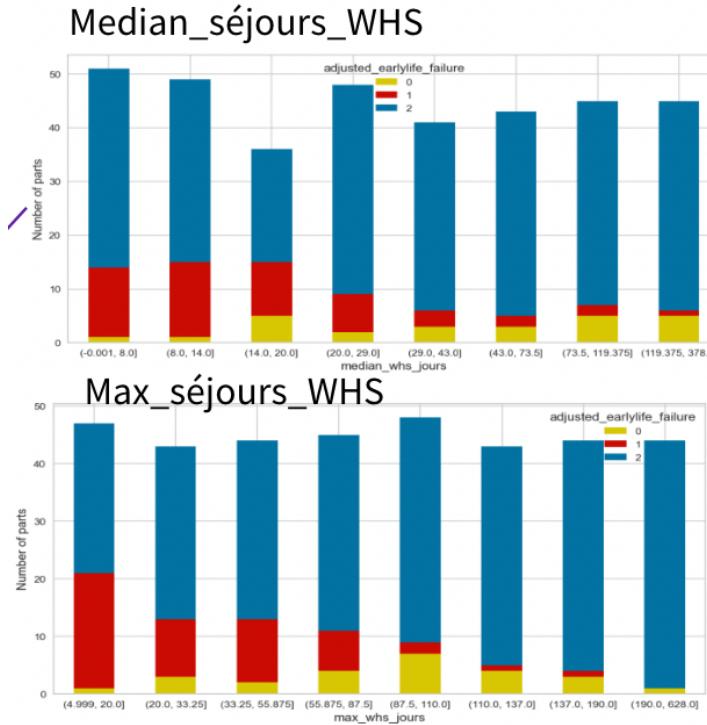


FIGURE 51 – Impact de séjour dans WHS

Pour analyse de cette partie, veuillez consulter section 6.1 pour plus de détails.

- Taux de FOA est plus élevé pour les pièces réparées plusieurs fois
Quand on classe les pièces de cette référence, on trouve *repair quantity* est un des features importants. Donc on regarde sa distribution :

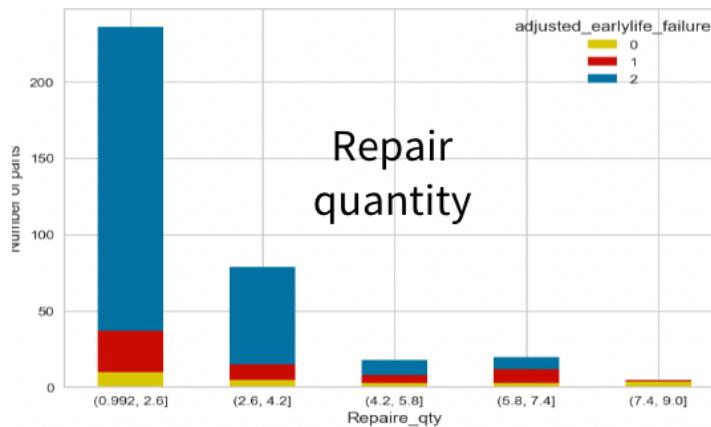


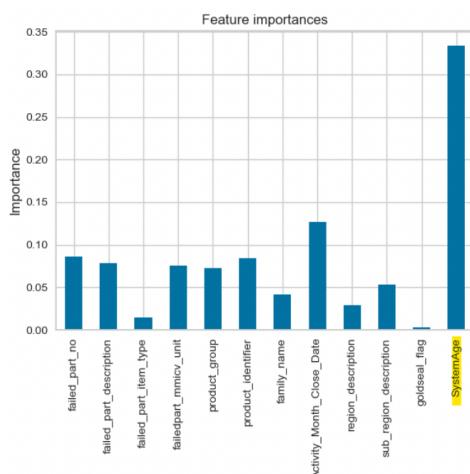
FIGURE 52 – Distribution de Repair Quantity

Évidemment, à mesure que le nombre de réparations des pièces augmente, le taux de FOA augmente. Mais ce phénomène est seulement constaté sur

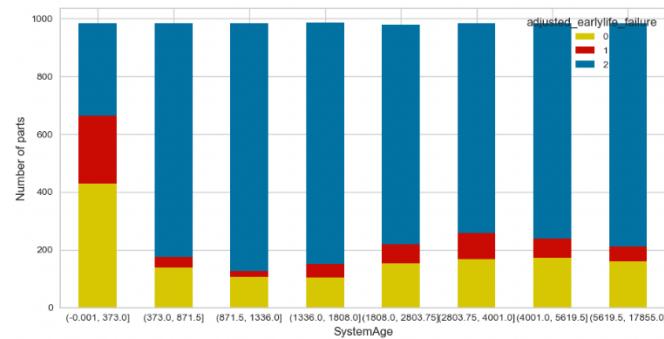
quelques références, pas toutes les segmentations car tous les groupes de pièces ne comptent pas un grand nombre de pièces ayant été réparées à de multiples reprises, et même s'il y en a, leur nombre est très limité, ce qui ne suffit pas à soutenir notre conclusion.

- **Pièces nouvelles :** En sachant que notre analyse est limité par les pièces réparées au moins une fois en raison des contraints de traçabilité, on voudrait aussi prendre en compte les pièces nouvelles. Cette fois-ci, on a collecté toutes les données pour les pièces **Prime** de catégorie Coil.

Pour les pièces prime, d'abord on fait la prédiction en utilisant une partie de features et obtient un accuracy de 76% à peu près. Comme on a vu dans l'image suivant, l'âge du système est le feature le plus important. En regardant la distribution, on trouve que **les pièces installées dans les systèmes plus jeunes ont le taux de FOA plus élevé.**



(a) Feature importance pour les pièces primes



(b) La distribution d'âge du système-pièces prime

FIGURE 53 – Analyse des pièces primes

Il n'y a plus tendance observé sauf la corrélation d'âge du système. En regardant d'autres distributions, les taux de FOA sont équilibre dans chaque intervalle.

7 Conseils pour la future

7.1 Constraints des données

Comme on avait dit depuis le début, toutes les pièces ne sont pas traçable et on limite des pièces traçable avec précision, ce sont les pièces réparé au moins une fois et bien enregistré et mis à jour chaque étape. Donc, quelles que soient les analyses que nous effectuons, elles ne concernent qu'une petite partie de toutes les pièces. Afin de rendre nos analyses plus précises et de découvrir davantage de conclusions possibles, notre première étape d'amélioration doit être de comprendre la raison pour laquelle certaines pièces ne peuvent pas être suivies, et de tenter de l'améliorer.

Au cours de tout notre processus de collecte de données, les étapes suivantes ont conduit à la perte de données sur les pièces :

— Cherche de RMA

Qu'il s'agisse de pièces FOA ou non-FOA, nous devons suivre les pièces selon les RMAs. Mais tous les RMAs ne sont pas correctement enregistrés. Certaines RMA de pièces endommagées sont nulles et certains enregistrements RMA ne peuvent pas être tracé (par exemple, l'enregistrement de deux RMA en même temps). Cette étape nous a fait perdre environ 10 % des pièces.

— Cherche de repair PO.

Pour les pièces qui ne sont pas FOA, afin de suivre les mesures qu'il a prises avant cette installation, on doit chercher RMA en forward et trouver le dernier repair PO. Cela perd beaucoup de pièces qui n'ont pas été réparées ou qui sont jetées directement après avoir été cassées. Le taux de perte de données de cette étape est d'environ 50%, ce qui changera avec différentes parties.

Au contraire, pour les pièces FOA, bien que nous n'ayons pas besoin de chercher les repair POs, mais afin d'assurer l'équité de l'analyse, nous ne conservons aussi que les pièces qui ont été réparées au moins une fois.

— Trace de repair PO.

Après avoir obtenu les repair POs, on doit tracer l'histoire de cette pièce. Comme on le voit en backward, on est sûr que ces pièces sont toutes consommées, mais dans les fichiers de tracing, il y a des pièces de *Part Status Not consumed*. Comme ça on est sûr que ces fichiers ne sont pas du tout fiables, donc on retire ces données. En effet, pourquoi il y en a des pièces qu'on est sûr qu'ils sont consommées mais marqué comme ***Not consumed***. Pour répondre à cette question, j'ai constaté que toutes ces parties qui suivent les interruptions auront les mots ***RETURNED TO VENDOR*** dans la colonne des commentaires. C'est pourquoi nous ne pouvons pas suivre correctement ces pièces.

On a perdu environ 10% des pièces à cette étape.

— Nettoyage des données.

Afin d'assurer l'exactitude de nos données, nous les avons nettoyées après les avoir collectées. Peut-être en raison de mal-enseignement des données, il y a des erreurs évidentes dans certaines données. Par exemple, la date d'installation du système est une date future. C'est pas possible, donc j'ai supprimé les données comme ça.

Dans cette étape, on a perdu environ 5% de données.

Donc le plus grand problème est que l'on a perdu beaucoup de pièces nouvelles ou bien rentré au fournisseur. C'est parce que nous ne pouvons pas suivre le processus de la sortie de l'usine à la première installation de pièces. Nous commençons à partir de l'incident de

dommages et suivre la pièce par un argument RMA en backward. Cette méthode de suivi n'est pas cent pourcent fiable, mais prend également beaucoup de temps. Par conséquent, on regarde qu'une petite partie des pièces et obtient les conclusion pas très fiable.

Avant que les données ne soient solides, toutes les conclusions obtenues ne sont pas très fiables, mais ce n'est pas un problème que je peux résoudre, c'est le problème de l'ensemble des données GE. Peut-être qu'un jour, nous résoudrons ce problème et trouverons quelque chose de nouveau lorsque nous regarderons en arrière.

7.2 Comment retirer les contraints

En fait parmi les stagiaires on discute souvent pourquoi il n'y a pas une base de données unique qui contient toutes les information qu'on a besoins pour le transport des pièces. Au contraire, on doit les collecter de différents bases de données. Il faut demander plusieurs autorisations, communiquer avec plusieurs collègues pour savoir comment utiliser ces données, télécharger plusieurs gros fichier et les mettre ensemble. Franchement c'est un peu ennuyeux et un peu une perte de temps. Mais est-il vraiment difficile de les unifier dans une seule base de données.. ? Je ne crois pas.

J'ai trouvé deux méthodes utilisés par les autres entreprises pour gérer les pièces ou les marchandises.

7.2.1 RFID Technologie

La technologie Radio Frequency Identification (RFID)[16] est un système de suivi automatique d'objets ou de personnes grâce à des étiquettes RFID contenant des informations stockées. Ces étiquettes sont dotées d'une puce électronique et d'une antenne. Lorsqu'un lecteur RFID émet un signal radiofréquence, il alimente passivement la puce RFID, qui répond en émettant ses données. Cette technologie est largement utilisée dans divers secteurs, notamment la gestion des stocks, le suivi des actifs, le contrôle d'accès, et même dans les passeports électroniques.

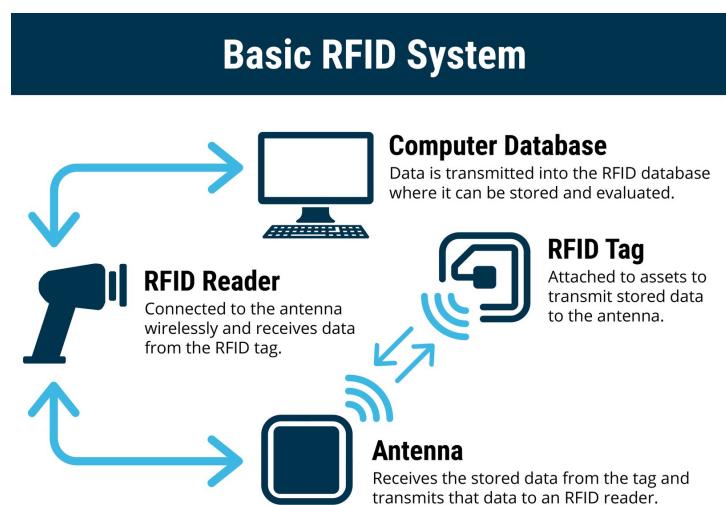


FIGURE 54 – Présentation de RFID[17]

Cette technologie est de plus en plus populaire, voici les statistiques de marketing d'après Statista[18] :

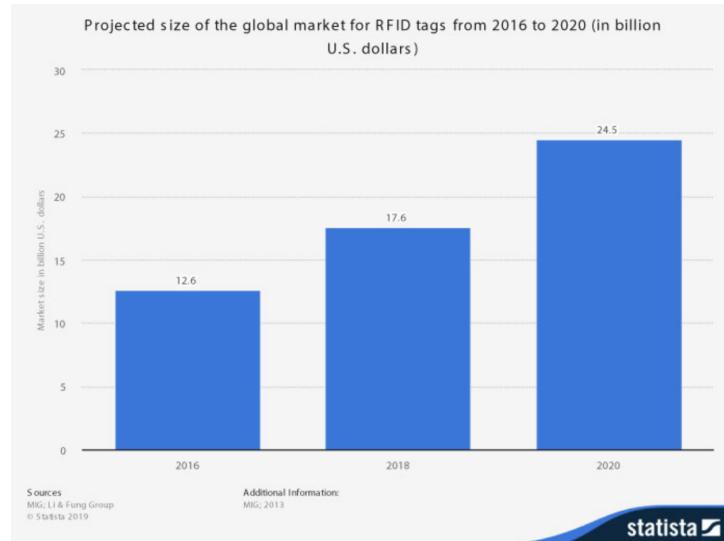


FIGURE 55 – Taille de marketing RFID de 2016-2020

Par exemple, Amazon et Zara utilisent étiquettes RFID pour traçer leurs marchandises depuis long temps[19].

- **Avantages :** La RFID offre de nombreux avantages, notamment l'automatisation des processus, l'efficacité de la lecture à distance, la sécurité des données grâce à la possibilité de cryptage, et la traçabilité des produits tout au long de la chaîne d'approvisionnement. C'est pour ça, il est utilisé par beaucoup d'entreprises surtout dans l'hôpital ou système médicale.
- **Inconvénients :** La question de savoir si la RFID aura un impact négatif sur les dispositifs médicaux nécessite encore des expériences et une observation à long terme. Une fois la RFID utilisée pour suivre les pièces, son impact possible sur l'équipement doit être pris en compte lors de la recherche et du développement. Concernant ça, Food and Drug Administration (FDA) a élaboré ce document d'orientation (que l'on peut trouver dans les références[20]) pour aider l'industrie et le personnel à identifier et à traiter de manière appropriée les considérations spécifiques liées à l'incorporation et à l'intégration de la technologie sans fil radiofréquence (RF) dans les dispositifs médicaux.

— **Processus d'utilisation**

Au lieu de peser les pièces automatiquement les pièces, on peut enregistrer le poids, la date de production, le nom de famille et d'autres informations des pièces dans l'étiquette électronique lorsque les pièces quittent l'usine. Lorsque les pièces passent par l'entrepôt ou le centre de réparation, les informations de transport des pièces sont enregistrées via lecteur de RFID. Voici les processus plus exactement :

- Lors de la fabrication des composants, attachez des étiquettes RFID à chaque composant et stockez des informations pertinentes, telles que la date de fabrication et le numéro de lot, dans la mémoire de l'étiquette.
- Utilisez un dispositif de lecture RFID pour scanner l'étiquette RFID de chaque composant afin de suivre son état de transport et sa localisation. Ces scans peuvent être effectués tout au long de la chaîne d'approvisionnement, de la livraison du fournisseur à l'usine de fabrication, à la réception, à la production et à l'expédition.
- Téléchargez les données RFID scannées dans un système de gestion des données

dédié pour une analyse et un suivi ultérieurs.

En fin de compte, la technologie RFID a révolutionné la gestion de l'information et de la logistique dans de nombreux domaines. Elle continue de s'étendre et de s'améliorer pour répondre aux besoins croissants de suivi et de gestion des données. Son utilisation efficace peut apporter des avantages considérables en termes d'efficacité opérationnelle, de réduction des erreurs humaines et de meilleure visibilité des opérations.

7.2.2 QR code ou bar code

Mais RFID est coûteux et peut potentiellement perturber les dispositifs médicaux. On peut simplement coller des codes QR ou des codes-barres sur les pièces pour simplifier l'enregistrement des informations des pièces à leur sortie d'usine. Par rapport à la RFID, la génération et l'impression de codes QR ou de codes-barres sont moins coûteuses et ne nécessitent pas d'investissement élevé dans le matériel. De plus, les technologies de codes QR et de codes-barres sont généralement faciles à déployer sans nécessiter d'intégration complexe dans les systèmes existants. Bien qu'elles ne puissent pas offrir certaines fonctionnalités en temps réel et d'automatisation comme la RFID, elles offrent tout de même une manière fiable d'enregistrer et de suivre les informations.

Voici les processus pour utiliser QR code ou code bar pour mieux tracer les pièces :

- **Génération des étiquettes** : Lors de la fabrication des pièces, générez des étiquettes QR ou des codes-barres avec un code d'identification unique pour chaque pièce. Ces étiquettes peuvent contenir des informations telles que la description de la pièce, la date de fabrication, le numéro de lot, les informations du fournisseur, etc.
- **Impression des étiquettes** : Utilisez une imprimante d'étiquettes pour imprimer les étiquettes QR ou les codes-barres générés et attachez-les aux pièces correspondantes.
- **Numérisation et enregistrement** : Tout au long de la chaîne d'approvisionnement, utilisez des dispositifs de numérisation pour scanner les codes QR ou les codes-barres des pièces et enregistrer la position et l'état de chaque pièce. Les dispositifs de numérisation peuvent être des scanners portatifs ou des applications de numérisation sur des smartphones.
- **Gestion des données** : Téléchargez les données de codes QR ou de codes-barres numérisés dans un système de gestion des données. Ce système stockera et gérera les informations de toutes les pièces pour un suivi, une analyse et des rapports ultérieurs.

Ce sont les deux méthodes possibles pour améliorer la traçabilité des pièces mais pas uniques. Si nous pouvons utiliser une méthode plus fiable pour suivre et enregistrer les pièces, la qualité des données sera grandement améliorée et nous pourrons utiliser ces données pour obtenir des résultats plus intéressants.

8 Résumé du stage

Cette expérience de stage chez GEHC a été bien plus qu'une simple opportunité professionnelle. C'était une étape cruciale dans mon développement en tant qu'individu et en tant qu'ingénieur en herbe. C'était ma première plongée dans le monde du travail, une transition excitante et instructive depuis l'envoi de candidatures jusqu'à l'intégration au sein d'une entreprise prestigieuse.

Dès le début, j'ai ressenti un grand enthousiasme pour ce stage, une attente impatiente d'apprendre et de grandir dans un environnement professionnel. GEHC a non seulement comblé ces attentes, mais les a également dépassées. J'y ai trouvé bien plus que des connaissances théoriques, j'ai trouvé une communauté de collègues gentils et bienveillants, ainsi que des amis formidables parmi les autres stagiaires. Ensemble, nous avons exploré les tenants et aboutissants de la vie professionnelle, partageant nos expériences et nos découvertes.

Ce stage m'a offert une occasion unique d'appliquer les connaissances acquises à l'école dans un contexte réel. J'ai travaillé sur des projets concrets qui avaient un impact direct sur l'entreprise. Mais ce qui a été le plus précieux, c'est l'apprentissage des compétences essentielles pour devenir un ingénieur accompli. J'ai observé des ingénieurs exceptionnels au sein de l'entreprise, tels que ma superviseure Céline et Robert. Leur passion pour leur travail, leurs années d'expérience et leur habileté à résoudre des problèmes complexes m'ont profondément impressionné. Ils ont été des mentors formidables, partageant leurs connaissances et leur expertise de manière généreuse.

Un moment clé de mon stage a été ma tentative de créer un outil de prédiction pour anticiper les défaillances des pièces. J'ai investi beaucoup de temps et d'efforts pour développer un tel outil. Cependant, ce projet m'a enseigné une leçon précieuse. En tant qu'ingénieur, il ne suffit pas d'avoir des compétences techniques solides, il est également essentiel de comprendre en profondeur le contexte et les besoins réels de l'entreprise. Mon initiative bien intentionnée a montré que l'analyse préalable et la compréhension approfondie sont tout aussi cruciales que la capacité à créer des solutions techniques.

En somme, ce stage a été bien plus qu'une simple expérience de travail. Il a été une étape d'apprentissage, de croissance personnelle et professionnelle. Il m'a appris que pour réussir en tant qu'ingénieur, il faut non seulement maîtriser les compétences techniques, mais aussi posséder un sens aigu de l'analyse, une compréhension profonde du contexte et une capacité à travailler en équipe. Ces leçons resteront gravées dans ma mémoire et guideront mon cheminement professionnel futur.

9 Annexe

9.1 Abréviations utilisées

FOA Failure On Arrival	4
GEHC General Electric Healthcare	5
FE Feild Engineer	6
RMA Return Merchandise Authorizations	15
PO Purchase Order	15
SO Sale Order	15
RFID Radio Frequency Identification	49
FDA Food and Drug Administration	50

Références

- [1] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, “Scikit-learn : Machine learning in Python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
- [2] ——. (2023) sklearn.clustering. [Online]. Available : <https://scikit-learn.org/stable/modules/clustering.html#clustering>
- [3] J. A. Hartigan and M. A. Wong, “Algorithm as 136 : A k-means clustering algorithm,” *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, vol. 28, no. 1, pp. 100–108, 1979. [Online]. Available : <http://www.jstor.org/stable/2346830>
- [4] D. Müllner, “Modern hierarchical, agglomerative clustering algorithms,” 2011.
- [5] M. Ester, H.-P. Kriegel, J. Sander, X. Xu *et al.*, “A density-based algorithm for discovering clusters in large spatial databases with noise,” in *kdd*, vol. 96, no. 34, 1996, pp. 226–231.
- [6] A. Ng, M. Jordan, and Y. Weiss, “On spectral clustering : Analysis and an algorithm,” in *Advances in Neural Information Processing Systems*, T. Dietterich, S. Becker, and Z. Ghahramani, Eds., vol. 14. MIT Press, 2001. [Online]. Available : https://proceedings.neurips.cc/paper_files/paper/2001/file/801272ee79cfde7fa5960571fee36b9b-Paper.pdf
- [7] D. D. Science. (2021) 3 easy steps to understand and implement spectral clustering in python. [Online]. Available : <https://www.youtube.com/watch?v=YHz0PHcuJnk>
- [8] L. Breiman, “Random forests,” *Machine learning*, vol. 45, pp. 5–32, 2001.
- [9] Anonyme. (2023) Algorithme n°2 – comprendre comment fonctionne un random forest en 5 min. [Online]. Available : <https://france.devoteam.com/paroles-dexperts/algorithme-n2-comprendre-comment-fonctionne-un-random-forest-en-5-min/>
- [10] J. G. Dy and C. E. Brodley, “Feature selection for unsupervised learning,” *Journal of machine learning research*, vol. 5, no. Aug, pp. 845–889, 2004.
- [11] V. Kumar and S. Minz, “Feature selection : a literature review,” *SmartCR*, vol. 4, no. 3, pp. 211–229, 2014.
- [12] J. Mertz. (2023) Documenting python code : A complete guide. [Online]. Available : <https://realpython.com/documenting-python-code/>
- [13] A. R. a.-c. D. N. D. F. F. f. G. B. g. J. L. A. j. J.-F. B. j. R. R. R. R. L. I. S. F. s. T. S. s. T. K. t. T. K. T. Y. S. s. Adam Turner. (2023) Sphinx makes it easy to create intelligent and beautiful documentation. [Online]. Available : <https://www.sphinx-doc.org/en/master/index.html>
- [14] avcourt. (2019) Auto-generated python documentation with sphinx (see comments for update fix). [Online]. Available : <https://www.youtube.com/watch?v=b4iFyrLQQh4>
- [15] P. Bhandari. (2022) Inductive reasoning | types, examples, explanation. [Online]. Available : <https://www.scribbr.com/methodology/inductive-reasoning/>
- [16] U. goverment. (2018) Radio frequency identification (rfid). [Online]. Available : <https://www.fda.gov/radiation-emitting-products/electromagnetic-compatibility-emc/radio-frequency-identification-rfid>

- [17] T. Electronics. (2022) Rfid : The technology making industries smarter. [Online]. Available : <https://www.statista.com/statistics/781314/global-rfid-technology-market-revenue-by-application/>
- [18] Statista. (2020) Taille de marketing rfid de 2016-2020. [Online]. Available : <https://www.statista.com/statistics/781314/global-rfid-technology-market-revenue-by-application/>
- [19] K. Domdouzis, B. Kumar, and C. Anumba, “Radio-frequency identification (rfid) applications : A brief introduction,” *Advanced Engineering Informatics*, vol. 21, no. 4, pp. 350–355, 2007.
- [20] FDA. (2013) Guidance for industry and food and drug administration staff. [Online]. Available : <https://www.fda.gov/media/130442/download>