PolyPublie
Polytechnique Montréal

POLYTECHNIQUE MONTRÉAL
UNIVERSITÉ D'INGÉNIERIE

Titre: Title:	Méthode d'analyse causale des défauts sur une ligne d'assemblage multiproduit
Auteur: Author:	Louis Puech
Date:	2022
Type:	Mémoire ou thèse / Dissertation or Thesis
Référence: Citation:	Puech, L. (2022). Méthode d'analyse causale des défauts sur une ligne d'assemblage multiproduit [Master's thesis, Polytechnique Montréal]. PolyPublie. https://publications.polymtl.ca/10361/

Document en libre accès dans PolyPublie
Open Access document in PolyPublie

URL de PolyPublie: PolyPublie URL:	https://publications.polymtl.ca/10361/
Directeurs de recherche: Advisors:	Robert Pellerin, & Camélia Dadouchi
Programme: Program:	Maîtrise recherche en génie industriel

Ce fichier a été téléchargé à partir de PolyPublie, le dépôt institutionnel de Polytechnique Montréal
This file has been downloaded from PolyPublie, the institutional repository of Polytechnique Montréal

https://publications.polymtl.ca

POLYTECHNIQUE MONTRÉAL

affiliée à l'Université de Montréal

Méthode d'analyse causale des défauts sur une ligne d'assemblage multiproduit

LOUIS PUECH

Département de mathématiques et de génie industriel

Mémoire présenté en vue de l'obtention du diplôme de *Maîtrise ès sciences appliquées*

Génie industriel

Mai 2022

© Louis Puech, 2022.

POLYTECHNIQUE MONTRÉAL

affiliée à l'Université de Montréal

Ce mémoire intitulé :

Méthode d'analyse causale des défauts sur une ligne d'assemblage multiproduit

présenté par Louis PUECH

en vue de l'obtention du diplôme de *Maîtrise ès sciences appliquées*

a été dûment accepté par le jury d'examen constitué de :

Martin TRÉPANIER, président

Robert PELLERIN, membre et directeur de recherche

Camélia DADOUCHI, membre et codirectrice de recherche

Samira KEIVANPOUR, membre

DÉDICACE

À tous ceux qui m'ont soutenu durant la réalisation de ma maîtrise de recherche.

REMERCIEMENTS

Je tiens à exprimer ma sincère reconnaissance à mes directeurs de recherche, Robert Pellerin et Camélia Dadouchi, pour leur soutien constructif et leur engagement tout au long de ma maîtrise. Je tiens à les remercier pour leur impact significatif sur mon travail et pour les leçons productives qu'ils m'ont enseignées au cours de ma formation.

Je tiens à remercier Olga Moreva et Stephan Schwarz ainsi que Samuel Lupien, Zacharie St-Pierre, et Alexandre Leblanc-Richard de m’avoir ouvert les portes de leurs entreprises et de m’avoir partagé leurs expériences concernant ma problématique.

Je tiens à remercier le professeur Bruno Agard pour son support durant la réalisation de la seconde étude de cas.

En outre, je tiens à remercier tous mes collègues pour l'aide qu'ils m'ont apportée pendant mes années universitaires.

RÉSUMÉ

L’apparition grandissante des lignes d’assemblage multiproduit au sein des industries manufacturières au cours de ces dernières années s’explique par la démocratisation du concept de personnalisation en masse des produits. Ces lignes d’assemblage multiproduit permettent d’augmenter le nombre de références produit par une entreprise, tout en gardant des coûts de fabrication faible. Cependant, les méthodes classiques d’analyse des causes racines des défauts tels que l’analyse des modes de défaillance et de leurs effets (FMEA) ou la méthode des six sigmas peinent à fournir des résultats satisfaisants sur de telles lignes d’assemblage.

Plus récemment, l'utilisation de méthodes de valorisation des données, extraites des systèmes industriels, a connu une forte croissance. Ces données ont un fort potentiel pour faciliter l'identification de produits défectueux et l’origine de ces défauts. L’objectif principal de cette recherche est d’ailleurs de **faciliter l’identification des causes racines des défauts sur les lignes d’assemblage multiproduit** par la valorisation de données.

Pour y arriver, nous avons proposé une méthode d’analyse des causes racines qui repose sur une succession d’algorithmes d’apprentissages automatisés (AA), un partitionnement de nos produits à l’aide d’un algorithme de partitionnement hiérarchique, une analyse statistique des taux de défauts des groupes obtenus précédemment, et finalement, une classification de ces groupes à l’aide d’un algorithme d’arbres de décision. Nous avons testé notre méthode sur une ligne d’assemblage automobile. Nous avons ainsi identifié que la succession de certains types de voitures dans la séquence de production augmente le taux de défauts. Puis, nous avons testé notre méthode dans un autre contexte industriel pour vérifier si celle-ci est généralisable. L’utilisation de notre méthode sur les données issues d’une machine d’assemblage de pneu utilisant une stratégie par lot de fabrication nous a permis d’identifier une opération particulière responsable de la majorité des défauts lors de l’assemblage des pneus. Au final, il semble donc que la combinaison de multiples techniques de valorisation de données permet d’améliorer le processus d’identification des causes racines de défauts sur une ligne d’assemblage multiproduits même si le nombre de défauts par référence est très faible.

ABSTRACT

The growing appearance of multi-product assembly lines in the manufacturing industry in recent years is explained by the democratization of the concept of mass customization of products. These multi-product assembly lines make it possible to increase the number of references produced by a company while keeping manufacturing costs low. However, the classical methods of root cause analysis of defects, such as the failure mode and effects analysis (FMEA) or the six sigma method, struggle to provide satisfactory results on these assembly lines.

More recently, the use of methods for exploiting data, extracted from industrial systems, has experienced strong growth. This data has a strong potential to facilitate the identification of defective products and the origin of these defects. As such, the main objective of this research is to facilitate the identification of root causes of defects on multi-product assembly lines through data mining.

To achieve this, we have proposed a root cause analysis method based on a succession of machine learning (ML) algorithms, a partitioning of our products using a hierarchical algorithm, a statistical analysis of the defect rates of the groups obtained previously, and finally, a classification of these groups using a decision tree algorithm. We tested our method through a case study concerning a car assembly line. We have identified that the succession of certain types of cars in the production sequence increases the defect rate. Then we test our method in another industrial context to check if it is generalizable. Using our method on data from a tire assembly machine using a batch strategy allowed us to identify a particular operation responsible for the majority of defects during tire assembly. In the end, it seems that the combination of multiple data mining techniques can improve the process of identifying the root causes of defects on a multi-product assembly line, even if the number of defects per reference is very low.

TABLE DES MATIÈRES

LISTE DES TABLEAUX

LISTE DES FIGURES

LISTE DES SIGLES ET ABRÉVIATIONS

AA	Apprentissage Automatisé
AGV	Automated Guided Vehicul ou véhicule à guidage automatique
AUC	Aera Under the Curve ou aire sous la courbe
AUROC	Aire sous la courbe ROC
CART	Classification And Regression Trees
CRISP-DM	Cross-Industry Standard Process for Data Mining
DOE	Design Of Experiment ou plan d'expériences
DRM	Design Research Methotology
EOL	End Of Line test ou test en fin de ligne
ERP	Enterprise Resources Planning ou progiciel de gestion intégré
GMB	Gradiant Boosting Machine
HCA	Hierarchical Cluster Analysis ou algorithmes de regroupement hiérarchique
IA	Intelligence Artificielle
MCC	Matthews Correlation Coefficient
ML	Machine learning ou apprentissage automatisé
RCA	Root Cause Analysis ou analyse des cause racine
ROS	Random OverSampling
RUS	Random Under Sampling
SGD	Stochastic Gradient Descent ou descente du gradient stochastique
SMOTE	Synthetic Minority Oversampling Technique
SVM	Support-Vector Machine ou machine à support de vecteur
VIN	Vehivule Identification Number ou numéro d'identification du véhicule

LISTE DES ANNEXES

CHAPITRE 1 INTRODUCTION

Sur le marché mondial hautement concurrentiel d'aujourd'hui, la qualité de production est un levier important de performance. Pour être compétitif, il faut une qualité de production presque parfaite, car la concurrence pousse les entreprises à réduire leur marge et les défauts représentent un coût important. Selon Töpfer (2017), les entreprises du secteur automobile dépensaient plus de 1000 € par véhicule en 2004 pour retravailler, voire mettre au rebut, les pièces défectueuses d'un produit lors des contrôles qualité. Ce coût aurait encore augmenté depuis (Hirsch et al., 2018).

Afin de réduire le nombre de défauts sur ces lignes d'assemblage, les industriels utilisent des outils afin d'identifier et d'éliminer les causes racines des défauts. Les méthodes les plus utilisées sont l'analyse des modes de défaillance et de leurs effets (FMEA) ou la méthode six sigma. Cependant, ces outils présentent certaines limites, en plus de représenter un coût important. Ils se concentrent sur un processus ou un produit en particulier et prennent du temps avant d'obtenir des résultats probants (Sand et al., 2016).

Or, lors des dernières années, le concept de personnalisation en masse a été adopté par de nombreuses industries. Ce concept vise à combiner les avantages des faibles coûts unitaires des processus de production de masse avec la fabrication de produits répondant à des exigences individuelles uniques. De ce fait, de nombreuses entreprises doivent faire face à un grand nombre d'options pour adapter les produits aux désirs individuels des clients. Elles ont développé des lignes d'assemblage à modèle mixte afin de produire de nombreuses références de produit qui répond au mieux aux demandes des clients. Ce concept de ligne de production mixte rend la détection des défauts ou l'analyse de leur cause racine de plus en plus difficile. L'augmentation du nombre de références a pour impact direct la réduction du nombre de défauts par référence de produits. Or, il est difficile, voire impossible même pour un expert, de relier un défaut à une cause si celle-ci a trop peu d'occurrences.

Dans un même temps, l'apparition de l'industrie 4.0 a permis de développer de nouvelles techniques pour détecter des défauts et identifier leurs origines sur les lignes d'assemblage. Ces techniques se basent sur des méthodes de valorisation des données extraites des systèmes industriels, qui ont connu une forte croissance pour atteindre environ 1000 exaoctets de données par an en 2018 (Tao et al., 2018). Ces données ont un potentiel intéressant. Elles peuvent mener à

des améliorations à tous les niveaux des entreprises. Cependant, selon Manns et al. (2015) et Moeuf et al. (2018), ces données demeurent peu exploitées. Plusieurs méthodes sont présentées dans la littérature pour identifier les produits défectueux et l'origine de ces défauts, incluant des méthodes d'inspections visuelles systématiques assistées par ordinateur ou des méthodes de détection de défauts basés sur des techniques d'apprentissage automatisées (AA).

La valorisation des données des chaînes d'assemblage fait aussi face à plusieurs obstacles techniques: le volume et l'hétérogénéité des types de données stockées et la faible représentation des pièces défectueuses dans les données disponibles complexifient leur identification (Nedelkoski et Stojanovski, 2019). De plus, la multitude de références produites sur une même ligne d'assemblage accentue ce phénomène en réduisant encore plus le nombre de défauts par références produites. L'apport à la recherche de ce mémoire réside dans **la proposition d'une méthode d'analyse des causes racines des défauts sur une ligne d'assemblage multiproduit basée sur les données.**

Ce mémoire est structuré en sept chapitres. Nous commencerons par un état de l'art des avancées technologiques liées à la détection et l'analyse des causes racines des défauts sur les lignes d'assemblage. Une analyse critique de cette littérature permet ainsi d'identifier les opportunités de recherche de l'étude. Ensuite, le chapitre 3 présente les objectifs spécifiques et la méthodologie de recherche permettant d'atteindre ces derniers. Le chapitre 4 décrit le contexte dans lequel évoluent nos deux partenaires industriels. Le chapitre 5 présentera la méthode d'analyse proposée des causes racines des défauts sur les lignes d'assemblage multiproduit. La validation de la faisabilité et l'analyse de performance de la méthode seront présentées au chapitre 6. Enfin, nous terminerons ce mémoire par une discussion portant sur les contributions scientifiques, les limitations et les opportunités de recherche futures qui découlent de ce travail.

CHAPITRE 2 REVUE DE LITTÉRATURE

Ce chapitre a pour but de recenser les progrès récents liés à notre sujet dans la littérature. Le chapitre débute avec la présentation des notions importantes pour conduire une analyse causale. Dans un second temps, nous présenterons les notions importantes concernant la valorisation des données. Ensuite, nous présenterons la stratégie de recherche permettant d'identifier dans la littérature les travaux en lien avec le sujet. Les articles retenus sont alors présentés sommairement. Le chapitre se conclut par une revue critique de ces articles et vise à mettre en évidence les limitations et les faiblesses des travaux répertoriés afin d'identifier les opportunités de recherches.

2.1 L'analyse causale

Cette section a pour objectif d'introduire les méthodes d'analyses des causes racines en y abordant la terminologie et les traits essentiels associés.

2.1.1 Terminologie

L'ASQ (American Society for Quality) propose une définition de l'analyse des causes racines (« Root Cause analysis ») :

> *"A root cause is defined as a factor that caused a nonconformance and should be permanently eliminated through process improvement. The root cause is the core issue—the highest-level cause—that sets in motion the entire cause-and-effect reaction that ultimately leads to the problem(s).*
>
> *Root cause analysis (RCA) is defined as a collective term that describes a wide range of approaches, tool, and techniques used to uncover causes of problems. Some RCA approaches are geared more toward identifying true root causes than others, some are more general problem-solving techniques, and others simply offer support for the core activity of root cause analysis." (ASQ, 2021)*

Une cause racine est à l'origine d'une succession d'évènements qui mène à un état particulier, dans notre cas, un défaut. C'est la cause fondamentale d'un problème observé au travers de symptômes. Cette cause racine est mise en évidence grâce à des méthodes d'analyses de cause racine. Ces méthodes englobent les approches, les outils, les techniques de mise en évidence et de résolution de problèmes. L'idée de l'analyse des causes racines est de reconnaître que toute non-performance dans un système quelconque est issue d'un effet de causalité en chaîne que l'on peut remonter et

résoudre pour pallier cette non-performance (Andersen et Fagerhaug, 2006). Cette définition est représentée dans la figure 2.1.

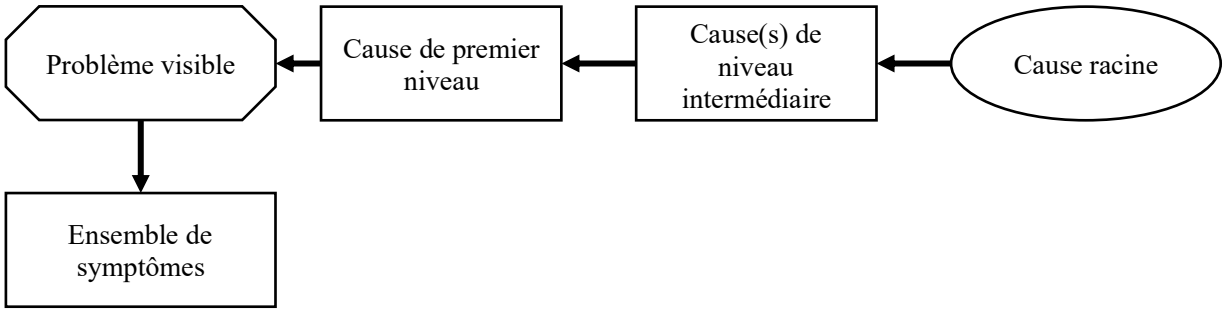


Figure 2.1 Relation de cause à effet, du problème jusqu'à la cause racine

L'analyse de cause racine ou l'analyse causale est une approche présente dans tous les domaines. Elle permet d'expliquer la cause d'une pathologie en médecine, d'analyser les failles de sécurité dans le domaine de l'informatique et d'expliquer les raisons de non-conformité des pièces en sortie de ligne d'assemblage.

Dans notre projet de recherche, les symptômes seront l'ensemble des défauts rencontrés sur les produits et les causes racines seront les causes fondamentales qui permettront d'expliquer ces défauts.

2.1.2 Étapes et activités de l'analyse causale

Le processus général pour réaliser une analyse des causes racines est présenté dans la figure 2.2 (Andersen et Fagerhaug, 2006). Cette méthode comporte cinq étapes : (1) la définition du problème, (2) la collecte d'information, (3) l'identification des causes possibles, (4) l'identification de la cause fondamentale, et enfin (5) la recommandation et la mise en place de solutions.

Figure 2.2 Les étapes du processus d'analyse de cause racine

1-L'identification du problème : consiste à reconnaître le problème observé via les symptômes qu'il déclenche. Ces symptômes doivent être les signaux d'alerte qui permettent d'identifier une situation problématique. Une fois clairement identifié, nous pouvons lier le problème à un signal d'alerte en particulier et réaliser un suivi temporel de ce dernier.

2-La collecte d'information : vise à renseigner le problème. Cela consiste à récolter toutes les informations disponibles pour décrire le problème afin de préparer les étapes suivantes.

3-L'identification de toutes les causes possibles : a pour objectif d'identifier la séquence d'évènements qui a mené au problème, les conditions qui ont permis l'émergence du problème, etc. L'équipe responsable de la conduite de l'analyse de cause racine peut procéder à une recherche de causes en utilisant la méthode des cinq pourquoi. Elle peut construire des diagrammes de cause à effet pour représenter la situation dans son ensemble.

4-L'identification de la cause racine : permet de mettre en lumière une des causes identifiées lors de l'étape précédant comme celle qui est fondamentalement responsable du problème. Beaucoup d'enjeux techniques résident dans cette phase. Souvent, la connaissance des experts dans le domaine permet d'atteindre une bonne identification de la cause racine.

5- Proposition et implantation de la solution : consiste à recommander et à implanter une solution pour répondre au problème constaté. Ces solutions peuvent être réactives pour répondre au problème immédiat et proactives pour éviter que le problème ne se reproduise.

La méthode présentée ci-dessus est applicable dans n'importe quel domaine. Les concepts de problème et de causes sont évidemment différents entre chaque secteur. Toutefois, la méthode de mise en évidence des causes pour expliquer un problème est universelle.

Dans l'industrie, une grande variété de méthodes conventionnelles de RCA sont appliquées, par exemple, les cinq pourquoi, la cartographie des causes, l'analyse des modes de défaillance et de leurs effets (AMDE), l'analyse des arbres de défaillance (ADF), etc. Ces approches sont encore largement appliquées par les praticiens en raison de leur simplicité. Cependant, avec la complexité croissante des problèmes de qualité et les exigences élevées en matière de précision et d'efficacité à l'ère de l'industrie 4.0, les méthodes traditionnelles de RCA sont critiquées sur plusieurs points :

- Les structures causales en forme d'arbres de décision des méthodes classique de RCA se concentrent sur les relations causales linéaires, ce qui entraîne une limitation des

interactions non linéaires dans les problèmes de qualité (Aurisicchio et al., 2016 ; Yuniarto, 2012) ;

- Les méthodes traditionnelles ne peuvent pas déterminer la pertinence des causes racines identifiées (Chemweno et al., 2016) ;
- Les résultats des RCA dépendent largement des connaissances et de l'expérience des experts qui mènent l'analyse, or dans un contexte industriel avec un fort taux de renouvellement, ces compétences sont difficiles à transférer et à préserver. (Mueller et al., 2018) ; et
- L'analyse manuelle lourde fait des RCA une tâche qui prend beaucoup de temps et demande beaucoup de travail (Xu & Dang, 2020).

Une solution proposée dans la littérature est l'utilisation de méthodes de valorisation des données que nous présenterons dans la section suivante.

2.2 La valorisation des données

Cette section a pour objectif d'introduire les méthodes de valorisations des données. La valorisation des données peut être définie comme le processus de collecte, de traitement et d'analyse de données permettant l'utilisation optimale de celles-ci dans la poursuite d'un objectif donné. Elle permet de manipuler un grand volume de données et d'en extraire de la connaissance exploitable. Parmi les méthodes utilisées en valorisation de données, nous retrouvons l'exploration des données, l'apprentissage automatisé, les mégadonnées et l'intelligence artificielle.

2.2.1 Terminologie

La valorisation des données fait partie du domaine de la science des données. De nombreux termes ont été utilisés pour la désigner au cours de ces dernières années : la découverte de connaissance dans les bases des données (*knowledge discovery in databases*), l'extraction des connaissances (*knowledge extraction*), l'analyse de données ou de patterns (*data/pattern analysis*), l'archéologie des données (*data archeology),* etc.

Turban et al. (2010) définissent la valorisation des données comme étant : "*the process that uses statistical, mathematical, artificial intelligence and machine-learning techniques to extract and*

identify useful information and subsequently gain knowledge from large databases". Selon eux, le processus de valorisation des données est un processus de découverte de connaissances intéressantes à partir de grandes quantités de données à l'aide de différentes méthodes. Une définition similaire est fournie par Berson et al. (2000), Lejeune (2001), Ahmed 2004 et Berry et Linoff (2004). Ils définissent l'exploration des données comme étant le processus d'extraire et de détecter les modèles cachés ou des informations dans une large base de données. L'objectif des processus d'exploration de données est de construire un modèle prédictif ou un modèle descriptif efficace à partir d'une grande quantité de données. Ce modèle doit non seulement prédire ou expliquer au mieux les données, mais il doit également pouvoir être généralisé à de nouveaux jeux de données.

2.2.2 Les étapes de la valorisation des données

Un processus typique de valorisation des données est présenté dans la figure 2.3. Cette méthode est appelée CRISP-DM (Cross-Industry Standard Process for Data Mining) (IBM, 2021), et est utilisée par les entreprises pour réaliser leur projet de valorisation de données.

I. L'étape de la **compréhension de l'entreprise** consiste à déterminer les objectifs de l'étude de valorisation de données, en se basant sur la connaissance de l'entreprise. Nous définissons aussi les critères de réussite de notre projet.

II. **La compréhension des données** implique l'étude des données disponibles pour le projet. Cette étape est déterminante pour la suite du projet, car elle permet d'éviter les problèmes au cours des étapes suivantes, spécialement l'étape de préparation des données.

III. **La préparation des données,** au vu de la modélisation, est l'étape la plus longue du projet, même si une réalisation rigoureuse des deux premières étapes permet de réduire significativement celle-ci. En fonction des objectifs de l'étude, la préparation des données comporte généralement les tâches suivantes : la fusion de table de données, la sélection d'un sous-ensemble de données à étudier pour l'analyse, le calcul de nouveaux attributs, le tri des données en vue de la modélisation, la suppression ou le remplacement des blancs ou des valeurs manquantes, le fractionnement en sous-ensembles d'apprentissage et de test des données.

IV. L'étape de **modélisation**, consiste au développement d'un modèle répondant à l'objectif identifier lors de la première étape.

V. Lors de l'étape de **l'évaluation** du modèle, nous vérifions si notre modelé répond aux critères définis lors de la première étape du projet.

VI. Le **déploiement** est l'étape consistant à utiliser les nouvelles connaissances obtenues au cours du projet pour apporter des améliorations au sein de l'entreprise.

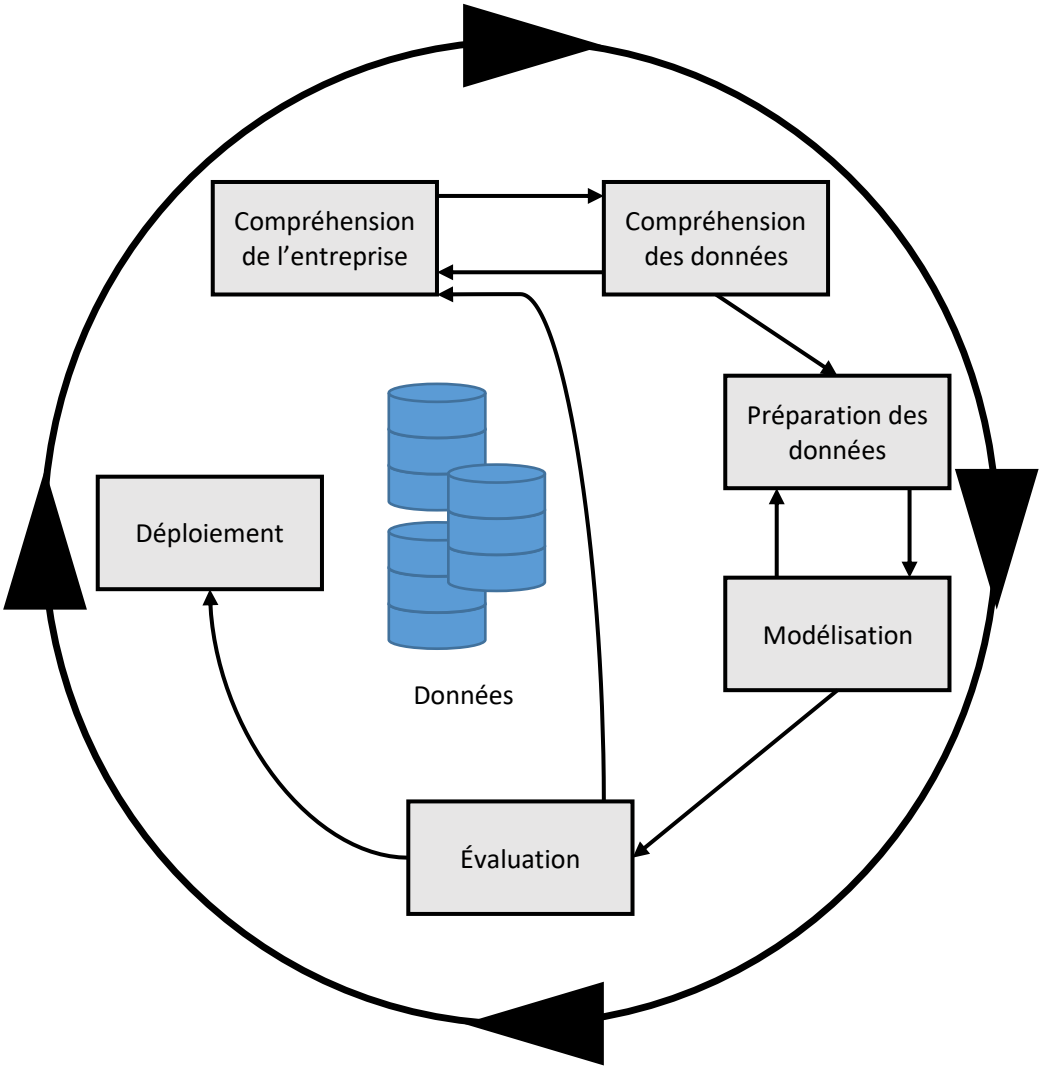


Figure 2.3: Représentation de la méthodologie CRISP-DM

2.2.3 Les techniques utilisées lors de la valorisation des données

Nous représentons dans la figure suivante un diagramme de Venn donnant les domaines connexes à la valorisation des données (Kulin et al. 2021).

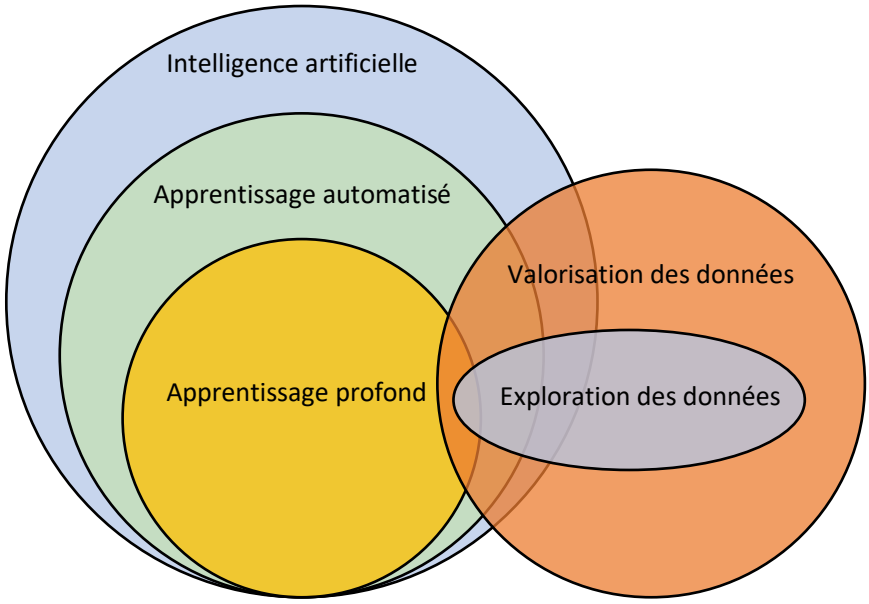


Figure 2.4: Diagramme de Venn de la valorisation des données

L'exploration des données ou data mining est une activité de la valorisation des données. Elle fait référence à l'application d'algorithme pour l'extraction de modèles à partir de données. Pour cela, elle utilise des algorithmes d'intelligence artificielle, d'apprentissage automatisé et d'apprentissage profond.

L'intelligence artificielle (IA), vise à la création de machines intelligentes mimant l'intelligence humaine. Des techniques d'IA telle que la reconnaissance des formes et les réseaux neuronaux ont reçu beaucoup d'attention dans la littérature scientifique et sont fréquemment utilisées pour réaliser des opérations de valorisation des données.

L'apprentissage automatisé (AA) ou machine learning, est une sous-catégorie de l'IA. L'AA est définie par Mehryar M. et al., (2018*)* comme étant des algorithmes capables d'apprendre des données qui visent l'amélioration de système.

L'apprentissage profond ou Deep learning, est une sous-catégorie de l'AA. LeCun et al., (2015) proposent comme définition de l'apprentissage profond un modèle de calcul composé de plusieurs couches de traitement pour apprendre des représentations de données.

Nous utiliserons l'ensemble de ces techniques dans notre stratégie de recherche afin d'inclure tous les articles traitant de la valorisation de données sur les lignes d'assemblage.

2.3 Stratégie de recherche et résultat

2.3.1 Définition de la stratégie de recherche

La revue de littérature cible les articles scientifiques et les articles de conférence de la base de données Compendex. Compendex est une des plus grandes bases de données de documents scientifiques en ligne. Elle offre une vue d'ensemble des progrès technologiques dans de multiples domaines comme la science, la technologie, les arts, la médecine, etc. Elle a permis d'identifier les articles proposant des méthodes d'analyse causale basées sur les données au sein de différentes entreprises. Nous nous intéressons aux articles publiés entre les années 2000 et 2021.

Trois points clés ont été identifiés pour conduire cette recherche d'article : **les défauts**, **les chaînes d'assemblage** et **les méthodes d'apprentissage automatique.** Pour identifier les termes de recherches adéquats à chacun de ces thèmes, une stratégie de recherche itérative a été conduite. Dans un premier temps, une recherche avec les mots-clés (defect* OR default* OR failure* OR fault) AND (assembly line OR assembly lines) AND (data mining OR machine learning OR deep learning OR artificial intelligence) nous a permis de recueillir 48 résultats.

Après une première lecture des titres et des résumés des articles, nous avons ajouté des termes d'exclusion afin de retirer les articles qui ne sont pas pertinents pour notre recherche. Concrètement, les critères d'exclusion suivants ont été appliqués : NOT (Maintenance) NOT (schedul*) NOT (balenc*).

Tableau 2.1 Mots-clés pour réaliser la recherche sur Compendex

Défauts	**Ligne d'assemblage**	**Apprentissage automatisé**	**Critère d'exclusion**
defect* OR default* OR failure* OR fault	"assembly line" OR "assembly lines"	"data mining" OR "machine learning" OR "deep learning" OR "artificial intelligence"	NOT (Maintenance) NOT (schedul*) NOT (balanc*)

Le tableau 2.1 répertorie les mots-clés utilisés pour extraire les articles de la base de données Compenex. La recherche des mots-clés sur Compendex se fait exclusivement sur les titres et les résumés des articles de conférences et les articles de journaux. La lecture de certains articles référencés dans la première sélection ainsi que des recherches supplémentaires sur *Google Gcholar* ont permis de faire ressortir d'autres contributions scientifiques pertinentes pour ce mémoire. Enfin, après la lecture complète des articles sélectionnés, une liste finale de publications scientifiques a été établie. Le tableau 2.2 présente la liste finale des articles sélectionnés pour l'analyse critique.

Tableau 2.2 Articles sélectionnés

	Auteurs	**Articles**
1	Osama et Pantea (2020)	Assembly Line Anomaly Detection and Root Cause Analysis Using Machine Learning.
2	Baranwal et al. (2019)	Five deep learning recipes for the mask-making industry.
3	Gardner et all. (2000)	Solving Tough Semiconductor Manufacturing Problems Using Data Mining
4	Han et all. (2019)	A root-cause analysis method for fault diagnosis in condenser
5	Hirsch et al. (2019)	Data-Driven Fault Diagnosis in End-of-Line Testing of Complex Products
6	Huang et all. (2010)	Fault Diagnosis of Analog Circuits Based on Machine Learning
7	Kane et Andhare (2016)	Application of Psychoacoustics for Gear Fault Diagnosis Using Artificial Neural Network
8	Lad et al. (2016)	High-Throughput Shape Classification Using Support Vector Machine
9	Laxman et al. (2009)	Temporal data mining for root-cause analysis of machine faults in automotive assembly lines
10	Li et al. (2018)	Deep Learning for Smart Industry: Efficient Manufacture Inspection System with Fog Computing
11	Mangal et Kumar (2016)	Using Big Data to Enhance the Bosch Production Line Performance: A Kaggle Challenge.
12	Maurya (2016)	Bayesian Optimization for Predicting Rare Internal Failures in Manufacturing Processes
13	Mueller et al. (2018)	Automated root cause analysis of non-conformities with machine learning algorithms
14	Nedelkoski et Stojanovski (2019)	Machine Learning for Large Scale Manufacturing Data with Limited Information.

Tableau 2.3 Articles sélectionnés (suite et fin)

15	Pavlyshenko (2016)	Machine Learning, Linear and Bayesian Models for Logistic Regression in Failure Detection Problems
16	Rahmatov et all. (2019)	Machine learning–based automated image processing for quality management in industrial Internet of Things
17	Rodriguez et all. (2010)	Failure Detection in Assembly: Force Signature Analysis
18	Sand et al. (2016)	Towards an Inline Quick Reaction System for Actuator Manufacturing Using Data Mining
19	Sarkar, (2004)	Clustering of Event Sequences for Failure Root Cause Analysis
20	Sassi et al. (2019)	A Smart Monitoring System for Automatic Welding Defect Detection
21	Schnell et all. (2019)	Data mining in lithium-ion battery cell production
22	Silva Peres et al. (2019)	Multistage Quality Control Using Machine Learning in the Automotive Industry
23	Wang et al. (2019)	Assembly defect detection of atomizers based on machine vision
24	Xu et Zhu, (2020)	Intelligent manufacturing Lie Group Machine Learning: real-time and efficient inspection system based on fog computing.

2.3.2 Résultat

Sur les 24 articles précédemment sélectionnés, 14 proposent une méthode de détection des défauts sur une ligne d'assemblage. Les dix autres articles proposent des méthodes qui permettent d'identifier les causes racines des défauts de leur jeu de données.

2.3.2.1 Méthode de détection des défauts

Dans cette section, nous séparons les articles en fonction des types de données qu'ils utilisent pour développer leur modèle. Dans un premier temps, nous nous intéresserons aux six articles qui utilisent des images pour détecter des défauts sur les produits. Puis, nous discuterons des sept autres articles qui utilisent d'autres types de données pour développer leur modèle.

2.3.2.1.1 Méthode de détection des défauts utilisant des images

Baranwal et al. (2019), Wang et al. (2019), Sassi et al. (2019), Li et al. (2018) et Rahmatov et al. (2019) utilisent un « *convolutional neural network (CNN)* » afin de prédire à partir d'images de pièces si celles-ci contiennent un défaut et déterminer leur catégorie.

Baranwal et al. (2019) présentent un modèle de CNN capable de déterminer si un robot d'une ligne d'assemblage de circuits imprimés a correctement pris un composant électronique. Leur modèle VGG16 permet de déterminer si le composant est correctement attrapé par le robot. Le modèle retourne un des huit défauts de préhension avec une précision supérieure à 99%.

Wang et al. (2019) entraînent un modèle pour détecter quatre types de défauts sur des radiographies de vaporisateur. Les radiographies des vaporisateurs sont manuellement marquées en fonction de la présence ou non d'un défaut et du type de celui-ci. N'ayant pas assez d'images pour entraîner son modèle, ils ont décidé de réutiliser plusieurs fois les mêmes images. Pour ne pas simplement doubler les radiographies, il met en place un déplacement aléatoire des vaporisateurs sur les images doublées et une variation aléatoire de la luminosité des images afin de simuler des conditions d'éclairage différentes selon les images. Leur modèle, MobilNet, est capable de détecter à 100% les trois types de défauts les plus gros, mais seulement à 83,33% des défauts plus petits. Il détecte 100% des pièces correctes.

Sassi et al. (2019) utilisent le RN DensNet afin de détecter deux catégories de défauts sur une tête d'injecteur. Le RN cherche à minimiser l'opposé de la vraisemblance (*negative log likelihood loss*) lors de l'apprentissage du modèle. Afin d'optimiser l'apprentissage, l'auteur utilise la méthode de la descente du gradient stochastique (SGD) qu'il couple au moment de Nesterov afin d'accélérer celui-ci. Les images de la base de données sont manuellement marquées en fonction des défauts rencontrés ou non. La base de données initiale contenait seulement 44 injecteurs défectueux. Pour réaliser l'apprentissage de son modèle malgré le peu d'images à disposition, il utilise une technique de « data augmentation » en réalisant une rotation aléatoire et un renversement des images disponible. À l'aide de ces techniques et de l'utilisation d'apprentissage par transfert, Sassi et al. (2019) ont été capables d'entraîner un modèle avec sept millions de paramètres d'entrée. La base d'apprentissage était constituée de 306 images, leur modèle était capable d'identifier les défauts avec une justesse de 97,22% et un rappel de 100%.

Li et al. (2018) proposent un modèle de CNN couplé au *Fog Computing* pour classifier des types de défauts et estimer leurs amplitudes. Ce modèle est basé sur une fonction de perte à objectifs multiples. Li et al. (2018) utilisent le SGD pour minimiser sa fonction objective représentant la différence entre les prédictions du modèle et la réalité. Afin d'avoir une prédiction rapide lors de l'analyse d'une image, les auteurs implémentent une sortie rapide à leur modèle pour avoir une

prédiction du type de défauts et sa magnitude le plus rapidement possible. Il teste leur modèle grâce à des images de ‘tile’. Les images sont collectées et marquées manuellement en fonction du type de défauts rencontrés et des experts indiquent leurs magnitudes. Les auteurs arrivent à la conclusion que leur modèle est capable de déterminer les pièces qui présentent des défauts et qu’il est plus juste et performant que certaines techniques de classification existantes en comparant les courbes ROC de la méthode étudiée avec celle des méthodes de classification classiques.

Rahmatov et al. (2019) proposent une méthode pour automatiser le contrôle qualité d’un produit en fin de ligne d’assemblage. Ils utilisent un apprentissage multi-instance avec des techniques de traitement d’image pour détecter les défauts dans une ligne de production de CPU. Chaque image d’un CPU est séparée en plusieurs zones de défauts possibles à l’aide de techniques floues de segmentation d'images et de reconnaissance de motifs. Puis pour chaque zone, ils prédissent si un composant est manquant à l’aide d’un CNN. Si l’une des parties renvoie un défaut, alors ils considèrent que le processeur est défectueux. Rahmatov et al. (2019) testent leur modèle avec 150 processeurs, dont 50 sont défectueux. Ils comparent sa performance par rapport à d’autres modèles. Globalement, leur modèle est le plus performant sauf pour détecter les défauts sur le support de processeur.

Xu et Zhu (2020) développent un algorithme de classification basé le « Lie Groupe machine learning.». Cette méthode utilise la distance intrinsèque pour calculer la distance entre deux points, au contraire de l’algorithme de partitionnement par la méthode des K centroïdes ou algorithme k-mean en anglais qui utilise la distance euclidienne. De plus, pour réduire les temps de calcul, le modèle est couplé à un système de « Fog computing » afin de réduire la charge de travail du serveur de calcul. Pour comparer la performance de son modèle à des modèles existants, Xu et Zhu (2020) collectent 150 images de cartes électroniques et 50 autres pour le tester. Chaque image est marquée manuellement en fonction de si elle présente un défaut ou non et si oui, à quelle catégorie des 12 défauts il appartient. Leur méthode est plus performante que la méthode K-mean et sa précision (accuracy) est supérieure de 35,77 % comparé aux méthodes existantes.

2.3.2.1.2 Méthode de détections utilisant d’autres formes de données

Cane et Andhare (2016) développent une méthode pour automatiser la détection de défauts sur des boîtes de vitesse au contrôle qualité de fin de ligne d’assemblage. Ils créent six variables psychoacoustiques pour caractériser le bruit que fait chaque boîte de vitesse. Ils utilisent ces six

variables et deux autres caractérisant le fonctionnement de la boîte de vitesse (la charge et la vitesse de rotation) pour entraîner un réseau de neurones (RN). Ils réalisent un apprentissage itératif de leur RN afin de réduire l'erreur moyenne au carré (MSE ou Mean Square Error) entre la sortie du RN et la valeur cible. À chaque itération de l'apprentissage, le gradient de la MSE est utilisé pour ajuster le poids des neurones. Ils utilisent comme condition de sortie une MSE inférieure à 10^{-5}, un gradient inférieur à 10^{-10}, ou plus de 5000 itérations. Après l'apprentissage, leur modèle est capable d'identifier correctement une boîte de vitesse en bonne condition de fonctionnement avec une précision de 99% et d'identifier une boîte de vitesse défectueuse avec une précision de 98%.

Maurya (2016), Pavlyshenko (2016), Nedelkoski et Stojanovski (2019), Mangal et Kumar (2016) et Silva Peres et al. (2019) utilisent deux méthodes de Gradient Boosting Machine (GBM) afin de classifier les pièces d'une ligne d'assemblage en fonction de la présence ou non d'un défaut. Maurya (2016), Pavlyshenko (2016), Nedelkoski et Stojanovsk (2019) et Mangal et Kumar (2016) utilisent tous les quatre la même base de données. Celle-ci a été utilisée lors d'un concours proposé par Bosch sur le site de Kaggle. La base de données représente les mesures effectuées sur un produit le long d'une ligne d'assemblage. L'objectif du concours étant de prédire en fonction de ces données si le produit va avoir un défaut ou non. Pour comparer les algorithmes proposés, le Matthews correlation coefficient (MCC) est utilisé comme critère de performance. La base de données est découpée en trois tables qui contiennent un seul type de données : catégorielle (2140 éléments de mesure), numérique (968 éléments de mesure) et des dates (1156 éléments de mesure). Le jeu de données d'entraînement regroupe en tout 1184687 mesures et le jeu de validation en a 1183748. Chaque produit est repéré grâce à un ID. Sur les 2140 éléments de mesures catégorielles, 500 sont multivariables, 1490 sont binaires et 150 sont vides. Afin de faire l'apprentissage de son modèle, Maurya (2016) réalise trois opérations de prétraitement des données. Il joint l'ensemble des trois tables de données, puis transforme les données catégorielles en données binaires fictives à l'aide d'une technique de « one-hot encoding » et affecte la valeur 0 à l'ensemble des données manquantes. Afin de réduire le temps d'apprentissage du modèle, l'auteur propose de créer quatre sous-tables de données pour l'apprentissage de son modèle. Finalement, après optimisation, Maurya (2016) propose un modèle de prédiction basé sur l'algorithme de Gradiant Boosting Machine (GBM) avec un MCC d'environ 0,46.

Pavlyshenko (2016), en plus des opérations de prétraitement réalisé par Maurya (2016), calcule le temps passé sur la ligne d'assemblage pour chaque pièce. Il détermine l'importance de chaque

variable de son modèle en appliquant le classifieur XGBoost à un sous-ensemble aléatoire de ces données d'entraînement. Puis pour la suite de son étude, il considère seulement les 500 variables les plus importantes. Il entraîne son modèle de *gradient boosting* avec ces 500 variables et obtient un modèle de prédiction avec une aire sous la courbe (AUC) de 0,753 et un MCC de 0,26 si l'on considère que les pièces sont défectueuses à partir d'un seuil de 26% de chance de défauts.

Nedelkoski et Stojanovski (2019) et Mangal et Kumar (2016) cherchent aussi à déterminer la présence ou non de défaut sur la ligne d'assemblage. Cependant, leurs modèles sont basés sur le XGBoost. XGBoost ou eXtreme Gradient Boosting, qui est une optimisation du gradient boosted trees.

Nedelkoski et Stojanovski (2019) font le choix de considérer seulement les données numériques et les dates pour créer leur modèle. Après une phase d'exploration de la base de données, ils créent 16 nouvelles variables à partir d'une comparaison des colonnes de la base de données en fonction du « start time » et de « ID » des pièces. Ils entraînent un premier modèle. Celui-ci utilise 754 variables sur les 4000 disponibles. Ils découpent leur base de données en quatre catégories en fonction de deux critères : les pièces qui ont un « ID » consécutif et celle qui présente un duplicata ou non. Ils remarquent que les pièces qui n'ont pas d'« ID » consécutif ne présentent pas de défauts. Ils décident donc de baser l'apprentissage seulement sur les pièces qui ont un « ID » consécutif. Ils optimisent leur modèle final en fonction du MCC et obtient finalement un MCC de 0,49421.

À la différence de Nedelkoski et Stojanovski (2019), Mangal et Kumar (2016) décident de considérer l'ensemble des données disponibles pour l'apprentissage de leur modèle de prédiction. Après avoir retiré les données catégorielles vides de leur base de données, ils les transforment en une variable numérique unique à l'aide de la technique « *Follow the Regularized Leader* » (FTRL). Ils entraînent un premier modèle en fonction des données numériques, des dates et de la variable extraite des données catégorielles afin de déterminer l'importance de chacune des variables. Ils décident par la suite d'optimiser un modèle qui prend en compte seulement les 200 variables les plus importantes. Tout comme Nedelkoski et Stojanovski (2019), ils optimisent leur modèle afin de maximiser le MCC et obtiennent un MCC de 0,21514.

Silva Peres et al. (2019) présentent un modèle de classification qui détermine si le châssis d'une voiture présente un défaut. Pour ce faire, ils utilisent 29 mesures extraites d'une station de contrôle du châssis placé sur une ligne d'assemblage multiproduit de Volkswagen. Dans un premier temps,

Silva Peres et al. (2019) éliminent les données qui ont plus de 85% de valeurs manquantes. Puis, pour remédier au problème de la faible représentation des pièces avec défaut dans la base de données, ils utilisent la méthode de *Random Under-Sampling* (RUS) afin de créer un jeu de données équilibré pour entraîner leur modèle. Leur modèle final permet de déterminer la présence d'un défaut avec un AUC de 0,972.

Lad et al. (2016) proposent une méthode pour déterminer si un cachet est entier ou non avant son emballage. Pour ce faire, il extrait 12 paramètres de deux capteurs IR qui observent la chute d'un cachet. Ils utilisent un modèle de machine à support de vecteur (SVM), afin de prédire si le cachet est entier ou non. Leur modèle est capable de trier les cachets avec une précision (accuracy) de 95,24%.

Rodriguez et al. (2010) proposent une méthode similaire pour détecter les erreurs dans un assemblage automatique à l'aide d'un capteur de force. Leur modèle est basé sur une SVM. Ils testent leur méthode sur un assemblage de coque de protection sur une carte électronique de téléphone. Après avoir déterminé l'orientation optimale de leur capteur, ils utilisent la méthode d'analyse des composants principaux ou PCA, afin d'identifier les paramètres qui ont le plus d'impact sur le modèle. Puis, ces données sont utilisées comme données d'entrées dans une SVM. Leur modèle est capable de détecter les défauts avec une précision de 99,8%.

2.3.2.2 Méthode d'identification des causes des défauts

Osama et Pantea (2020) proposent une méthode pour détecter les anomalies et identifier leurs causes grâce à des données de production pour deux produits. Ils comparent plusieurs techniques de détection des défauts qu'ils comparent grâce aux aires sous la courbe (ROC/AUROC) et au Rank Power. Puis, ils utilisent des techniques statistiques d'analyse des causes des défauts afin d'identifier grâce à un Pareto sept causes qui expliquent ces défauts. Les données utiliées pour entraîner les différents modèles correspondent à des mesures réalisées automatiquement sur les produits à l'aide d'un système de vision automatique utilisé pour le processus d'inspection. Ils ont à leur disposition 309 attributs pour le premier produit et 158 attributs pour le second. Après prétraitement des données, les caractéristiques ont été agrégées en prenant les valeurs minimum et maximum pour chaque type de mesure, et ce, principalement pour conserver la variance des données. Les résultats ont montré qu'il y a 62 points de données anormaux pour le second produit en utilisant l'algorithme ABOD et 343 points de données anormaux pour le premier produit en

utilisant l'algorithme KNN sans qu'il y ait une présence claire de sur-rejet dans les machines d'assemblage pour les deux séries. En outre, les résultats ont montré qu'il y a sept causes de rejet pour chaque série, alors que les trois premières causes sont responsables de 86% et 85% des taux de rejet dans les séries des produits.

Sarkar (2004) a proposé une méthode basée sur le regroupement des séquences d'événements menant à un défaut pour en identifier la cause racine de ces défauts. Il crée une base de données constituée de séquences d'événements menant à une défaillance en surveillant un système électromécanique. Chaque fois que le système signale une défaillance, il enregistre les n événements précédents dans une base de données. Il a collecté 334 séquences tampons d'événements provenant de l'équipement et a trouvé 75 messages de défaillance distincts parmi ces 334 séquences. Il calcule la distance entre deux séquences en appliquant un algorithme de correspondance de séquence populaire, utilisé en bio-informatique, appelé "alignement global" : algorithme de type Needleman-Wunsch". Cet algorithme crée une matrice constituée de la distance par paire entre deux séquences tampons. Il utilise cette distance pour mettre en œuvre une méthode de regroupement hiérarchique afin de générer k clusters. Il calcule la distance entre les nouveaux clusters et les anciens en utilisant la "distance moyenne à l'intérieur". Le nombre de clusters a un impact direct sur l'interprétation du résultat. Plus le nombre de clusters est élevé, plus les l'homogénéité à l'intérieur d'un segment augmente. Cela augmente la précision du modèle sur l'ensemble d'entraînement, mais diminue sa capacité à être généralisé. Pour déterminer le nombre optimal de clusters, il note les deux codes de défaut les plus courants de chaque cluster. Si deux clusters partagent les mêmes codes de défaut les plus courants, il considère qu'il a surclassé sa base de données, et donc, il doit réduire le nombre de clusters de son étude. Dans son étude de cas, il obtient huit clusters. Il a ensuite utilisé ses connaissances en ingénierie pour examiner de plus près chacun de ces clusters afin d'identifier la cause racine de la panne.

Sand et al. (2016) ont créé une méthode pour détecter et signaler rapidement les anomalies sur une ligne d'assemblage d'actionneur électromagnétique. Ils ont utilisé les données collectées pendant une journée sur tous les composants défectueux pour valider leur approche. Leur base de données comprend 84 pièces contenant 115 paramètres sélectionnés parmi les paramètres du processus le long de la ligne d'assemblage et les données de qualité des tests en fin de ligne (EOL). Les paramètres du processus comprennent, par exemple, la force de pression d'une pièce fournie sur la ligne d'assemblage. Ils utilisent une combinaison d'une méthode de partitionnement et d'un

algorithme d'arbre de décision pour analyser ces données. Ils testent toutes les combinaisons possibles entre les algorithmes de partitionnement suivants : partitionnement hiérarchique, carte auto-organisatrice (SOM), partitionnement en k-means, partitionnement par maximisation de l'espérance (EM), et algorithme d'arbre de décision (Random Forest (RF), Random Tree (RT), Best First Decision Tree (BFTree), et C4.5). D'après les résultats, tous les algorithmes d'arbre de décision donnent de bons résultats avec la méthode de partitionnement hiérarchique, sauf l'algorithme RF. Le BFTtree fournit le meilleur résultat global avec l'algorithme de partitionnement hiérarchique. Pour le RCA, ils utilisent la séparation du nœud de l'arbre fournie par le BFTree et les connaissances techniques pour identifier la cause première de l'augmentation de 100 % du taux d'erreur, ce qui permet d'accélérer l'interprétation de la cause première des défauts.

Hirsch et al. (2019) ont présenté une méthode pour déterminer le composant défectueux d'un moteur électrique parmi 85. Ils exploitent leur modèle uniquement à partir des données des moteurs défectueux. Ces données proviennent de trois sources : les caractéristiques du moteur, les tests effectués sur la chaîne de montage et les opérations de reprise effectuées sur les moteurs. Après avoir éliminé les moteurs avec plus de 20% de données manquantes, ils utilisent "KNN input" pour remplacer les données manquantes par leurs plus proches voisins. Enfin, ils suppriment les variables dont la variance est nulle ou presque nulle, normalisent et centrent toutes les variables. Ils testent les algorithmes AdaBoost et RF et des algorithmes de classification (KNN et C5.0) avec différentes stratégies d'échantillonnage (RUS, Random Over-Sampling ou ROS, technique de Synthetic Minority Oversampling Technique ou SMOTE), ou avec la sélection de caractéristiques réalisées avec l'algorithme de Boruta. Pour évaluer les performances des algorithmes, ils calculent la probabilité de déterminer correctement le composant défectueux à la p° tentative. Les résultats montrent que RF en association avec Boruta fournit les meilleurs résultats. Leur modèle peut déterminer le composant défectueux avec une précision de 33% à la première tentative et de 42% à la deuxième tentative. Cela équivaut à dire qu'il y a 60% de chances de trouver le composant défectueux en deux tentatives, ce qui est mieux que les quatre tentatives requises en moyenne par les ingénieurs qualité de l'entreprise.

Huang et al. (2010) ont présenté une méthode de diagnostic des défauts qui s'appuie sur l'apprentissage automatique pour séparer les dispositifs défectueux en deux catégories selon leur importance (grave ou faible), puis transmettre le dispositif défectueux à l'outil de diagnostic approprié. Ils ont entraîné leur modèle de diagnostic avec des données d'équipements artificiels qui

représente un amplificateur radio à faible bruit. Leur méthode est basée sur l'utilisation d'un filtre de défauts et de deux SVM. Le filtre de défauts est entraîné pour distinguer les défauts graves des défauts faibles dans les appareils. Les dispositifs défectueux sont transmis au classificateur. Ils ont utilisé un classificateur multiclasse avec N sorties, où N est le nombre de défauts catastrophiques modélisés dans la phase de prédiagnostic. Les dispositifs légèrement défectueux sont transmis à un modèle de régression afin d'identifier le défaut et sa localisation. Selon l'étude de cas, cette méthode permet d'obtenir une performance globale élevée en matière de diagnostic.

Schnell et al. (2019) ont présenté une méthode permettant d'identifier la cause profonde des défauts sur des batteries de téléphones portables de faible capacité. Ils ont utilisé les données d'équipement de la chaîne de montage pour entraîner leur modèle. Ils ont sélectionné les données pertinentes en supprimant toutes les caractéristiques contenant des données sans variation. Ils ont ensuite converti les attributs non numériques en attributs numériques. Les données manquantes ont été ajoutées manuellement. Enfin, ils ont normalisé les données pour assurer un formatage correct et empêcher que les différentes échelles de paramètres n'affectent les résultats de l'analyse. Après toutes les étapes de prétraitement, la base de données du modèle comprenait 113 cellules de batterie et 88 paramètres de production, principalement des données d'assemblage. Ensuite, pour analyser l'ensemble des données préparées, différentes techniques de modélisation ont été comparées : le modèle linéaire généralisé (GLM), la régression par vecteur de support (SVR), les réseaux neuronaux artificiels (ANN), les arbres de décision (DT), la forêt aléatoire (RF) et les arbres boostés par gradient (GBT). La base de la comparaison quantitative des modèles dans leur analyse est l'erreur quadratique moyenne (RMSE), une mesure de qualité standard pour les modèles de régression prédictive utilisée pour comparer les différentes techniques de modélisation. D'après les résultats, le GLM et le GBT fournissent les meilleurs résultats (RMSE de 0,42 Ah). En raison de leur facilité d'interprétation, l'analyse des modèles GLM et DT a été approfondi afin d'identifier les principaux facteurs menant aux défauts.

Han et al. (2019) ont présenté une méthode d'analyse des causes profondes basée sur des modèles de régression multiple. Ils s'intéressent aux causes de l'augmentation de la pression dans des condenseurs dans une centrale thermique. Ces condenseurs sont sous vide pour des raisons de sécurité. D'après la littérature, les variables qui affectent la pression dans un condenseur sont la température d'entrée du condenseur, le débit d'entrée et la température d'entrée d'eau. Leur méthode est basée sur trois parties. Tout d'abord, une représentation linéaire par morceaux est utilisée pour

obtenir les tendances des séquences temporelles historiques. Pour chaque variable, l'algorithme bottom-up est utilisé ici pour obtenir les représentations linéaires par morceaux. Deuxièmement, un seuil de changement des variables est utilisé pour combiner les tendances qualitatives et les statistiques. Pour ce faire, ils ont calculé le seuil des changements significatifs pour chaque variable. Ils ont déterminé les tendances de chaque variable et la combinaison des tendances est obtenue à partir des données historiques. Enfin, une analyse de l'augmentation de la pression du modèle de régression multiple est utilisée pour déterminer la contribution de chaque variable à l'augmentation de la pression en fonction du comportement des autres variables. En analysant cet indice, ils identifient deux causes qui conduisent à une fuite de vide dans le condenseur.

Mueller et al. (2018) ont développé une méthode d'automatisation de la RCA basée sur un algorithme d'arbre de décision. Pour valider leur approche, ils créent un modèle pour prédire la rugosité d'un trou de perçage. Ils utilisent des données artificielles pour complètement contrôler les données d'entrée afin de tester le comportement de leur modèle face à diverses situations. Ils discrétisent les données d'entrée de leur algorithme pour améliorer sa performance. Grâce à l'arbre fourni par C5.0, ils identifient la cause racine de défaut. Ils ont ensuite testé la réaction du modèle en fonction de plusieurs modifications des données d'entrée. En faisant varier les données d'entrée du modèle et en comparant sa performance à un algorithme de Monte-Carlo, ils sont capables de tester la performance de leur modèle dans plusieurs situations et d'identifier les limites de leur modèle.

Gardner et al. (2000) ont présenté un algorithme d'AA non supervisé pour effectuer un RCA. Ils ont utilisé une combinaison de deux algorithmes : un réseau neuronal de cartes auto-organisées (SOM) et un algorithme d'induction de règles. Ils testent leur méthode pour identifier les facteurs de critique de qualité dans la fabrication de plaquettes. Les données ont été collectées pour 2500 plaquettes pendant 2 mois. La base de données était constituée de 17 246 entrées pour lesquelles on mesurait 133 paramètres, le tout organisé dans un fichier Excel. Il a fallu deux mois pour collecter manuellement les données, les examiner, puis en corriger l'intégrité. Les données comprenaient le nombre de réussites/échecs par plaquette de 39 tests fonctionnels, toutes les données de contrôle du processus et certaines données sur les étapes du processus. L'algorithme de réseau neuronal SOM effectue un type de "régression multivariée non linéaire". L'algorithme crée une topologie relationnelle bidimensionnelle, appelée "carte des clusters". La carte bidimensionnelle est simplement une construction mathématique qui sert de treillis pour

l'organisation des données. Les dimensions X et Y n'ont aucune signification. La présence d'un cluster sur la carte indique qu'une relation statistique significative existe au sein des données. La taille du cluster indique la force de la relation. La relation spatiale des clusters entre eux montre leur similarité relative. Par conséquent, les modèles significatifs peuvent être vus comme des clusters. L'induction de règles est un algorithme supplémentaire d'exploration de données non supervisée qui fonctionne en synergie avec la carte des clusters. Il génère des expressions logiques (règles) qui identifient les attributs de données qui discriminent le plus entre les clusters, expliquant ainsi les clusters. Ils ont utilisé cette méthode dans deux scénarios. Le premier scénario permet de comprendre une baisse périodique de 5% et 2% des rendements de la ligne de production. La méthode proposée a permis d'identifier deux causes fondamentales expliquant ces baisses de rendement. Dans le second scénario, ils ont essayé d'expliquer une variation importante du facteur de gain (bêta) du transistor dans un nouveau produit. La méthode a permis d'identifier la combinaison de deux causes racines comme étant la cause de la variation du gain béta.

Laxman et al. ont utilisé une méthode basée sur un algorithme heuristique et l'exploration de données temporelles d'un journal des pannes d'une usine de moteurs afin de trouver les causes des défauts. Le journal des pannes d'une machine est une séquence ordonnée dans le temps constituée de codes qui représente des pannes qui se sont produites sur une ligne. Chaque code est enregistré de manière unique afin de pouvoir localiser et d'identifier la panne sur la ligne. L'objectif du processus d'exploration de données ici est de découvrir tous les épisodes fréquents. Un épisode est dit fréquent si sa fréquence d'apparitions dans la base de données dépasse un seuil. Pour réduire le nombre d'épisodes à étudier, Laxman et al. (2009) présentent une méthode heuristique de découverte des épisodes fréquents. Tout d'abord, tous les épisodes fréquents de taille 1 sont trouvés. Ces épisodes sont ensuite combinés pour obtenir des épisodes candidats de taille 2 en utilisant une procédure de génération de candidats. Une fois que les épisodes fréquents de taille 2 sont ainsi obtenus, ils sont utilisés pour construire des épisodes candidats de taille 3, et ainsi de suite. Laxman et al. (2009) ajoutent une sous-contrainte pour incorporer la connaissance du domaine de l'usine, un temps d'expiration entre le premier et le dernier événement d'un épisode et un filtre pour ne considérer que les événements dont la durée d'arrêt est comprise dans l'intervalle [1-1800]. De plus, ils n'affichent que l'épisode qui ne concerne que la même machine ou la suivante, car l'épisode qui concerne plusieurs machines sur toute la chaîne de montage sera difficile à interpréter et donc inutile pour l'analyse des causes des pannes. De cet épisode, ils peuvent extraire quelques règles

qui peuvent être utilisées pour déterminer la cause racine d'une faute. Leur partenaire industriel ont expliqué la cause profonde d'un code d'erreur dans la machine nouvellement installée avec cette méthode. Cette méthode peut déterminer si une erreur en entraîne une autre en l'appliquant à notre journal des erreurs.

2.3.3 Revue critique

L'état de l'art des travaux de recherche sur les techniques d'apprentissage automatisé sur les chaînes d'assemblage montre un intérêt récent de la part de la communauté scientifique sur l'exploitation des méthodes de fouille et de valorisation de données industrielles pour expliquer et détecter ces défauts.

Cependant, cet intérêt semble légèrement plus important pour la détection de défaut que pour leur explication. Lors de la définition de stratégie de recherche, nous n'avons pas précisé si nous nous intéresserons plus à la détection des défauts ou à leur explication au sein des lignes d'assemblage. Ainsi, 14 des 24 articles traitent de la détection des défauts, tandis que 10 articles traitent de l'analyse des causes racines des défauts sur les lignes d'assemblage.

Osama et Pantea (2020) mettent en garde vis-à-vis de la stabilité des modèles de détection des défauts. Selon eux, il est bien connu dans la littérature que les techniques de détection d'anomalies sont confrontées à des problèmes d'instabilité en raison des données et de la nature de l'apprentissage non supervisé. Baranwal et al. (2019) note eux aussi que leur modèle est sensible aux conditions d'éclairage et nécessite une phase de réapprentissage chaque fois que leur modèle est implémenté sur une nouvelle ligne. Afin de gérer ces types de problèmes, Wang et al. (2009) utilise des techniques de « data augmentation » afin de simuler un changement dans les données d'entre pour faire l'apprentissage de son modèle dans toutes les configurations possibles. Mueller et al. (2019) propose une solution similaire en utilisant une portée plus importante que la portée réelle pour chaque variable. Osama et Pantea (2020), envisagent comme solution à ce problème d'utiliser plus de données afin de traiter le plus de cas possible, mais cela augmente les temps de calcul lors de l'apprentissage du modèle.

Sur les 14 articles qui traitent des détections des défauts sur les lignes d'assemblage, seuls les articles de Baranwal et al. (2019) et Huang et al. (2010) présentent leur méthode de prétraitement des données. Or, cette étape est selon la littérature l'une des plus importantes. Concernant les

articles traitant de l'analyse des causes racines des défauts, si l'on reprend les différentes étapes du processus d'analyse de cause racine, on remarque que les méthodes présentées dans l'état de l'art vont plus ou moins loin dans cette démarche. Toutes les méthodes vont au moins jusqu'à la 3° étape :l'identification de toutes les causes possibles. Cependant l'étape 4 : l'identification de la cause racine reste majoritairement dépendante de l'analyse d'un expert. Sur les dix articles retenus, six font appel à un expert pour formellement identifier la cause racine du problème sur la ligne d'assemblage. Dans ces six articles, l'expert utilise les conclusions des méthodes présentées pour mener leur analyse. Ils utilisent ces méthodes comme des outils pour arriver à des conclusions qu'il ne serait pas évident à trouver avec les méthodes classiques d'analyse.

Cependant, nous pouvons identifier une grande lacune dans la littérature scientifique. Aucun article scientifique ne traite des chaînes de montage multiproduit. Or, ce genre de ligne d'assemblage est l'une des solutions adoptées par les industries pour répondre au besoin de personnalisation en masse.

2.3.4 Conclusion

Cette revue de littérature nous a permis de constater que l'analyse causale des défauts sur une chaîne d'assemblage multiproduit soulève de multiples problématiques. À l'aide des travaux trouvés dans la littérature, nous avons constaté qu'il n'existe pas de méthode d'analyse des causes racines sur une ligne d'assemblage multiproduit. Nous allons donc tenter de pallier ce manquement en proposant une méthode d'analyse causale des défauts spécifique à ce contexte. La méthodologie de recherche utilisée pour répondre à ces besoins est décrite au prochain chapitre.

CHAPITRE 3 MÉTHODOLOGIE DE RECHERCHE

Ce chapitre présente l'approche de recherche utilisée pour mener à bien cette étude. Dans un premier lieu, les objectifs spécifiques de la recherche seront présentés. Ensuite, nous décrirons la méthodologie de recherche utilisée, avant de présenter les études de cas réalisées au sein des entreprises partenaires qui ont servi de base au développement et à la validation de la méthode proposée.

3.1 Objectifs de recherche

Habituellement, lors de l'apparition d'un défaut sur les lignes d'assemblages, celui-ci est corrigé immédiatement. Cette démarche est en accord avec la gestion au plus juste. L'opérateur qui constate le défaut le corrige immédiatement, ou bien utilise un signal d'alerte pour demander de l'aide pour corriger le défaut dans les plus brefs délais. La plupart du temps, ces problèmes sont réglés en quelques secondes. Mais s'il s'agit d'un problème grave, alors celui-ci est corrigé dans les délais les plus courts avant qu'il ne génère de la non-qualité en aval dans le processus de production. Ces mesures permettent de corriger les défauts au plus tôt, mais ne permettent pas nécessairement d'en supprimer la cause. De plus, elle représente un coût pour les entreprises puisqu'elles peuvent entraîner des pertes de productivité, en raison des ralentissements ou de l'arrêt de la ligne de production. Identifier la cause racine de ces défauts pour les éliminer définitivement favoriserait l'augmentation de la productivité des entreprises. Des méthodes d'analyse des causes racines existent dans la littérature scientifique, mais aucune ne traite le cas des lignes d'assemblage multiproduits.

Sur l'ensemble des articles identifié lors de la revue de littératures, peu d'articles traitent directement de l'analyse des causes racines des défauts sur les lignes d'assemblage. Sur les 24 articles identifiés, seulement 10 abordent le sujet ou sont dédiés à cette problématique. De plus, les analyses réalisées dans ces articles sont entièrement centrées autour d'un produit unique. Cela leur permet d'avoir une quantité de données importante par produit disponible pour leur analyse, contrairement au cas des lignes d'assemblage multiproduits, ou la quantité de données par produit est bien plus faible, et ne permet pas une analyse distincte par produit. De plus, dans le cas d'une analyse d'une chaîne d'assemblage classique, le processus de production est identique pour tous

les produits. Tandis que pour une ligne d'assemblage multiproduit, celui-ci peut varier en fonction des références, ce qui complexifie grandement l'analyse des causes racines des défauts.

Par conséquent, l'objectif de ce mémoire est de proposer une méthode pour **faciliter l'identification des causes racines des défauts sur les lignes d'assemblage multiproduits en exploitant les données disponibles.** On doit toutefois noter que la recherche d'une cause unique est peu probable a priori. Comme en témoigne la revue de la littérature, les modèles de valorisation de données en support à l'analyse causale varient grandement d'un cas à l'autre. Les spécificités de chaque processus d'assemblage, l'hétérogénéité des données disponibles et les caractéristiques intrinsèques à chaque produit étudié ont amené les chercheurs à préconiser un ensemble varié de techniques de prétraitement et d'analyse. Reconnaissant cet obstacle, nous chercherons à atteindre les trois sous objectifs spécifiques suivant:

Sous-objectif 1 : *déterminer les méthodes de préparation et d'analyse des données potentiellement applicables au domaine de production multiproduit pour la réalisation d'une analyse causale des défauts.*

Sous-objectif 2 : *développer une méthode d'analyse causale des défauts appliquable au ligne d'assemblage multiproduit.*

Sous-objectif 3 : *mesurer le caractère de généralisation de la méthode proposée.*

3.2 Méthodologie de recherche

Compte tenu de la nouveauté du phénomène étudié et de l'objectif de recherche visant à développer un cadre appliqué pour déterminer les causes racines des défauts sur les lignes d'assemblage multiproduit, nous choisissons d'utiliser une méthodologie de recherche de type empirique et expérimentale nommée DRM, ou Design Research Methotology, qui est proposé par Blessing et Chakrabarti (2009).

Notre démarche est limitée en partie par la durée nécessaire pour implanter et valider la méthode proposée dans un contexte réel. En effet, la qualité de l'implantation est tributaire de nombreux facteurs qui pourraient avoir un impact sur l'applicabilité de notre méthode. Heureusement, la DRM ne nécessite pas une implantation complète de la solution pour évaluer la validité et la pertinence des résultats.

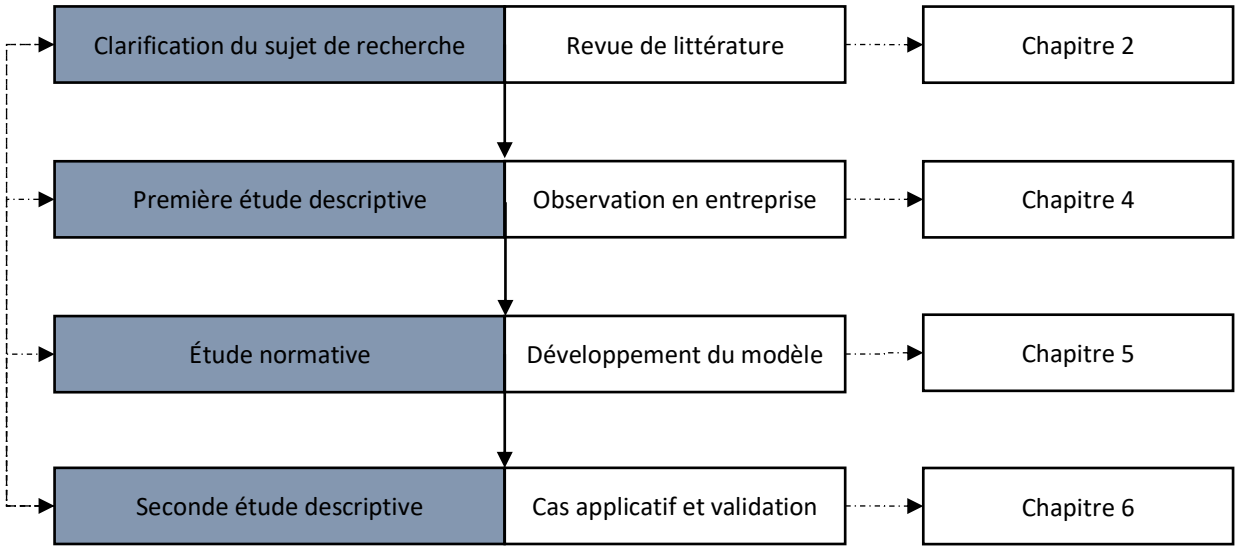


Figure 3.1: Méthodologie de la DRM

La méthodologie DRM est constituée de quatre étapes :

1. La première étape consiste en une **clarification du sujet de recherche**. Celle-ci a été réalisée au Chapitre 2 lors de la revue de littérature sur les méthodes de valorisation des données concernant les défauts sur les chaînes d'assemblage. Elle nous a permis, d'une part, de valider le besoin et la pertinence de notre projet, mais également d'identifier les techniques de RCA les plus pertinentes pour le développement de notre méthode.

2. La seconde étape de la DRM consiste en une **première étude descriptive.** Le sujet de recherche étant bien défini, le chercheur a donc un objectif clair à accomplir. Il s'agit d'une phase d'observation et d'analyse en entreprise, afin de déterminer les besoins et contraintes spécifiques du partenaire. Elle permet de confronter la littérature académique à la réalité du terrain des entreprises et donc de prendre du recul vis-à-vis des connaissances accumulées lors de la revue de littérature. Elle permet aussi de mettre en évidence les limitations des méthodes présentées dans la revue de littérature et donc de confirmer l'intérêt des contributions du projet pour le partenaire industriel. De plus, nous identifierons toutes les

bonnes pratiques implantées chez notre partenaire pour les incorporer à notre modèle. Cette étape fait l'objet du chapitre 4

3. La troisième étape de la DRM consiste en une **étude normative**. Durant cette étape, le chercheur développe un modèle répondant à la problématique de recherche selon les besoins et les contraintes des partenaires industrielles tout en exploitant les bonnes pratiques issues de la littérature scientifique. Ce modèle doit toutefois tenter de combler les lacunes connues de ces méthodes. Le chapitre 5 présente le modèle d'analyse causale développé dans le cadre de cette recherche.
4. Finalement, la quatrième étape de la DRM consiste en une **seconde étude descriptive**. Durant cette étude, on applique le modèle développé précédemment sur un cas d'entreprise, afin de tester sa performance (chapitre 6). Dans le cadre de ce mémoire, nous testerons également notre modèle sur un deuxième cas d'étude, afin d'évaluer sa généralisation dans un contexte de production par lot de fabrication (chapitre 7).

3.3 Conclusion

Ce chapitre nous a permis d'introduire notre objectif de recherche : **faciliter l'identification des causes racines des défauts sur les lignes d'assemblage multiproduit en exploitant les données disponibles**. De cette problématique découlent trois sous-objectifs :

-*déterminer les méthodes de préparation et d'analyse des données potentiellement applicables au domaine de production multiproduit pour la réalisation d'une analyse causale des défauts*,

-*développer une méthode d'analyse causale des défauts applicables aux lignes d'assemblage multiproduits*, et

-*mesurer le caractère de généralisation de la méthode proposée*.

Nous utiliserons une méthodologie de recherche de type DRM pour mener notre étude. Cette méthodologie est solidement ancrée sur la réalité de cas industriels. À l'aide des informations recueillies lors de la revue de littérature et de la première étude descriptive, nous développons une méthode d'identification des causes racines des défauts sur les lignes d'assemblage multiproduits. Nous présenterons lors du prochain chapitre la première étude descriptive du cas d'étude afin d'améliorer la compréhension du contexte ainsi que des besoins du partenaire.

CHAPITRE 4 PREMIÈRE ÉTUDE DESCRIPTIVE

La revue de littérature a déjà permis d'identifier les méthodes de RCA utilisées dans la littérature. Ce chapitre décrit la seconde étape de la méthodologie de recherche DRM exposée au chapitre précédent. Dans ce chapitre, nous présenterons dans un premier temps le contexte et l'étude de cas de notre premier partenaire industriel. Nous cherchons ainsi à identifier ses contraintes et ses besoins. Puis, nous reviendrons sur les stratégies d'identification des causes racines des défauts extraites de la revue de littératures. Finalement, nous présenterons les résultats de l'application de ces méthodes issues de la littérature sur les jeux de données de notre partenaire industriel afin d'en évaluer la pertinence et de guider le développement ultérieur de notre modèle.

4.1 Description de l'étude de cas

Les constructeurs automobiles font face à une concurrence de plus en plus forte. La qualité de production devient donc un des leviers importants pour améliorer la compétitivité à travers l'augmentation de la productivité. En effet, selon Töpfer (2017) en 2004, les entreprises du secteur automobile dépensaient plus de 1000 € par véhicule pour retravailler, voire mettre au rebut, les pièces défectueuses d'un produit lors des contrôles qualités et ce coût aurait encore augmenté depuis (Hirsch et al., 2018). Une bonne qualité de production permettrait donc d'éviter ces dépenses inutiles.

De plus, afin de répondre aux exigences toujours croissantes des clients, le concept de personnalisation en masse a été largement adopté dans ce secteur industriel. Cela permet au client de personnaliser totalement leur véhicule en fonction de leur besoin et de leur envie. Ce principe a pour effet une augmentation du nombre de références produites par les entreprises du secteur. Pour pouvoir continuer de produire à bas coût tout en répondant aux exigences des clients, les entreprises ont développé des lignes d'assemblage dites à modèles mixtes, appelées lignes d'assemblage multiproduits. Cependant, l'augmentation du nombre de références sur les lignes d'assemblage rend la détection des défauts et l'analyse de leur cause racine de plus en plus compliquée pour les experts.

C'est dans ce contexte que le projet de notre premier partenaire industriel s'inscrit. Il s'agit d'un constructeur automobile allemand, avec plus de deux millions de véhicules livrés en 2021 et un chiffre d'affaires de plusieurs milliards d'euros. Il fait partie des leaders mondiaux du marché.

Selon eux, les méthodes d'apprentissage automatisé seraient une solution potentielle pour identifier les causes racines des défauts rencontrés sur leur ligne d'assemblage. Pour ce faire, il a mis en place des outils informatiques qui collectent différentes données sur les lignes d'assemblage pour pouvoir étudier les défauts et identifier leurs causes racines. Le partenaire industriel ne s'intéresse pas aux causes racines évidentes, telles que des problèmes liés à des livraisons de composant défectueux puisque les méthodes classiques de RCA sont capables de facilement identifier ces problèmes.

Leur objectif est plutôt d'expliquer les défauts qui ont lieu sur leurs lignes d'assemblage et non pas de détecter les véhicules défectueux. Il cherche notamment à identifier des types de véhicules qui ont un pourcentage de défauts plus important. Ces véhicules seraient différenciés en fonction de leurs options et de leurs caractéristiques.

Pour réaliser notre étude, notre partenaire industriel nous a fourni différents types de données. Les données qualité nous indiquent pour chaque défaut l'identifiant de la voiture défectueuse et le nom et la catégorie du défaut rencontré. Les données comprennent aussi l'identifiant et les caractéristiques principales de chaque véhicule. Les données d'ordonnancement qui présentent la séquence de production des véhicules sont aussi disponibles.

4.2 Stratégie issue de la revue de littérature

Grâce à notre revue de littérature, nous avons pu identifier les méthodes pertinentes à appliquer dans notre cas d'étude. Tout d'abord, Sand et al. (2016), Hirsch et al. (2019), Schnell et al. (2019), et Mueller et al. (2018) basent toute leur méthode d'identification des causes racines des défauts sur des algorithmes d'arbres de décision. Sand et al. (2016), Hirsch et al. (2019) utilisent l'algorithme RF sur les paramètres de production de leurs produits défectueux ainsi que sur les résultats des tests EOL. Schnell et al. (2019), et Mueller et al. (2018) quant à eux, utilisent un algorithme d'arbres de décision sur les données de production de l'ensemble de leurs produits pour déterminer les causes racines des défauts, Mueller et al. (2018) utilisent l'algorithme C5.0.

De même, les algorithmes de partitionnement utilisés dans les études de cas de Sand et al. (2016), Hirsch et al. (2019), Sarkar (2004), Laxman et all. (2009) semblent faciliter l'identification des causes racines des défauts. En effet, en regroupant des produits ou des séquences d'évènements,

ces auteurs sont capables d'augmenter la quantité des données à analyser en considérant les données du groupe et non de l'individu.

Sarkar (2004) a pour sa part proposé une méthode basée sur le regroupement des séquences d'évènements menant à un défaut pour en identifier la cause racine. Il calcule la différence entre deux séquences à l'aide de l'algorithme d'alignement global de type Needleman-Wunsch. Une fois les regroupements de produits créés, il a ensuite utilisé ses connaissances en ingénierie pour examiner de plus près chacun de ces clusters afin d'identifier la cause racine de la panne. Si l'on remplace dans la méthode de Sakar les évènements par la production d'un véhicule, alors cette méthode semble applicable.

De plus, suite à une discussion avec notre partenaire industriel, nous avons accepté l'hypothèse que seule la voiture précédemment fabriquée pouvait avoir un impact sur la qualité de production d'un véhicule en cours de fabrication. Nous devons donc seulement considérer des séquences de deux voitures.

Cependant se pose la question de comment calculer la distance entre deux séquences. Contrairement à l'étude de cas de Sarkar qui a un nombre fini d'évènements possibles, nous avons un nombre infini de voitures 'constructibles'. Nous ne pouvons donc pas utiliser l'algorithme d'alignement global de type Needleman-Wunsch pour rassembler les voitures en différents groupes. Par ailleurs, Laxman et al. (2009) proposent une méthode similaire à la différence qu'ils utilisent une méthode heuristique pour déterminer quelle séquence d'évènements étudiés. Ils regroupent ces différentes séquences en groupe en fonction des défauts rencontrés. Afin d'identifier les causes racines des défauts, ils calculent pour chaque couple de défauts rencontré un indice de confiance qui permet d'identifier si un défaut en entraîne un autre. Tout comme la méthode précédente, cette méthode n'est pas applicable à notre cas d'étude pour les mêmes raisons.

Osama et Pantea (2020) combinent un algorithme ABOD et un Pareto pour identifier les causes racines des défauts. Cependant, cette méthode ne peut être appliquée sur notre cas d'étude. En effet, l'analyse de Osama et Pantea (2020) se base sur la détection des points aberrants d'un processus de fabrication de produits distincts. Il est donc nécessaire d'avoir suffisamment de données d'entrée par référence pour que la méthode fonctionne. Or dans notre cas, nous avons quasiment un nombre infini de produits. Il est donc impossible d'utiliser cette technique pour détecter les points aberrants du processus.

4.3 Résultat

Lors de la phase d'exploration, nous testons l'ensemble des méthodes identifié comme pertinentes lors de la revue de littérature pour vérifier leur pertinence.

Dans un premier temps, pour vérifier si nous pouvons concentrer notre analyse sur certains types de défauts ou de voitures, nous calculons deux métriques. La première est la probabilité qu'une voiture avec une caractéristique particulière soit défectueuse. Nous calculons cette métrique pour différentes caractéristiques : 1- le modèle de la voitures, 2- les variantes du modèle, 3- le type de moteur, 4-le type de direction, 5- type de boite de vitesse, et 6- le type de carrosserie. Nous représentons dans la figure ci-dessous, la métrique pour les deux première caractéristiques des voitures.

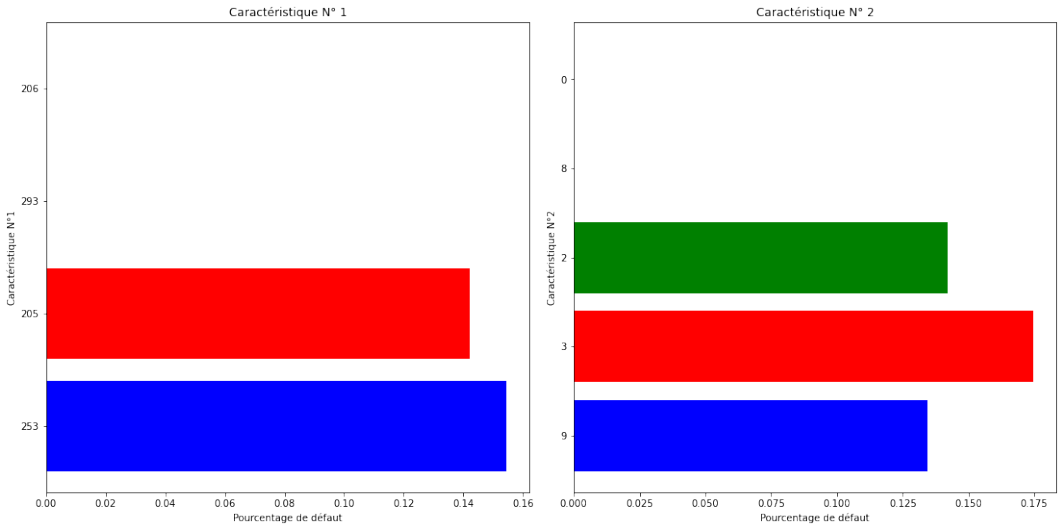


Figure 4.1: Pourcentage de défaut des voitures selon les caractéristiques des labels 1 et 2

On remarque, dans la figure 4.1, que les voitures avec 0% de défaut sur les deux premiers labels de chaque caractéristique sont exclusivement des voitures actuellement en démarrage ('*ramp-up'*), on a donc fait le choix d'ignorer les défauts sur ces véhicules.

Pour évaluer la probabilité conditionnée d'avoir un défaut, sachant une caractéristique, nous proposons la métrique M. Celle-ci correspond à la soustraction entre la probabilité d'assemblée une voiture défectueuse sachant que celle-ci possède une caractéristique en particulier et la probabilité d'assembler une voiture non défectueuse sachant que celle-ci possède la même

caractéristique que précédemment. Si on pose A l'évènement d'avoir la caractéristique i et D l'évènement d'avoir une voiture défectueuse, alors on a :

$$M = P(D|A) - P(\overline{D}|A)$$

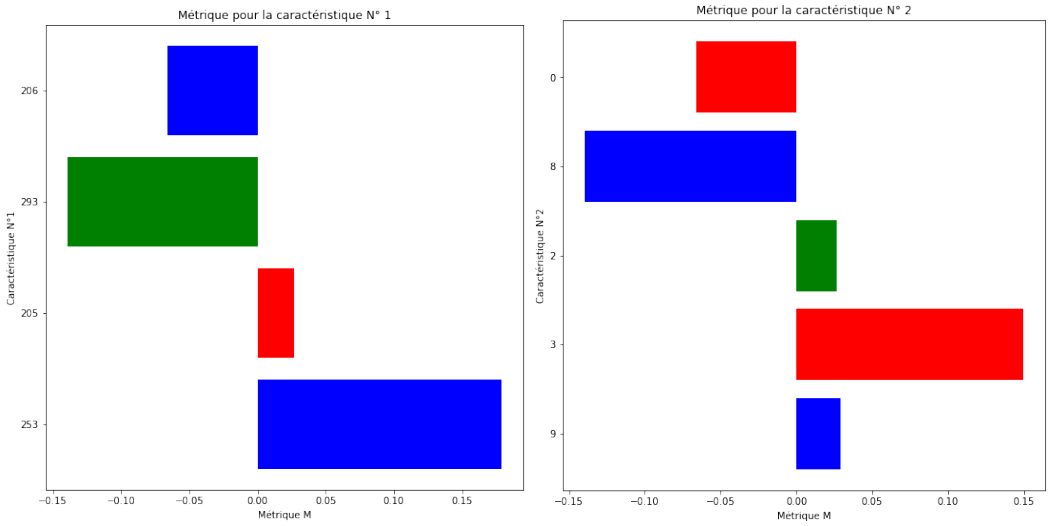


Figure 4.2: Métrique M des voitures selon les caractéristiques des labels 1 et 2

On en conclut que si l'on exclut les voitures en phase de démarrage (*ramp-up*), les défauts sont répartis sur l'ensemble des caractéristiques des voitures, et donc, on ne peut pas concentrer notre analyse sur une seule caractéristique en particulier.

Dans un second temps, on reprend les méthodes de Sand et al. (2016), Hirsch et al. (2019), Schnell et al. (2019), et Mueller et al. (2018) pour créer un arbre de décision basé sur les sept caractéristiques principales des voitures pour séparer les voitures défectueuses des voitures non défectueuses. On transforme les sept colonnes des caractéristiques en X colonnes, avec X le nombre de labels totaux sur les 7 colonnes contenant des données binaires précisant si une voiture possède le label ou non. Voici un exemple pour une colonne dans la figure ci-dessous. Nous positionnons la transformation de chaque colonnes côte à côte.

Voiture	Caractéristique 1
1	A
2	B
3	C

Transformation

Voiture	A	B	C
1	1	0	0
2	0	1	0
3	0	0	1

Figure 4.3: Transformation des données catégorielles en données binaires

On utilise comme algorithme d'arbre de décision une version optimisée de CART disponible sur la bibliothèque python sklearn. On obtient l'arbre de décision de la figure 4.4 avec un score de précision de 0,87. Pour cette représentation, on utilise comme critère d'arrêt une décroissance minimum du facteur d'impureté de 0,001. La version complète de l'arbre de décision est disponible en annexe.

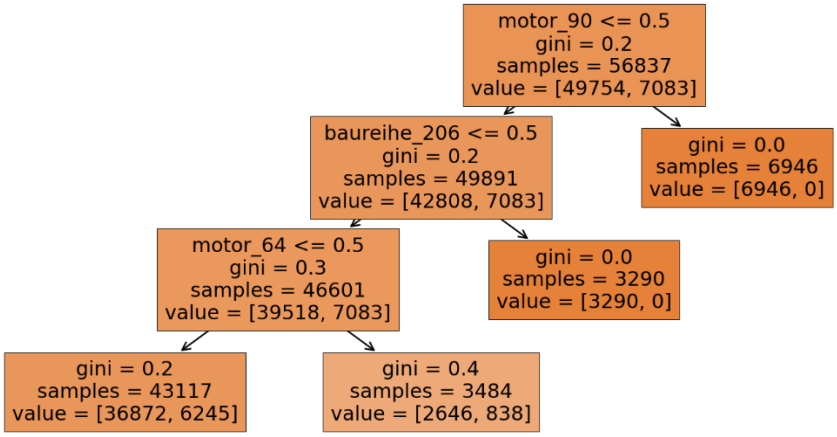


Figure 4.4: Arbre de décision séparant les voitures défectueuses et non défectueuses

Cet arbre ne nous permet pas d'identifier un ensemble de caractéristiques qui expliqueraient les défauts.

Par la suite, toujours en utilisant un arbre de décision, on propose de rassembler les différents types de défauts dans des groupes différents et de prédire à quel groupe de défauts appartient une voiture en fonction de ces caractéristiques pour déterminer si un ensemble de caractéristiques expliquerait ce groupe de défauts.

Pour regrouper les défauts entre eux, on utilise le code alphanumérique rattaché à chaque type de défaut. Dans un premier temps, on transforme ce code de cinq caractères en une valeur numérique. Celle-ci vaut la somme de chaque caractère transformé à l'aide de la transformation sur la base 36, multiplier par 36^{5-x} avec x la position du caractère dans le code. Par exemple, le code : 64an2 devinent :

$$Valeur_{64an2} = 6 * 36^4 + 4 * 36^3 + 10 * 36^2 + 23 * 36^1 + 2 * 36^0 = 10278110$$

On utilise un algorithme de partitionnement hiérarchique afin de regrouper nos types de

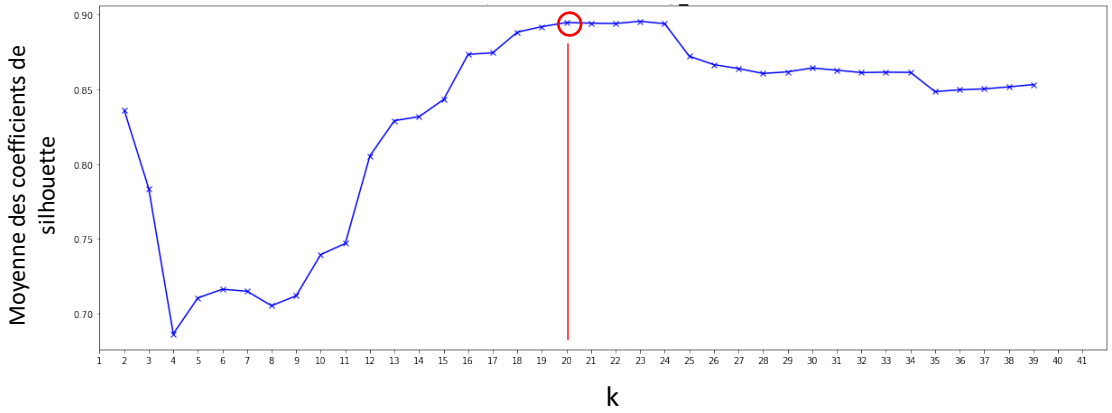


Figure 4.5: Méthode de la silhouette pour déterminer le nombre optimal de clusters de type de défauts

défauts dans différents groupes. On définit la distance entre deux codes comme étant la valeur absolue de la différence entre les valeurs de ces codes. Pour déterminer le nombre optimal de clusters, on utilise la méthode de la silhouette. On obtient la courbe de la valeur moyenne du score de la silhouette en fonction du nombre de clusters dans la figure 4.5. On pose ici le nombre optimal de clusters à 20.

On rattache chaque véhicule défectueux à son groupe en fonction du type de défaut rencontré lors de l'assemblage; si un véhicule n'est pas défectueux, on lui affecte le groupe 0.

On réutilise l'algorithme de sklearn pour prédire à quel groupe de défaut appartient l'ensemble des voitures en fonction des caractéristiques des véhicules (Figure 4.6). De même, on essaye de prédire quel type de défauts rencontre une voiture défectueuse en fonction seulement des caractéristiques des véhicules défectueux (Figure 4.7). Pour ces représentations, nous utilisons comme critère d'arrêt une décroissance minimum du facteur d'impureté de 0,001.

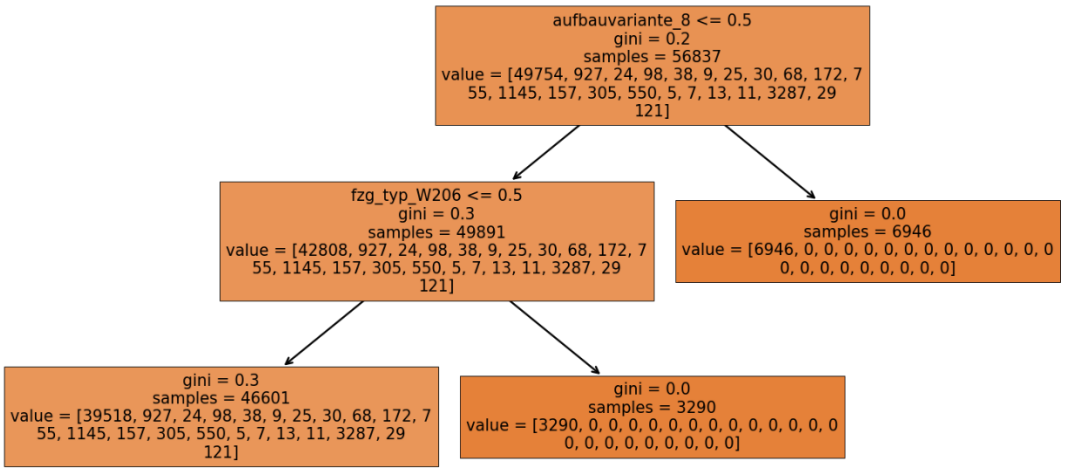


Figure 4.6: Arbre de décision séparant les voitures selon leur différent type de défauts

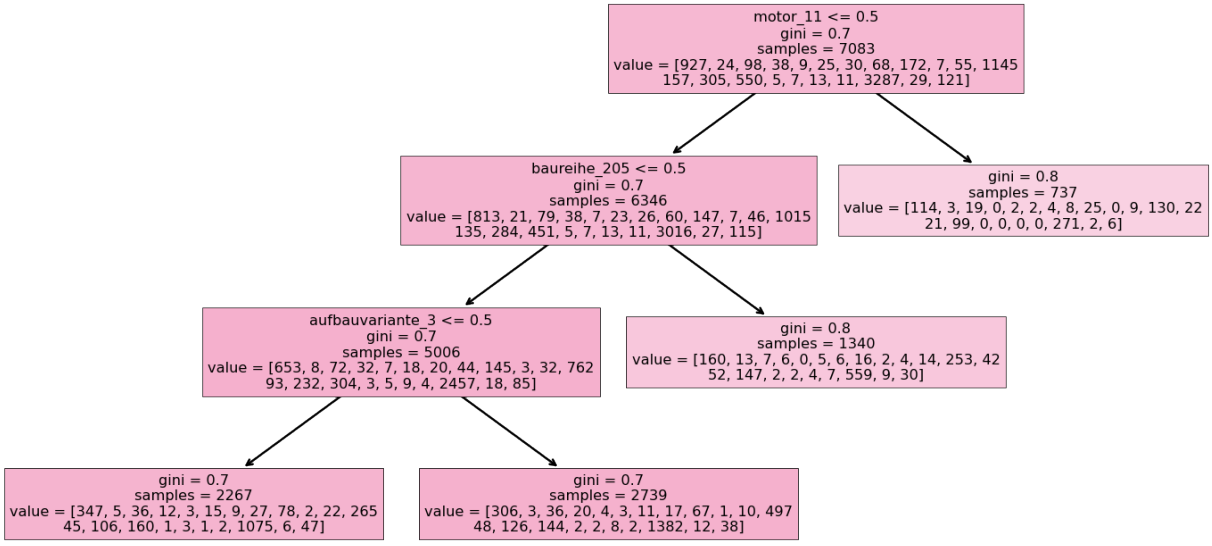


Figure 4.7: Arbre de décision séparant les voitures défectueuses selon leur type de défauts

Dans les deux arbres de décisions, nous ne sommes pas capables d'identifier un ensemble de caractéristiques qui expliquerait l'appartenance de certaines voitures à un type de défauts en particulier rencontré lors de l'assemblage.

Une autre solution utilisée dans la littérature est de regrouper les produits dans des groupes équivalents pour faciliter l'analyse des causes racines. On essaye de regrouper nos voitures en fonction de leurs caractéristiques principales. Pour ce faire, on réutilise la transformation des

caractéristiques principales présentée précédemment. On a en tout 198 combinaisons possibles dans la base de données si l'on considère seulement ces 7 caractéristiques. On utilise la méthode de la silhouette pour déterminer le nombre de clusters optimal, tel que présenté à la figure 4.8.

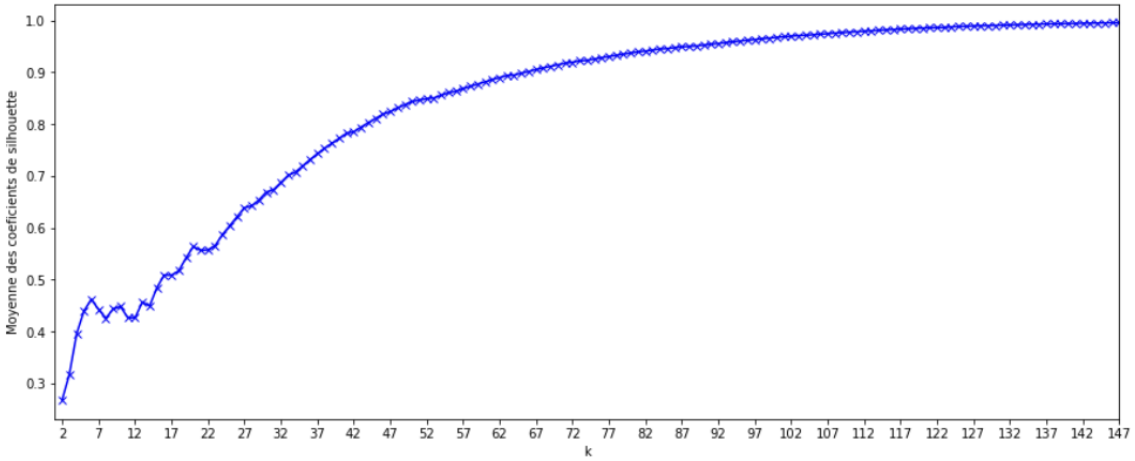


Figure 4.8: Méthode de la silhouette pour séparer les voitures selon leurs caractéristiques (données catégorielles)

Cette méthode ne nous permet pas de déterminer un nombre de clusters optimal pertinent. En effet, l'algorithme sépare chaque combinaison possible. Patnaik et al. (2012) proposent une méthode pour réaliser des partitionnements de données catégorielles. Leur méthode se base sur le remplacement des catégories par la probabilité d'avoir cette catégorie dans le jeu de données. On obtient donc avec cette méthode le graphique de la silhouette présenté à la Figure 4.9. Cependant, nous arrivons aux mêmes conclusions que précédemment.

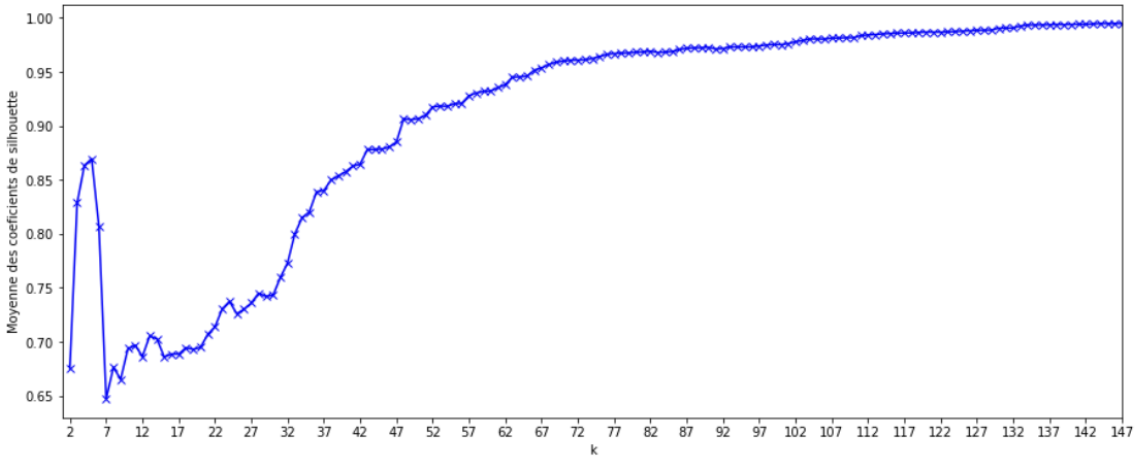


Figure 4.9:Méthode de la silhouette pour séparer les voitures selon leurs caractéristiques (probabilité)

4.4 Recommandations

On peut tirer de la revue de littératures et des expériences réalisées lors de la première étude descriptive quatre recommandations principales.

- Schnell et al. (2019), Mueller et al. (2018), Sand et al. (2016), Hirsch et al. (2019) utilisent tous des algorithmes d'arbres de décision pour identifier les causes racines des défauts rencontrées dans leur cas d'étude. En effet, en raison de leur facilité d'interprétation et leur représentation graphique sous la forme d'un arbre, ces méthodes sont facilement interprétables et permettent donc facilement d'identifier les causes racines des défauts. On a utilisé ces algorithmes d'arbres de décision lors de la phase d'exploration des données afin de classifier nos produits en fonction des caractéristiques des voitures pour prédire si une voiture est défectueuse ou non. Cependant, l'application directe de cette méthode ne nous permet pas d'identifier clairement des causes racines des défauts. On explique cela par le fait que le calcul de distance basé sur ces données n'a pas de sens « physique ». Conséquemment, on doit donc chercher une nouvelle façon de calculer la distance entre nos produits;
- De même, les algorithmes de partitionnement utilisés dans les études de cas de Sand et al. (2016), Hirsch et al. (2019), Sarkar (2004), Laxman et all. (2009) semblent faciliter l'identification des causes racines des défauts. En effet, en regroupant des produits ou des séquences d'évènements, ces auteurs sont capables d'augmenter la quantité de données à analyser en considérant les données du groupe et non de l'individu. Pour les mêmes raisons que le cas précédent, ces méthodes ne nous donnent pas des groupes de produit intéressants à étudier, car la distance entre deux produits n'a pas de sens « physique ». De même, lors de la phase d'exploration des données, nous avons essayé de regrouper nos défauts dans différentes catégories afin d'identifier si certains types de défauts étaient liés à certaines caractéristiques de voitures. Cependant, cette analyse ne nous permet pas de lier des défauts à des caractéristiques en particulier;
- Sarkar (2004) et Laxman et al. (2009) proposent des méthodes basées sur le regroupement de séquence d'évènements menant à un défaut. Même si les méthodes proposées ne sont pas applicables, il est intéressant de considérer la possibilité qu'une succession

d'évènements soit à l'origine d'un défaut. Dans notre cas, on pourrait étudier la succession de la production de deux véhicules; et

- Finalement, on rappelle que six articles qui traitent de l'identification des causes racines utilisent des experts pour confirmer les causes racines que leur méthode a identifiées. On peut donc considérer que ces méthodes fournissent seulement des pistes de réflexion pour les experts pour identifier de nouvelles causes racines.

4.5 Conclusion

L'objectif de notre partenaire industriel est donc d'expliquer les défauts qui ont lieu sur sa ligne d'assemblage. Le caractère multiproduit de celle-ci est la contrainte principale de notre partenaire industriel. Cette caractéristique, nous empêchant d'appliquer directement les méthodes d'identification des causes racines des défauts sur les lignes d'assemblage identifié dans la revue de littératures. En effet, celles-ci sont développées pour analyser les défauts d'un produit unique et non plusieurs produits distincts. Cependant, on peut extraire de ces méthodes plusieurs recommandations. Grâce à ces différentes recommandations, nous avons créé un modèle d'identification des causes racines que nous présenterons dans le prochain chapitre.

CHAPITRE 5 MODÈLE D'ANALYSE CAUSALE

Ce chapitre présente le développement du modèle proposé avec une description de chacune de ses composantes. Il comprend une description des activités initiales, la collecte de l'ensemble de données pertinentes à l'analyse et la préparation des données dans la section 5.2 . La section 5.3 présente les opérations de partitionnement réalisé sur nos données pour regrouper nos produits dans des clusters basés sur des données de production. Ensuite, la section 5.4 aborde l'utilisation d'analyse statistique des données afin d'identifier les clusters de produit responsable des défauts. Finalement, la section 5.6 explique la méthode d'analyse des clusters sélectionnés permettant d'identifier les causes racines des défauts.

5.1 Présentation du modèle

Notre modèle est basé sur une succession d'algorithmes d'apprentissage automatisé qui permet d'identifier les causes racines des défauts. Le modèle complet est illustré à la Figure 5.1.

Les trois premières étapes correspondent à la première phase et visent à préparer les données issues des lignes d'assemblage multiproduits. Ainsi, une fois la problématique de l'entreprise comprise (étape 1) et la compréhension des données disponibles (étape 2), nous préparons celle-ci pour l'analyse (étape 3).

Puis, nous utilisons un algorithme de partitionnement hiérarchique des données afin de regrouper les produits étudiés dans différents clusters (étape 4). Nous analysons le taux de défauts des produits en fonction de leurs clusters (étape 5.1) et de l'ordonnancement de la production (étape 5.2). Ces analyses nous permettront par la suite d'identifier des clusters de produits avec des taux de défaut importants. On désigne ces regroupements de produits comme des regroupements « problématiques » (étape 6). Cette deuxième phase, qui comprend les étapes quatre à six, permet ainsi d'identifier les causes racines des défauts.

La troisième phase vise à expliquer pourquoi ces regroupements sont « problématiques ». Nous proposons ici d'utiliser un algorithme d'arbre de décision pour séparer les produits « problématiques » des autres (étape 7). Ces arbres de décision permettent aux opérateurs d'identifier les caractéristiques propres au regroupement de produits dits « problématique » , et donc, d'identifier des causes de l'origine du taux de défauts important sur ces produits. Ceci permet aux opérateurs de prendre des actions correctives pour supprimer la cause des défauts (étape 8).

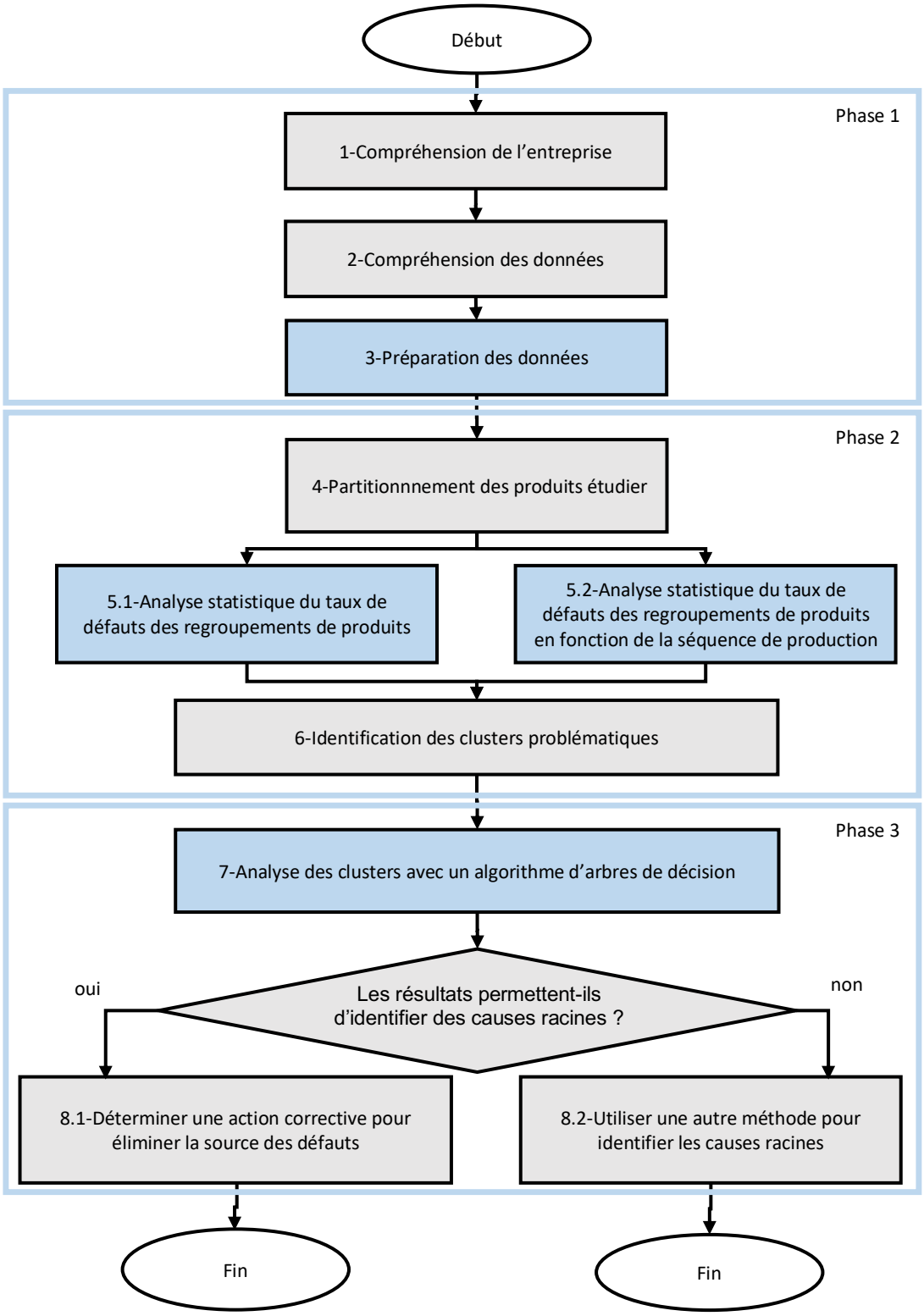


Figure 5.1 : Modèle générique proposé pour l'analyse causale des défauts au sein d'une chaîne d'assemblage multiproduit

5.2 Compréhension de l'entreprise et des données

L'étape 1, la **compréhension de l'entreprise** consiste à déterminer les objectifs de l'étude. On y définit la ligne d'assemblage ou la section de ligne que nous étudierons et l'intervalle de temps considéré pour l'étude. On relève toutes les problématiques liées à la ligne d'assemblage analysée, des produits particuliers, des évènements de maintenance, des problèmes d'approvisionnement, etc. L'objectif est d'identifier tous les évènements qui pourraient avoir un impact sur les résultats de notre analyse et qui ne seraient pas pertinents de considérer.

Une fois le cadre de notre analyse défini, on peut commencer l'étape 2, **la compréhension des données.** Cette étape implique l'étude des données disponibles pour l'analyse. Cette étape est déterminante pour la suite du projet, car elle permet d'éviter les problèmes au cours des étapes suivantes, notamment l'étape de préparation des données. Les types de données extraites des lignes de production multiproduit peuvent varier d'un cas à un autre. Cependant, certains types de données sont obligatoires pour pouvoir mener notre analyse. Idéalement, nous devons avoir à disposition quatre types ou tables de données :

- Une table nous permettant d'identifier quels produits sont défectueux. Il est ici nécessaire de faire la différence entre les différents types de défauts rencontrés lors de l'assemblage des produits et d'être capable de les localiser à la fois, si possible, dans le temps et dans l'espace :
- Une table nous donnant la séquence de production de la ligne d'assemblage :
- Une table caractérisant les produits, tels que les spécifications et caractéristiques des produits, leurs différentes options, etc
- Une table de données décrivant le processus d'assemblage des produits tel que le détail des opérations réalisées sur les produits, les temps de production des produits sur la ligne d'assemblage (planifiés et réels), les paramètres de production utilisés lors de l'assemblage du produit, etc.

À la fin de ces deux étapes, nous devrions être en position pour clairement définir le cadre de notre analyse, d'identifier les évènements pertinents qui pourrait avoir un impact néfaste sur les résultats de notre analyse, en plus d'avoir pris connaissance des données disponibles pour l'analyse.

5.3 Préparation des données

Une fois les données étudiées, nous préparons les données pour l'analyse (étape 3). On prend en compte, lors de la préparation des données, les évènements problématiques identifiés lors de la première étape afin de minimiser leur impact sur les résultats de l'analyse. Par exemple, si lors de notre intervalle de temps considéré l'entreprise a réalisé des opérations de maintenance qui ont entraîné une dégradation de la production, il est nécessaire d'exclure l'intervalle de temps correspondant à ces opérations de maintenance pour ne pas polluer nos résultats.

Concernant les données décrivant le processus d'assemblage, si ces dernières sont numériques, tels que des temps de production à chaque poste de travail le long de la ligne, nous devons les normaliser à l'aide de la normalisation Min/max. Si ces données sont catégorielles, on les transforme en données binaires. Ces opérations permettent de faciliter le partitionnement des produits dans l'étape 4.

Si les données caractérisant les produits sont des données catégorielles, on les transforme en données binaires pour faciliter leur maniement et la compréhension des clusters lors de l'étape 7.

5.4 Partitionnements des produits à étudier

Une fois les données préparées, nous réalisons un partitionnement des produits afin de les regrouper dans des clusters qui font sens vis-à-vis du processus d'assemblage.

Pour ce faire, nous utilisons un algorithme de partitionnement hiérarchique. Cet algorithme utilise la distance euclidienne pour calculer la distance entre les produits. Les algorithmes de partitionnement hiérarchique établissent des relations entre les données sous la forme d'un arbre de classement qui est représenté sous la forme d'un dendrogramme. Dans notre cas, nous utilisons une logique ascendante: chaque produit commence dans son propre cluster et les clusters sont successivement fusionnés. Le critère de liaison détermine la métrique utilisée pour la stratégie de fusion. Dans notre cas, nous cherchons à minimiser la somme des différences des « distances » au carré entre les produits de chaque cluster. L'un des avantages des algorithmes de partitionnement hiérarchique est leurs capacités à repérer des groupes de données de forme quelconque. Cependant, ils sont relativement lents et gourmands par rapport aux autres algorithmes de partitionnement.

Pour pouvoir utiliser cet algorithme de partitionnement, nous avons besoin des données que l'on utilise pour séparer nos produits et le nombre optimal de clusters.

On utilise comme données d'entrée pour notre algorithme les données préparées caractérisant le processus de production des produits le long de la ligne d'assemblage.

Afin de déterminer le nombre optimal de clusters, nous utilisons la méthode de la silhouette. Cette méthode consiste à calculer pour chaque nombre de clusters possible la moyenne des coefficients de silhouette de l'ensemble des produits étudiés puis à déterminer le nombre de clusters qui fournit la moyenne la plus proche de 1. Le coefficient de silhouette est une mesure de la qualité d'une partition d'un ensemble de données. Pour un produit, le coefficient de silhouette est la différence entre la distance moyenne avec les produits du même groupe que lui (cohésion) et la distance moyenne avec les produits des autres groupes voisins (séparation). Le coefficient de silhouette prend une valeur entre 1 et -1. Un coefficient proche de -1 signifie que le point est en moyenne plus proche du groupe voisin que du sien; il est donc mal classé. À l'inverse, un coefficient proche de 1 signifie que le point est en moyenne plus proche de son groupe que des groupes voisins; il est donc bien classé. Le nombre optimal de clusters est donc obtenu avec le k qui donne la moyenne des coefficients de silhouette de l'ensemble des produits le plus proche de 1.

Pour calculer le coefficient de silhouette, nous le définissons d'abord sur un point i dont le groupe est k=C(i). Il se base sur la distance moyenne du point à son groupe :

$$a(i) = \frac{1}{|I_k| - 1} \sum_{j \in I_k, j \neq i} d(x^i, x^j)$$

et la distance moyenne du point à son groupe voisin :

$$b(i) = min_{k\prime \neq k} \frac{1}{|I_{k\prime}|} \sum_{i\prime \in I_{k\prime}} d(x^i, x^{i\prime})$$

Le coefficient de silhouette du point i s'écrit alors :

$$s_{sil}(i) = \frac{b(i) - a(i)}{max(a(i), b(i))}$$

On peut le moyenner groupe par groupe pour comparer leurs homogénéités : ceux avec les coefficients de silhouette les plus forts sont les plus homogènes. Sur l'ensemble de la classification, il aura pour expression:

$$S_{sil}(i) = \frac{1}{K}\sum_{k=1}^{K}\frac{1}{|I_k|}\sum_{i\prime\in I_k} s_{sil}(i)$$

À la fin de cette étape, nous avons donc des clusters de produits qui font sens vis-à-vis du processus d'assemblage. Par la suite, nous ferons une analyse statistique des défauts en fonction des clusters et de l'ordonnancement de la production.

5.5 Analyse statistique des défauts et identification des clusters problématiques

Pour pouvoir identifier les clusters dits problématiques, nous basons notre sélection sur deux analyses. La première consiste simplement à calculer le pourcentage de produit défectueux dans chaques clusters obtenu lors de l'étape 4 et de comparer les taux de défaut des clusters au taux de défaut moyen du jeu de données. On considère un produit comme défectueux à partir du moment que celui-ci rencontre un défaut au cours de son processus d'assemblage. Nous pouvons affiner notre analyse en considérant seulement un certain type de défaut.

La seconde analyse consiste à considérer en plus la séquence de fabrication. Nous faisons l'hypothèse que lors de l'assemblage d'un produit, seul le produit construit précédemment peut avoir un impact sur la qualité du produit en cours de production. Nous calculons donc pour chaque couple de clusters de produit A-B crée lors de l'étape 4, le taux de produits défectueux du cluster A assemblés après un produit du cluster B.

De plus, pour seulement avoir des résultats probants, nous considérons seulement les clusters et les transitions entre les clusters qui ont un nombre d'occurrences minimales dans la séquence de production. Cette limite est fixée arbitrairement à 10.

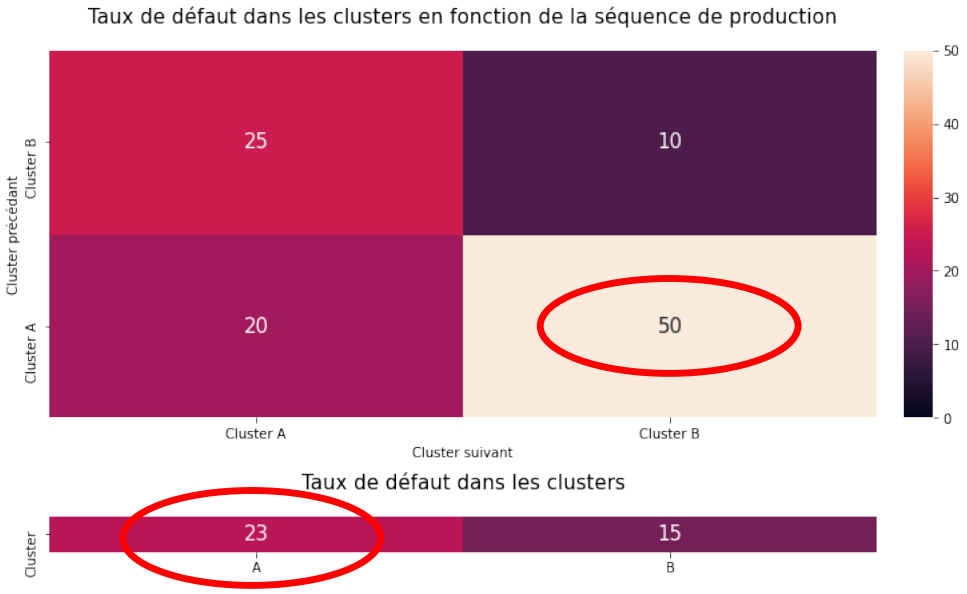


Figure 5.2: Exemple de graphique utilisé pour la sélection des clusters problématiques

Pour faciliter la sélection des groupes de produit dit « problématique », on regroupe nos deux analyses sous la forme d'un couple de cartes de densité présenté à la figure 5.2. Si l'on considère que la figure 5.2 est extraite d'un jeu de données avec un taux de défaut de 18%, alors on peut considérer que le cluster A est problématique, car son taux de défaut est bien plus important que le taux de défaut du jeu de données. De même, on remarque que la transition du cluster A vers le cluster B pose problème, puisque la moitié de nos produits du cluster B son défectueux lorsqu'ils sont fabriqués après un produit du cluster A, ce qui est très supérieur au taux de défaut du cluster B de 15%. On en conclut que les clusters A et B sont problématiques, car le cluster A et la transition du cluster A vers le cluster B ont toutes deux un taux de défauts important.

5.6 Analyse des clusters problématiques

Une fois les clusters problématiques identifiés, nous donnons du sens à ces clusters en utilisant un algorithme d'arbre de décision pour les classifier en fonction de données caractérisant nos produits.

Pour ce faire, on utilise l'algorithme CART dont l'acronyme signifie « *Classification and Regression Trees* ». Cette méthode se base sur une succession d'arbres qui cherche à diviser localement les données en plus petits segments en minimisant un critère d'évaluation. L'algorithme CART utilise comme critères d'évaluation l'indice de diversité de Gini qui mesure avec quelle fréquence un élément aléatoire de l'ensemble serait mal classé si son étiquette était choisie

aléatoirement selon la distribution des étiquettes dans le sous-ensemble. Il atteint sa valeur minimale de zéros lorsque tous les éléments de l'ensemble sont dans une même classe de la variable cible. Il en résulte un arbre de décision représenté par une série de divisions binaires débouchant sur des nœuds terminaux qui peuvent être décrits par un ensemble de règles spécifiques.

Grâce à ces règles spécifiques, nous sommes capables d'expliquer l'appartenance d'un produit à un cluster problématique et donc d'avoir des pistes de réflexion sur les causes racines des défauts.

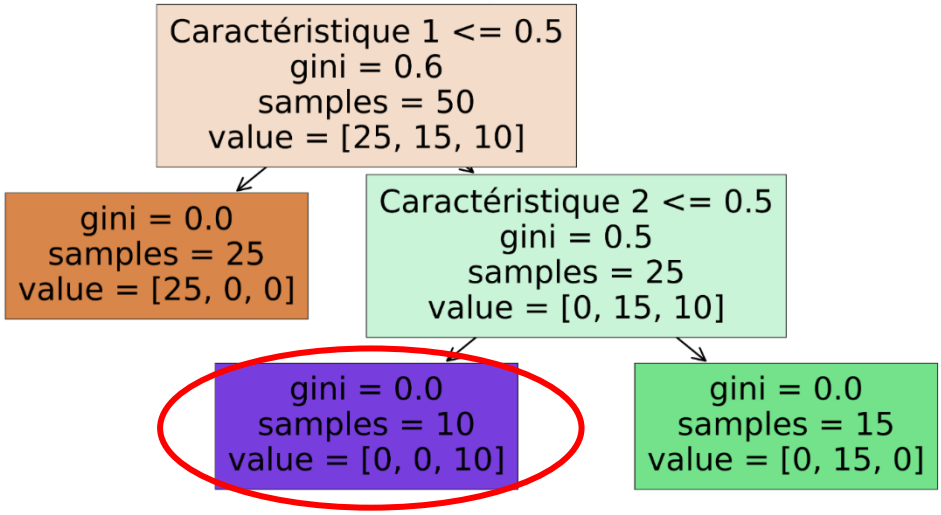


Figure 5.3 : Exemple d'un arbre de décision utilisé pour « expliquer » les clusters problématiques

Prenons l'exemple de la figure 5.3. Si l'on pose que le cluster « problématique » est le cluster 3, le nœud représenté en violet dans la figure, nous pouvons déduire que les produit problématiques sont les produits qui ont la caractéristique 1 mais n'ont pas la caractéristique 2.

Ces pistes de réflexion seront présentées aux experts qui pourront les utiliser afin de guider leur analyse des causes racines des défauts.

5.7 Conclusion

Notre modèle se base donc sur la succession de plusieurs méthodes d'apprentissage automatisé, un algorithme de regroupement hiérarchique pour regrouper les produits dans différents clusters, une analyse statistique des taux de défaut en fonction des clusters et de la séquence de production, et enfin, une analyse des clusters avec l'algorithme de classification CART. Dans le chapitre suivant, nous exécutons le modèle proposé sur l'ensemble de données collectées auprès de notre premier partenaire industriel afin de valider la pertinence de notre méthode.

CHAPITRE 6 APPLICATION DU MODÈLE ET VALIDATION

Les chapitres précédents ont permis de définir les concepts et le fonctionnement de notre modèle en tant que nouvelle méthode d'identification des causes racines des défauts sur les lignes d'assemblage multiproduit. Ce chapitre est dédié à la démonstration pratique du modèle en l'appliquant à un jeu de données fournies par notre partenaire industriel.

6.1 Compréhension de l'entreprise et des données

Premièrement, nous essayons de comprendre les objectifs de notre partenaire industriel et les particularités de sa ligne d'assemblage (étape 1). La ligne d'assemblage étudiée est une ligne d'assemblage automobile multiproduit. L'objectif de notre partenaire industriel est d'identifier les causes racines des défauts d'une ligne d'assemblage d'une de ces usines. Cette ligne d'assemblage est découpée en plusieurs sections. Dans notre cas, nous nous intéresserons à une seule section de la ligne, qui est elle-même découpée en quatre sous-sections. De plus, nous limitons notre analyse dans le temps en considérant seulement les voitures construites entre le premier janvier et le premier avril 2021, ce qui correspond à 63 111 voitures. On note que lors de cet intervalle de temps, deux modèles de voiture sont en cours de montée en puissance (*ramp-up*) en production.

Puis, nous étudions l'ensemble des données disponibles pour notre analyse (étape 2). Celle-ci provient de deux sources différentes. La première consiste à extraire de quatre fichiers Excel, un pour chaque mois de production de notre fenêtre d'étude, des données de production relatives à la section de la ligne d'assemblage étudiée. La seconde consiste à prélever des données du progiciel de gestion intégrée de l'entreprise.

Chaque fichier Excel traite un mois de production et est découpé en cinq feuilles de calcul qui fournit chacune un certain type de données.

La première feuille de calcul nous donne, pour chaque voiture produite, le numéro d'identification du véhicule (VIN) ainsi que l'horodatage du début de la production du véhicule dans la portion de la ligne d'assemblage étudiée.

La feuille de calcul suivante nous donne le nom et l'identifiant de toutes les opérations réalisées sur la ligne.

La troisième feuille fournit des informations sur les postes de travail de la ligne. On y retrouve un numéro d'identifiant pour chaque poste de travail, le nom de celui-ci et la liste des identifiants des opérations réalisées à chaque poste.

La quatrième feuille de calcul est un tableau à doubles entrées qui nous donne la charge de travail théorique nécessaire à la production d'un véhicule pour chaque poste de la ligne d'assemblage.

La dernière feuille de calcul est aussi un tableau à doubles entrées qui indique quelles sont les opérations réalisées sur chaque voiture.

Finalement, nous extrayons de l'ERP de l'entreprise les données relatives aux défauts de production, ainsi que celles relatives aux caractéristiques des produits. On obtient trois tables de données : l'une recense les défauts dits mécaniques, l'autre traite des défauts électriques et la dernière contient les caractéristiques de l'ensemble des voitures produites durant notre fenêtre de temps d'analyse. Chaque défaut est associé aux numéros VIN de la voiture et est rattaché à un type de défaut déjà créé par l'entreprise.

6.2 Préparation des données

Une fois que le cadre de notre analyse et les données disponibles sont bien définis, nous préparons ces données pour lancer les analyses (étape 3).

Dans un premier temps, nous transformons chaque feuille Excel en un fichier csv, afin de faciliter leur maniement. Puis, nous fusionnons les fichiers csv en fonction du type de feuille de calcul dont ils sont extraits. Ainsi, nous obtenons cinq tables de données qui regroupent par type de données l'ensemble des informations extraites des fichiers Excel.

La fusion des données des premières feuilles de calcul est stockée dans **Prod_seq_data**. Cette table contient l'ensemble des numéros VIN des voitures produites durant notre fenêtre d'analyse ainsi que l'horodatage du début de la production des voitures sur notre section de ligne d'assemblage. On a donc une table de données de deux colonnes que l'on trie selon l'horodatage du début de la production pour avoir la séquence de production sur notre ligne d'assemblage.

Les données extraites des secondes feuilles de calcul sont stockées dans une table de donnée que l'on nomme **Operation_data.** On y retrouve les identifiants et le nom des 6 520 opérations

réalisées le long de la section de la ligne d'assemblage étudiée, ainsi qu'une colonne précisant dans quelle sous-section de la ligne d'assemblage est réalisée l'opération.

La table de données **WP_data** est créée à partir de la fusion des troisièmes feuilles de calcul. Elle fournit pour chacun des 103 postes de travail situés le long de la section de la ligne d'assemblage le nom de la station de travail, l'ensemble des opérations réalisées à chaque poste et la sous-section d'appartenance du poste de travail.

La concaténation des tableaux à double entrées donne la charge de travail de chaque poste pour l'ensemble des voitures est stockée sous le nom de **WS_load_data.** On a donc pour chaque véhicule la charge de travail associée à chaque poste. La valeur moyenne de la charge de travail des postes est d'une unité de temps et varie entre 0 et 5 unités de temps par poste de travail. Nous normalisons ces données avec la méthode de normalisation min/max pour préparer leur utilisation lors de l'étape 4. On identifie pour chaque poste de travail à quelle sous-section celui-ci appartient grâce aux données de la table de données WP_data. Finalement, pour chaque sous-section, on calcule la somme de la charge de travail normalisée par voiture et on stocke ces quatre valeurs à la fin de la table WS_Load_data.

Finalement, les données relatives aux opérations réalisées sur les voitures sont stockées dans la table **AVO_data.** Cette table de données est un tableau à double entrées qui nous donne pour chaque opération parmi les 6 520 réalisée, si celle-ci est réalisée sur une voiture ou non sur une voiture. En moyenne, 692 opérations sont réalisées par voiture. On identifie dans quelle sous-section appartient chaque opération en croissant la table AVO_data, avec la colonne précisant la sous-section de chaque opération dans la table de données Operation_data.

Dans un second temps, on fusionne les tables de données traitant des défauts en une seule et même table que l'on nomme **Q_data**. On rencontre en tout 11 265 défauts sur la section de la ligne d'assemblage entre janvier et avril 2021. Ces défauts sont répartis entre 959 catégories de défauts.

À la suite d'une discussion avec notre partenaire industriel, on décide d'ignorer les défauts sur les deux modèles de voiture en cours de démarrage en production. On n'a plus que 7 083 défauts répartis sur 577 catégories de défauts dans notre base de données.

Après avoir déterminé la liste des codes VIN des voitures produit, on extrait de l'ERP l'ensemble des données caractérisant les voitures produites dont le numéro VIN est dans la table de données Prod_seq_data. On stocke ces données dans la table **Cara** qui nous indique, pour chaque voiture

produite, huit des caractéristiques principales de la voiture, telles que la référence du moteur, la ligne de la voiture ou bien sa couleur. On transforme ces huit colonnes en 155 colonnes, une pour chaque label des huit colonnes précédentes. Ces nouvelles colonnes contiennent des variables binaires qui valent 1 si la voiture possède le label de la colonne, 0 sinon.

La figure 6.1 nous donne un résumé des tables de données de l'étude après leur prétraitement.

Une fois que l'ensemble des données est préparé et trié, on peut commencer à regrouper nos produits dans différents clusters (étape 4).

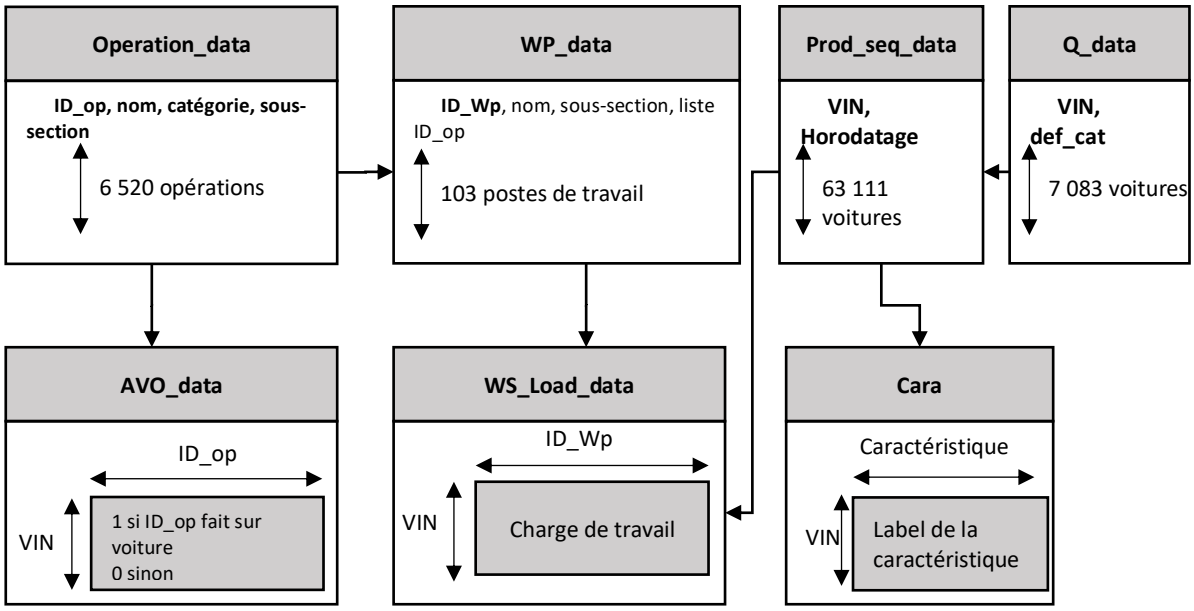


Figure 6.1: Représentation des données de la première étude de cas

6.3 Sélection et regroupement des produits à analyser

Dans notre cas d'analyse, on comprend bien la nécessité de regrouper les voitures dans des clusters. En effet, on ne peut pas concentrer notre étude sur un certain type de modèle de voiture, puisque toutes les voitures produites sont quasiment uniques. Il est donc nécessaire de les regrouper en clusters. Nous utiliserons donc l'ensemble des données disponibles caractérisant le processus d'assemblage pour réaliser nos différentes opérations de partitionnement.

Premièrement, nous utilisons les données de la table WS_Load_data comme données d'entrée pour l'algorithme de partition hiérarchique afin de créer les clusters. Puis, nous créons quatre clusters de produits en fonction des données de chaque sous-section de la ligne d'assemblage. On utilise

comme données d'entrée pour l'algorithme de partitionnement la charge normalisée des postes de travail de chaque sous-section. Finalement, on utilise les quatre dernières valeurs de la table de données WS_Load_data pour créer un dernier cluster basé sur la charge de la ligne d'assemblage.

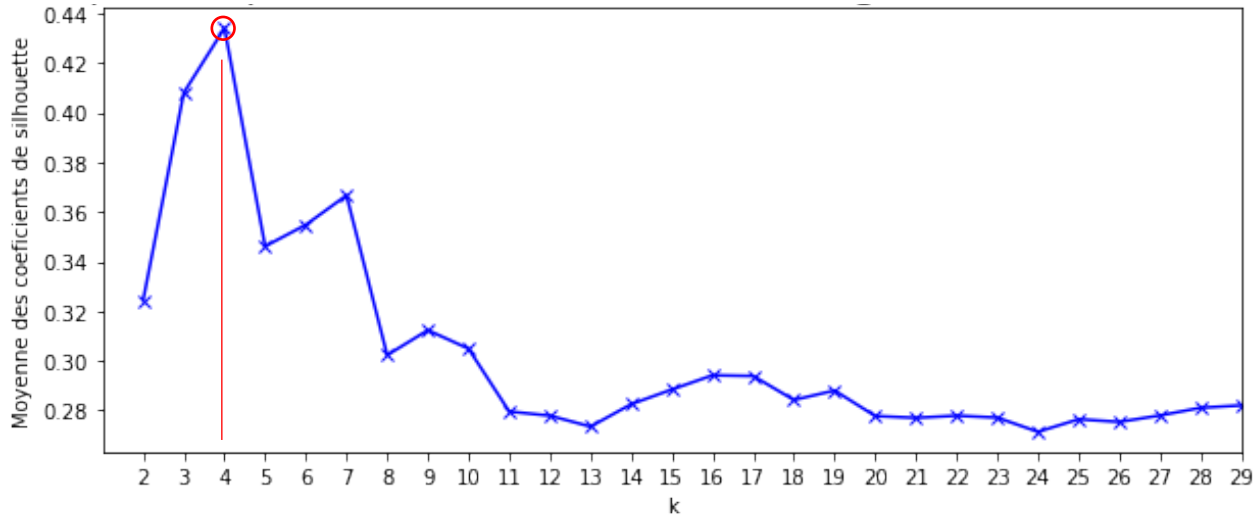


Figure 6.2. Méthode de la silhouette pour le partitionnement selon la charge de travail normalisée de la sous-section 3

À chaque fois que l'on utilise un algorithme de partitionnement, on détermine le nombre de clusters optimal en utilisant la méthode de la silhouette. On calcule pour k, représentant le nombre de clusters, allant de 2 à 15, le score moyen de la silhouette. Pour diminuer les temps de calcul, on calcule le score moyen de la silhouette sur seulement une fraction des données d'entrée de l'algorithme de partitionnement. Arbitrairement, on choisit de prendre seulement 30% des données disponibles. Le nombre optimal de clusters est obtenu avec le score moyen de la silhouette le plus proche de 1. Dans la figure 6.2, on détermine que le nombre optimal vaut 4. L'ensemble des graphes de la méthode de la silhouette sont disponibles en annexes.

Donc, on obtient six partitionnements des données basés sur WS_Load_data. Ces partitionnements sont résumés dans le tableau 6.1. On stocke dans une table de données appelée **Cluster_load** le numéro VIN de chaque voiture produite, ainsi que le cluster de chaque voiture pour chaque partitionnement de données basé sur la charge des postes de travail.

Tableau 6.1: Clusters basés sur les données de charges des postes de travail

Donnée	WS_Load_data					
Nom du partitionnement	Données normalisées	Sous-section 1	Sous-section 2	Sous-section 3	Sous-section 4	Somme de la charge normalisée selon les différentes sous-sections
Nombre de clusters	2	2	2	4	2	5

On utilise une approche similaire pour créer nos clusters à partir des données de la table AVO_data. On utilise directement les données brutes pour créer un partitionnement à l'échelle de la section étudiée. Puis, on utilise les données relatives aux opérations réalisées à chaque sous-section pour créer quatre partitionnements différents, chacun à l'échelle d'une sous-section. Ainsi, on obtient 5 partitionnements des données basés sur AVO_data qui sont résumés dans le tableau 6.2. On stocke dans une table de données appelée **Cluster_op** le numéro VIN de chaque voiture, ainsi que le cluster de chaque voiture pour chaque partitionnement de données basé sur les opérations réalisées sur la ligne d'assemblage.

Tableau 6.2: Clusters basés sur les données des opérations réalisées sur la voiture

Donnée	AVO_data				
Nom du partitionnement	Données brutes	Sous-section 1	Sous-section 2	Sous-section 3	Sous-section 4
Nombre de clusters	4	4	2	4	2

Finalement, on crée un dernier partitionnement des données basé à la fois sur les données de la table WS_Load_data et la table AVO_data. Ce partitionnement se base sur deux valeurs pour caractériser chaque voiture : la somme normalisée de l'ensemble des charges de travail effectuées sur la voiture et sur la somme normalisée du nombre d'opérations réalisées sur la voiture. On normalise les sommes en utilisant la méthode min/max. On obtient donc un dernier partitionnement des données avec 3 clusters différents. On stocke ce partitionnement de données dans **Cluster_op_load.**

6.4 Analyse statistique du taux de défauts

Après avoir créé nos 12 partitionnements de données, on réalise une analyse statistique du taux de défaut des voitures en fonction de leur cluster (étape 5.1) et de la séquence de production (étape 5.2). On considère qu'une voiture est défectueuse si elle rencontre au moins un défaut lors du processus d'assemblage. Avec cette définition d'un défaut, on a un taux de défaut de 9% sur notre jeu de données.

Dans un premier temps, on calcule pour chaque cluster de chaque partitionnement le taux de défaut des voitures du cluster. Puis, pour chaque couple de clusters A-B, on calcule le taux de défauts des voitures du cluster B sachant que la voiture produite précédemment était du cluster A. On reporte nos résultats dans un couple de cartes de densité qui nous donne pour chaque partitionnement le taux de défaut de chaque cluster et les taux de défaut des clusters en fonction des transitions entre eux. La figure 6.3 représente ce couple de cartes de densité des taux de défaut des clusters créés avec le partitionnement selon la charge de travail de la sous-section 3.

6.5 Identification des clusters « problématiques »

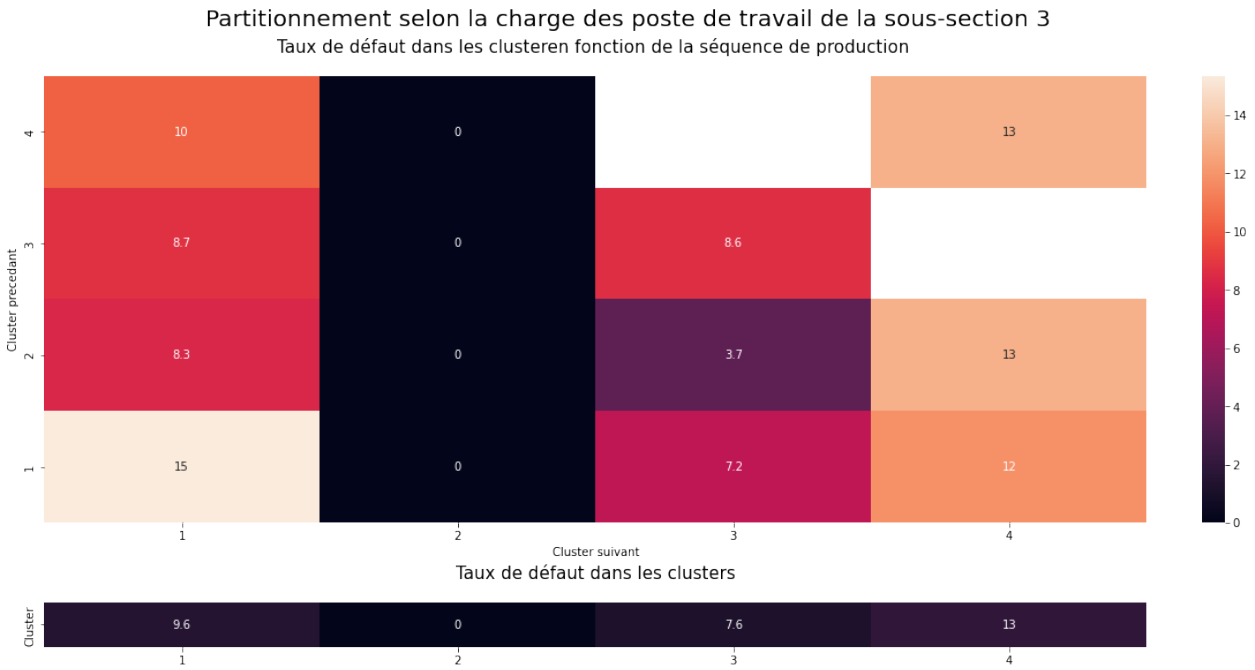


Figure 6.3: Couple de cartes de densité pour le partitionnement selon la charge de travail normalisée de la sous-section 3

Une fois que l'ensemble des couples de cartes de densité est créé, on peut utiliser ces graphiques pour identifier les clusters qui ont un taux de défaut important (étape 6). On remarque dans la figure 6.3 que le cluster 4 a un taux de défaut 13%, ce qui est quatre points au-dessus du taux de défaut du jeu de données. De plus, on remarque que la production successive de deux voitures du cluster 1 entraîne un taux de défauts de 15%, ce qui est 5% de plus que le taux de défaut du cluster 1. On peut en déduire que selon ce partitionnement, les clusters 1 et 4 sont problématiques. On remarque que les voitures du cluster 2 ont un taux de défaut de 0%. En effet, ce cluster regroupe exclusivement des voitures qui sont en démarrage de production.

De même, dans la figure 6.4, on note que les transitions entre les clusters 2 et 2 et les clusters 1 et 4 augmentent le taux de défaut de cinq et quatre pour cent respectivement par rapport au taux de défaut moyen des clusters 2 et 4. On peut donc dire que ces trois clusters sont problématiques. On remarque que les clusters 1 et 3 ont un taux de défaut de 0%. En effet, ces clusters regroupent exclusivement des voitures qui sont en démarrage de production.

L’ensemble des couples de cartes de densité sont disponibles en annexe.

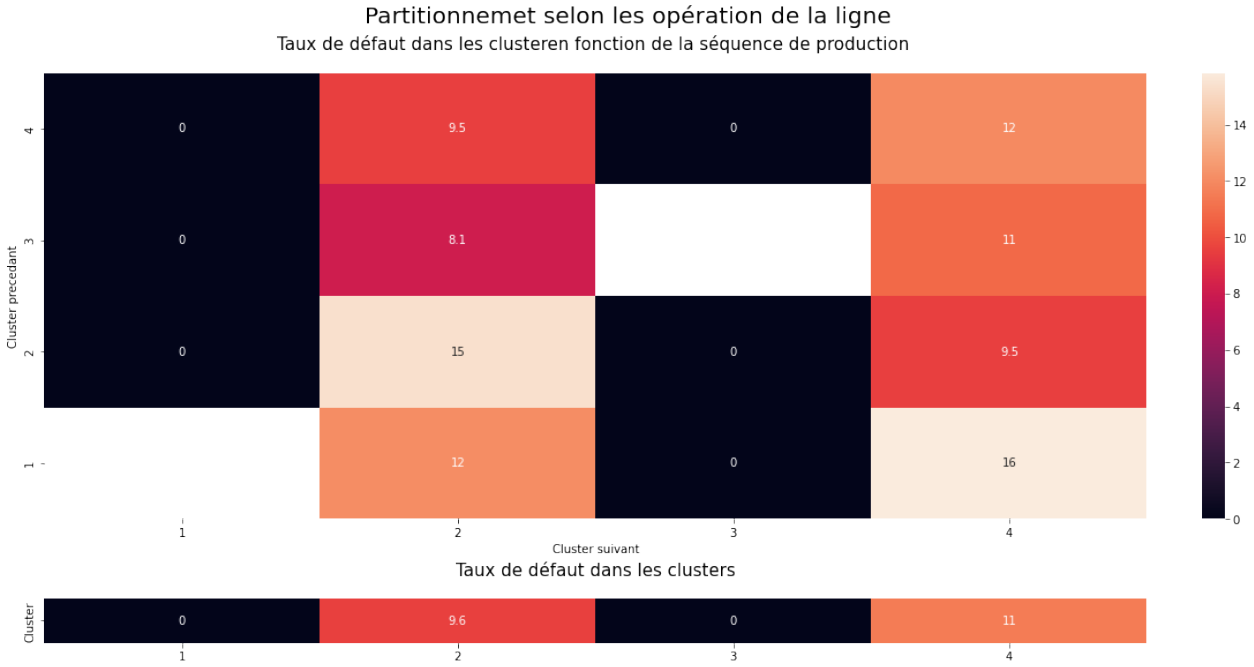


Figure 6.4: Couple de cartes de densité pour le partitionnement selon les opérations sur la ligne d'assemblage

6.6 Analyse des clusters

Une fois les clusters problématiques identifiés, nous cherchons à expliquer ces clusters en utilisant un algorithme d’arbre de décision pour séparer les clusters « problématiques » des autres en fonction de données caractérisant nos voitures (étape 7).

Dans notre cas, nous utilisons les données extraites de la table Cara qui caractérisent nos voitures. Pour chaque modèle de partitionnement créé lors de l’étape 4, on classifie nos clusters en fonction des caractéristiques de voitures en utilisant l’algorithme CART. On utilise les nœuds de notre arbre de décision pour expliquer un cluster et avoir des pistes de réflexion sur l’appartenance d’une voiture a un cluster dit problématique. Le but étant ici d’obtenir des pistes de réflexion sur l’origine des défauts sur la ligne d’assemblage. On utilise comme critère d’arrêt de séparation des branches une réduction minimum de l’impureté de 0,001.

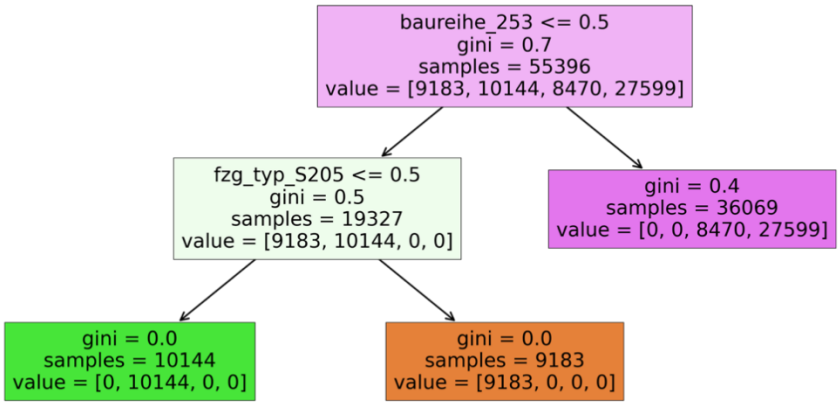


Figure 6.5: Arbre de décision classifiant les différents clusters en fonction des caractéristiques des voitures pour le partitionnement selon la charge de travail normalisée de la sous-section 3

En étudiant la figure 6.5, on en déduit que le cluster 1, représenté en orange dans la figure 6.5, regroupe des voitures qui n'ont pas la caractéristique « Baureihe_235 », mais ont la caractéristique « fzg_typ_S205 ». Le cluster 4, représenté en violet, regroupe les voitures qui ont « Baureihe_235 ». On peut en déduire que les voitures avec la caractéristique « Baureihe_235 » ont un taux de défaut plus important, et que cette caractéristique pourrait expliquer ces défauts. De plus, la répétition dans la séquence de production de voiture n'ayant pas la caractéristique « Baureihe_235 », mais ayant « fzg_typ_S205 », entraîne une augmentation du taux de défaut de quatre pour cent.

En étudiant la figure 6.6, on en déduit que le cluster 2, représenté en vert dans la figure 6.6, regroupe des voitures qui n'ont pas la caractéristique « Baureihe_235 », mais ont « fzg_typ_S205 ». Le cluster 4, représenté en violet, regroupe les voitures qui ont « Baureihe_235 ». Le cluster 1, représenté en orange, regroupe les voitures qui n'ont pas les caractéristiques « Baureihe_235 » et « fzg_typ_S205 », mais ont la caractéristique « fzg_typ_W206 ».

On peut en déduire la production successive de deux voitures avec la caractéristique « fzg_typ_S205 » et sans la caractéristique « Baureihe_235 » augmente la probabilité de défaut de 5%. De plus, les transitions entre des véhicules sans les caractéristiques « Baureihe_235 » et « fzg_typ_S205 » et avec la caractéristique « fzg_typ_W206 », ainsi que les véhicules avec la

caractéristique « Baureihe_235 » augmentent le taux de défaut sur les seconds types de véhicules de 5 %.

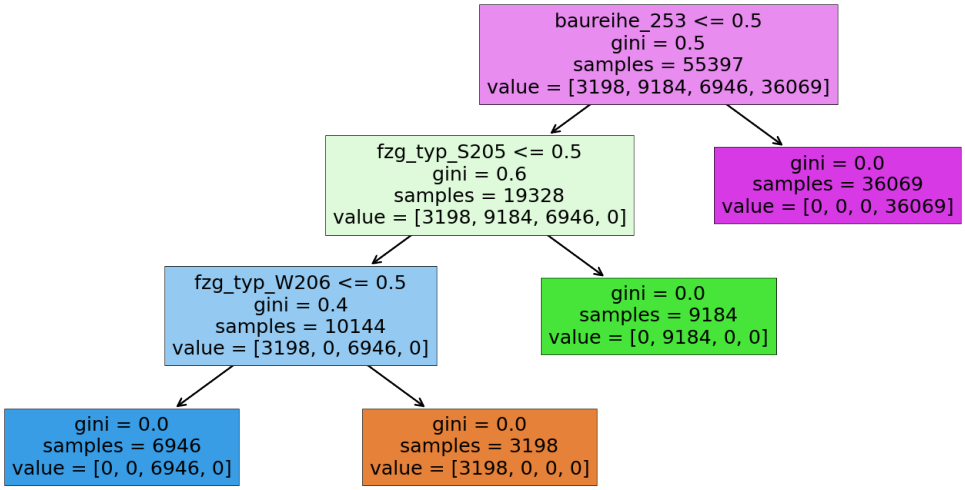


Figure 6.6: Arbre de décision classifiant les différents clusters en fonction des caractéristiques des voitures pour le partitionnement selon les opérations réalisées sur la ligne

Les conclusions de l'étape 7 nous permettent de facilement identifier les types de produits qui posent problème et de proposer des actions correctives pour éliminer la source des défauts (étape 8.1). Les planificateurs de production peuvent ainsi éviter les transitions entre clusters « problématiques » afin de diminuer le taux de défaut global de la ligne d'assemblage. De plus, ces conclusions peuvent être partagées avec le service qualité de l'entreprise pour fournir de nouvelle piste de réflexion pour les analyses causales des experts.

6.7 Conclusion

Ce chapitre visait à évaluer la validité de notre méthode au travers d'une étude de cas. Nous avons ainsi utilisé comme données les temps théoriques de production, les opérations réalisées lors de l'assemblage des voitures, la séquence de production et les caractéristiques des voitures produites. Après la préparation des données (phase 1), nous avons créé des clusters grâce à un algorithme HCA puis identifié des clusters « problématiques » à l'aide de deux analyses statistiques. Durant cette deuxième phase, nous avons identifié deux transitions entre clusters qui mènent à une augmentation du taux de défauts de 5 %. Puis, durant la troisième phase, nous avons expliqué ces clusters en utilisant un algorithme d'arbre de décision afin d'identifier leurs caractéristiques. Cette

approche a ainsi permis aux opérateurs de facilement proposer des actions correctives afin de diminuer le taux de défaut de la ligne d'assemblage.

Bien que ces résultats soient satisfaisants, rien ne garantit sa performance dans un autre contexte manufacturier. En effet, la méthode a été développée en prenant en compte les caractéristiques propres à l'environnement de production de notre partenaire industriel. Le prochain chapitre cherche justement à évaluer le caractère généralisable de l'approche proposée en la testant dans un autre environnement de production multiproduit.

CHAPITRE 7 TEST DE GÉNÉRALISATION DU MODÈLE

Dans ce chapitre, on propose d'applique notre méthode à un autre cas d'application pour vérifier si notre modèle est généralisable à d'autres contextes de production.

Notre second partenaire de recherche est un constructeur de pneumatique japonais. Nous avons eu accès aux données d'une usine située en Amérique du Nord et plus particulièrement aux données relatives à une machine responsable de toutes les opérations d'assemblage de leur pneumatique. Ce cas d'étude ressemble au cas d'étude précédent. Dans le premier cas, on étudie l'ensemble des opérations réalisées sur une ligne d'assemblage automobile, alors que dans le second, on s'intéresse à l'ensemble des opérations d'assemblage des pneumatiques réalisées sur une seule machine. Cependant, on peut noter deux différences importantes. Le second partenaire a un nombre de références bien plus faible que le premier et celui-ci travaille avec une stratégie de lot de fabrication. De plus, on remarque que les opérations réalisées sur les pneus du second partenaire sont totalement automatisées, tandis que les opérations réalisées sur les lignes d'assemblage automobiles du premier partenaire sont majoritairement manuelles.

L'application de notre méthode à ce cas d'étude nous permet donc de vérifier sa compatibilité à la stratégie de production en lot de fabrication et aux lignes d'assemblage entièrement automatisées.

7.1 Compréhension de l'entreprise et des données

Premièrement, nous analysons le contexte de notre partenaire industriel pour identifier le cadre de notre étude (étape 1). Celui-ci est centré sur une seule machine de production. Nous considérerons seulement les pneus produits entre le 18 octobre 2020 et le 18 novembre 2021. Lors de cette période, aucun événement notable n'a lieu. Nous n'aurons donc pas d'opération de préparation de données particulière à réaliser.

Puis, nous étudions les données disponibles pour notre analyse (étape 2). Celles-ci sont extraites de l'ERP de l'entreprise. La première table de données est extraite de l'ordonnancement de la production. On appelle cette table de données **Prod_data.** Elle nous permet d'avoir la séquence de production de notre machine d'assemblage. Elle est constituée de trois colonnes : l'identifiant du pneu produit, l'horodatage de la fin de l'assemblage du pneu et la référence du type de pneu

assemblé. Durant notre période d'analyse, la machine a produit en tout 63 références différentes de pneu, pour un total de 285 969 pneus.

La seconde table de données est appelée **Délai_data.** Elle contient l'ensemble des informations concernant les événements entraînants des retards rencontrés durant l'assemblage des pneus. Cette table est constituée de différentes colonnes : l'horodatage du délai rencontré lors de la production, l'identifiant et le nom du délai et de sa catégorie. On recense 123 types de délais différents et 438 017 délais durant notre période d'analyse.

Finalement, la dernière table de données est appelée **Product_parametre**. Elle liste l'ensemble des changements des paramètres de production. Cette table nous indique pour chaque changement de paramètre sur notre machine, l'horodatage de celui-ci, le paramètre de production modifié, son ancienne et sa nouvelle valeur. Durant la période d'analyse, on reporte 18 315 changements sur 527 paramètres de production.

On a donc trois tables de données que l'on va préparer pour l'analyse durant l'étape 3.

7.2 Préparation des données

Dans un premier temps, nous avons trié les événements entrainants des retards rencontrés durant l'assemblage de la table **Délai_data** pour conserver seulement ceux qui ont un rapport avec la qualité de production du pneu. Ainsi, on considérera qu'un pneu est défectueux si au moins un événement a lieu lors de sa production. Après avoir trié la table de données, on a plus que 67 177 délais répartis sur 48 catégories.

Puis, on ajoute à la table **Prod_data** les paramètres de production de chaque pneu en recoupant cette table avec la table de données **Product_parametre**.

Une représentation des données utilisées pour l'étude de cas après prétraitement est donnée à la figure 7.1.

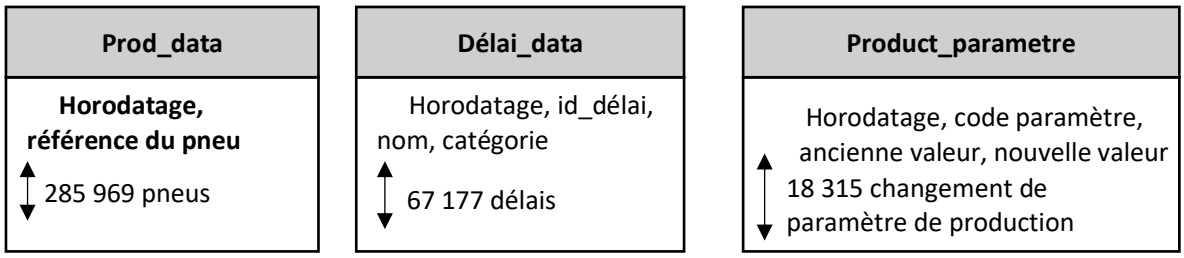


Figure 7.1: Représentation après prétraitement des tables de données utilisées dans la seconde étude de cas

7.3 Sélection et regroupement des produits à analyser

L'objectif de cette étape est de créer des regroupements de produits qui ont un sens du point de vue du processus d'assemblage des pneus (étape 4).

Contrairement à l'étude précédente, nous avons un nombre fini de produits que l'on fabrique par lot. Nous faisons l'hypothèse que réaliser une étude sur l'ensemble des références produites revient à faire plusieurs études distinctes, une pour chaque référence. Cette hypothèse se base principalement sur le fait que notre partenaire industriel utilise une stratégie de production par lot, et donc, l'impact que peut avoir une référence sur une autre est faible, car à chaque changement de référence/lot de fabrication la machine est arrêtée pour préparer la production du lot suivant.

De même, on ne peut pas utiliser les différences de temps de production ou les opérations réalisées sur les pneus pour les distinguer, car étant de la même référence, ces données sont identiques pour chaque pneu. On utilise donc les valeurs des paramètres de production utilisées lors de la fabrication des pneus.

Dans ce cas d'étude, il n'est pas nécessaire d'utiliser l'algorithme HCA pour regrouper nos produits dans des clusters. On peut simplement regrouper nos pneus en fonction des combinaisons de paramètres de production utilisés lors de l'assemblage des pneus étudiés, car le nombre de combinaison de paramètres de production est relativement faible pour chaque référence.

Si l'on prend l'exemple de la référence J00315, on a 167 pneus de cette référence qui sont produits dans nos données. On identifie 17 paramètres de production qui change au cours l'assemblage de cette référence. On a seulement 12 combinaisons différentes de ces 17 paramètres dans notre jeu de données. On peut donc créer des clusters en regroupant nos pneus en fonction des combinaisons des paramètres de production présents dans nos jeux de données.

7.4 Analyse statistique du taux de défauts

On calcule pour chaque combinaison de paramètres de production/cluster son taux de défaut (étape 5.1). On fait l'hypothèse qu'il n'est pas pertinent d'analyser le taux de défaut des clusters de pneus en fonction de la séquence de production (étape 5.2), car le nombre de transitions entre des clusters différents est très faible dans notre jeu de données. On explique cela par la stratégie des changements des paramètres de production. En effet, on change les paramètres de production seulement en réponse à l'assemblage d'un pneu défectueux. On n'a donc pas de va-et-vient entre les différentes combinaisons de paramètres de production/ clusters dans notre jeu de données.

7.5 Identification des clusters « problématiques »

On utilise donc seulement l'analyse de l'étape 5.1 pour identifier nos clusters « problématiques » (étape 6). On considère qu'un cluster est « problématique » s'il a un taux de défaut important. Pour considérer seulement les clusters pertinents pour l'analyse suivante, on impose un nombre minimum d'apparitions de notre cluster dans le jeu de données.

Tableau 7.1: Sélection des combinaisons "problématiques" pour la référence J000315.

490	181	451	460	494	16	495	492	493	491	452	457	488	487	454	453	489	Nombre	def	Selection
0.2	60.0	0.08	0.35	0.01	379.0	0.0	0.01	0.01	0.23	0.08	0.35	0.06	0.06	0.35	0.08	0.06	1	0.000000	False
0.2	60.0	0.08	0.4	0.01	379.0	0.0	0.01	0.01	0.23	0.08	0.304	0.06	0.06	0.35	0.08	0.06	2	0.000000	False
0.2	60.0	0.08	0.4	0.01	381.0	0.0	0.01	0.01	0.23	0.08	0.304	0.06	0.06	0.35	0.08	0.06	2	100.000000	False
0.2	60.0	0.08	0.4	0.01	381.0	0.0	0.01	0.01	0.23	0.08	0.304	0.12	0.12	0.35	0.08	0.12	8	37.500000	False
0.2	60.0	0.08	0.4	0.01	381.0	0.0	0.01	0.01	0.23	0.08	0.304	0.12	0.12	0.3	0.08	0.12	5	0.000000	False
0.2	60.0	0.07	0.4	0.01	381.0	0.0	0.01	0.01	0.23	0.07	0.304	0.12	0.12	0.3	0.07	0.12	20	30.000000	True
0.25	60.0	0.07	0.4	0.01	381.0	0.0	0.01	0.01	0.23	0.07	0.304	0.12	0.12	0.3	0.07	0.12	3	66.666667	False
0.25	62.0	0.07	0.4	0.01	381.0	0.0	0.01	0.01	0.23	0.07	0.304	0.12	0.12	0.3	0.07	0.12	1	0.000000	False
0.25	58.0	0.07	0.4	0.01	381.0	0.0	0.01	0.01	0.23	0.07	0.304	0.12	0.12	0.3	0.07	0.12	20	40.000000	True
0.28	58.0	0.07	0.4	0.01	381.0	0.0	0.01	0.01	0.25	0.07	0.304	0.12	0.12	0.3	0.07	0.12	3	0.000000	False
0.28	58.0	0.07	0.4	0.0	381.0	0.0	0.0	0.0	0.25	0.07	0.304	0.12	0.12	0.3	0.07	0.12	74	58.108108	True
0.44	58.0	0.07	0.4	0.0	381.0	0.0	0.0	0.0	0.25	0.07	0.304	0.12	0.12	0.3	0.07	0.12	28	50.000000	True

Par exemple, pour le cas de la référence J000315, on a identifié 12 combinaisons de paramètres de production possibles dans notre jeu de données. En considérant qu'un cluster est « problématique » si son taux de défaut est supérieur ou égal à 30%, et que pour être pertinent, il doit au moins apparaître 20 fois dans notre jeu de données. On identifie ainsi 4 clusters « problématiques », comme le montre le Tableau 6.3.

7.6 Analyse des clusters.

L'objectif de la dernière analyse est d'expliquer pourquoi un pneu appartient à un cluster problématique (étape 7). Normalement, selon la méthode introduite au chapitre précédent, on devrait créer un arbre de décision qui classifie les clusters en fonction des caractéristiques produits des pneus. Or, vu que l'on analyse chaque référence de pneu indépendamment, il n'y a pas de différences de caractéristique produit entre les pneus dans notre jeu de données. On utilise donc encore les paramètres de production pour séparer les clusters « problématiques » des autres. On utilise l'algorithme CART pour classifier les différentes combinaisons de valeur des paramètres de production en fonction du caractère « problématique » ou non de la combinaison.

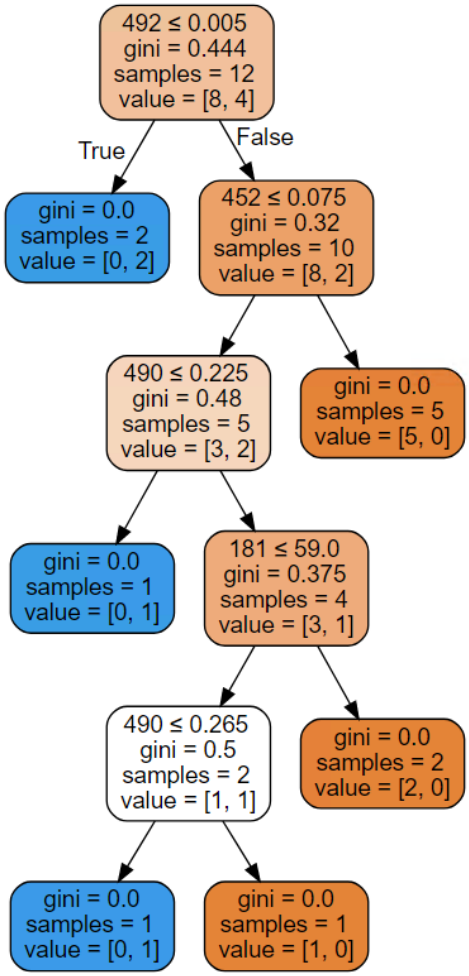


Figure 7.2: Arbre de décision séparant les combinaisons des variables de production du pneu J000315

On obtient donc un arbre de décision qui sépare les combinaisons « problématiques » des autres. On peut utiliser les tests effectués sur les paramètres de production sur les branches de notre arbre de décision pour expliquer pourquoi une combinaison de paramètres est dite problématique, et donc, expliqué le taux de défaut important des pneus de ce cluster. Par exemple, pour les pneus J000315, on obtient l'arbre de décision présenté à la figure 6.6. On peut donc dire que les clusters problématiques sont associés à des combinaisons de production ayant soi :

- le paramètre $492 \leq 0{,}005$;
- le paramètre $492 > 0{,}005$, le paramètre $452 \leq 0{,}075$ et le paramètre $490 \leq 0{,}225$; et
- le paramètre $492 > 0{,}005$, le paramètre $452 \leq 0{,}075$ et le paramètre $0{,}265 \geq 490 > 0{,}225$, et le paramètre $181 \leq 59$.

De plus, on peut déterminer l'importance d'un paramètre pour distinguer les combinaisons « problématiques » des autres en comptant le nombre de fois que chaque paramètre est utilisé dans notre arbre. On remarque donc que le paramètre 490 apparaît deux fois dans l'arbre. Ce paramètre est donc plus important que les autres pour identifier les combinaisons « problématiques » et explique leur haut taux de défauts. On en déduit que ce paramètre de production est un bon candidat pour expliquer les défauts du produit J000315.

On peut répéter cette analyse sur différent produit et déterminer les paramètres les plus importants pour expliquer le haut taux de défaut des clusters « problématique » en comptant sur tous les arbres de décision le nombre d'apparitions de chaque paramètre.

On réalise la même étude pour les sept références qui ont les taux de défauts les plus importants dans notre jeu de données. On obtient le tableau 7.2 qui nous donne un indice de l'importance des paramètres de production pour identifier les clusters « problématiques ». Les 9 premiers paramètres sont tous relatifs à une opération faite sur le flanc des pneus. On en déduit que c'est cette opération qui est la cause racine des défauts lors de l'assemblage des pneus.

Les opérateurs peuvent donc transmettre cette conclusion au service qualité de l'entreprise pour fournir cette nouvelle piste de réflexion pour les analyses causales des experts (étape 8.1). Le partenaire industriel a confirmé cette hypothèse en disant que le problème était déjà connu des services de qualité et était en cours d'investigation.

Tableau 7.2: Importance des paramètres pour les 7 références de pneus qui ont le plus haut taux de défauts

Paramètres	Occurrence
489	16
473	15
491	13
490	12
476	12
487	11
475	10
488	10
474	9
464	8

7.7 Conclusion

Ce chapitre a permis de démontrer le caractère généralisable de la méthode proposée. Nous avons appliqué notre méthode à un autre contexte de production : une ligne d'assemblage de pneu entièrement automatisé qui utilise une stratégie de fabrication par lot. Ce contexte particulier nous a permis de faire deux hypothèses, étudier l'ensemble des produits revient à faire une analyse par produit, la stratégie par lot de fabrication et la stratégie des changements des paramètres de production nous permettent d'ignorer les transitions entre les pneus en fonction de la séquence de production. Contrairement à l'analyse précédente, on ne peut pas considérer les mêmes données pour notre étude, car celle-ci est identique pour chaque produit. On utilise donc les paramètres de

productions. Afin de rassembler les conclusions des études par produit, on détermine l'importance des paramètres de production pour distinguer les combinaisons « problématiques » des autres. Pour ce faire, on compte le nombre de fois que chaque paramètre est utilisé dans nos arbres pour séparer les combinaisons problématiques. On remarque que les 9 premiers paramètres sont tous relatifs à une opération faite sur le flanc des pneus. On en déduit que c'est cette opération qui est la cause racine des défauts lors de l'assemblage des pneus. Cette conclusion a été confirmée par notre partenaire industriel.

CHAPITRE 8 CONCLUSION ET RECOMMANDATIONS

L'objectif principal de ce projet était de **faciliter l'identification des causes racines des défauts sur les lignes d'assemblage multiproduits par la valorisation de données**. Au cours de cette recherche, nous avons ainsi développé un modèle d'analyse causale en utilisant la méthodologie de recherche DRM. Celle-ci nous permettant de répondre à nos trois sous-objectifs de recherche.

Dans un premier temps, nous avons réalisé une revue de littérature systématique sur l'application des méthodes de valorisation de données sur les lignes d'assemblage. Celle-ci nous a permis d'identifier les méthodes de préparation des données et d'analyse des données applicables à ce contexte. Cependant, l'application directe de ces méthodes à notre cas d'étude ne nous a fourni aucun résultat pertinent. Contrairement à notre cas d'étude, les méthodes proposées dans la revue de littérature sont centrées sur l'analyse d'un produit unique, alors que dans notre cas, nous devons composer avec une ligne d'assemblage pouvait produire un nombre infini de produits. Toutefois, nous avons extrait de ces premiers tests quatre recommandations qui nous ont guidés dans le développement de notre propre méthode d'analyse causale.

Ainsi, nous avons développé, dans un deuxième temps, une méthode d'analyse causale des défauts applicable aux lignes d'assemblage multiproduits basées sur les données de temps de production, les opérations réalisées à chaque poste de travail et sur les données caractérisant les produits assemblés. Nous avons testé notre méthode sur une ligne d'assemblage automobile.

La première phase de cette méthode consiste à préparer les données extraites de la ligne d'assemblage multiproduit. Nous avons normalisé les temps de production des produits à chaque poste de travail pour facilement comparer et manier ces données. Nous avons aussi calculé les temps de production totaux pour certaines sections de notre ligne d'assemblage. Nous avons séparé les différentes opérations et les temps de production pour extraire seulement les données d'intérêt.

La seconde phase consiste à identifier des groupes de produits « problématiques ». Pour ce faire, nous avons utilisé un HCA pour regrouper nos produits dans des clusters. Puis, nous avons réalisé une étude statistique des taux de défauts des produits en fonction de leurs clusters et de la séquence de production pour identifier les clusters qui mènent aux défauts. En utilisant cette méthode d'analyse causale, nous avons découvert trois transitions entre des clusters qui mènent à une augmentation du taux de défauts sur la ligne d'assemblage étudié.

La dernière phase vise à expliquer ces clusters. Nous avons utilisé l'algorithme CART pour créer un arbre de décision qui sépare nos clusters à haut taux de défauts des autres. Grâce à l'analyse de cet arbre, nous avons identifié les caractéristiques majeures de chaque clusters « problématiques ». Notre méthode d'analyse nous a permis de conclure que, deux transitions entre deux types de véhicules particuliers augmentant le taux de défaut de 5%.

Finalement, nous avons mesuré le caractère généralisable de notre méthode en l'appliquant à un autre contexte de production : une ligne d'assemblage de pneu entièrement automatisée qui utilise une stratégie de fabrication par lot. Ce contexte particulier nous permet de faire des hypothèses qui facilitent notre analyse. Nous sommes arrivés à la conclusion qu'une opération faite sur les flans des pneus était la cause racine des défauts. Cette conclusion a été confirmée par notre partenaire industriel.

Notre méthode nous a donc permis d'identifier des causes racines de certains défauts sur des processus d'assemblage de nos partenaires industriels. Celle-ci est bien un ajout à la littérature scientifique, car aucun article de recherche ne traite de l'analyse des causes des défauts sur les lignes d'assemblage multiproduits. La force de notre modèle repose sur le fait que la séparation de nos produits est définie à l'aide de données de production et non des données caractérisant nos produits. Nous avons donc des groupes de produits ont un sens vis-à-vis du processus de production. De plus, nous prenons en compte la séquence de production de la ligne d'assemblage, ce qui nous permet d'identifier des causes racines en lien avec la transition entre deux types de produits.

Cependant, nous sommes bien conscients que le modèle proposé a des limites. Premièrement, nous avons validé notre méthode seulement au travers de deux cas d'étude. L'évaluation de la performance et de la robustesse du modèle nécessiterait la conduite d'un nombre plus important d'expérimentations, et ce, sur des lignes d'assemblage multiproduits variées. De plus, la portée de nos analyses a été limitée en raison de données manquantes. En effet, notre premier partenaire industriel n'était pas capable de préciser la localisation des défauts sur la section de ligne étudiée. Nous n'avons donc pas pu réduire l'échelle de notre analyse au niveau d'un poste de travail précis, ce qui nous aurait permis d'identifier des causes racines spécifiques à une opération et non à la ligne d'assemblage dans son ensemble.

Aussi, nous reconnaissons que le modèle n'est pas encore capable de formellement identifier les causes racines des défauts. Il fournit seulement des pistes de réflexion aux experts. Cependant, l'identification des clusters « problématiques » permet aux opérateurs d'identifier les types de produits à risque, et donc, de prendre des actions correctives afin de limiter les défauts sur la ligne d'assemblage.

D'un point de vue pratique, notre modèle requiert un nombre important de données et des compétences spécifiques en statistique et en analyse des données pour être appliqué. Or, ces compétences sont actuellement rares dans l'industrie, même si elles tendent à se démocratiser.

En conclusion, cette étude représente le premier pas vers le développement de nouvelle méthode d'analyse des causes racines des défauts sur les lignes d'assemblage multiproduit. Elle peut éventuellement servir de guide à de nouveaux chercheurs essayant de développer de nouvelles méthodes d'analyse causales.

RÉFÉRENCES

Ahmed, S. R., " Applications of data mining in retail business", International Conference on Information Technology: Coding and Computing, 2004. Proceedings. ITCC 2004. IEEE., Las Vegas, NV, USA, 5-7 April 2004, vol. 2, pp. 455-459.

Andersen, B., & Fagerhaug, T., *Root cause analysis: simplified tools and techniques*. Quality Press, 2006.

ASQ. (2021). What is root cause analysis (RCA)? [En ligne]. Disponible : https://asq.org/quality-resources/root- -analysis, consulté le 22 avril 2021.

Aurisicchio, M., Bracewell, R., & Hooey, B. L., *Rationale mapping and functional modelling enhanced root cause analysis*. Safety science, vol. 85, pp. 241-257, 2016.

Baranwal, A., Meyer, M., Nguyen, T., Pillai, S., Nakayamada, N., et al., "Five deep learning recipes for the mask-making industry", Proceedings of SPIE - The International Society for Optical Engineering, Monterey, California, USA, 25 October 2019, vol. 11148, p. 31-49.

Linoff, G. and Berry, M. J. A., "Data mining techniques: for marketing, sales, and customer relationship management". *3rd ed*., 2011.

A. Berson, S. Smith, K. Thearling,"Building Data Mining Applications for CRM" McGraw-Hill, 2000.

Blessing, L. T. M., et Chakrabarti, A., "DRM, a design research methodology", Springer, 2009.

Carvalho, T; P., Soares, F. A. A. M. N., Vita, R., P. Francisco, R., Basto, J., P., Alcala, . G. S., "A systematic literature review of machine learning methods applied to predictive maintenance", *Computers & Industrial Engineering*, vol. 137, p. 106024, 2019.

Chemweno, P., Morag, I., Sheikhalishahi, M., Pintelon, L., Muchiri, P., & Wakiru, J. ,"Development of a novel methodology for root cause analysis and selection of maintenance strategy for a thermal power plant: A data exploration approach", Engineering Failure Analysis, vol. 66, pp. 19-34, 2016.

Cinar, G T., Thompson, J., Srinivasan, S., "Cost-Sensitive Optimization of Automated Inspection",. *Proceedings - 2015 IEEE International Conference on Big Data, IEEE Big Data 2015*, Santa Clara, CA, USA, 29 Oct.-1 Nov. 2015, pp. 1211-1219

Dogget, A.M., “Root Cause Analysis: A Framework for Tool Selection”, *The Quality Management Journal*. 4e éd., vol. 12, p. 34, 2005.

Duan, P., He, Z., He, Y., Lui, F., Zhang, A., Zhou, D., “Root cause analysis approach based on reverse cascading decomposition in QFD and fuzzy weight ARM for quality accidents”, *Computers & Industrial Engineering*, vol. 147, p. 106643, 2020.

Forfas (Ireland) Expert Group on Future Skills Needs (Ireland)(EGFSN), “Assessing the demand for big data and analytics skills, 2013-2020”, 2014.

Gantz, J., & Reinsel, D., “Extracting value from chaos”. *IDC iview*, vol. 1142, no 2011, p. 1-12., 2011.

Gardner, R., Bieker, J., Elwell S. “Solving Tough Semiconductor Manufacturing Problems Using Data Mining”, *2000 IEEE/SEMI Advanced Semiconductor Manufacturing Conference*, Boston, MA, USA, 2-14 Sept. 2000, pp. 46-55.

Hammami, Z., Mouelhi, W., Ben Said, L., “On-line self- adaptive framework for tailoring a neural-agent learning model addressing dynamic real-time scheduling problems”, *Journal of Manufacturing Systems*. vol. 45, p. 97-108, 2017.

Han, B., Zhang, C., & Xie, C., “A root-cause analysis method for fault diagnosis in condenser”, *IOP Conference Series: Earth and Environmental Science*. 2020, vol. 446, No. 5, p. 052-055.

Huang, K., Stratigopoulos, H. G., & Mir, S., “Fault diagnosis of analog circuits based on machine learning”, *2010 Design, Automation & Test in Europe Conference & Exhibition.* IEEE, Dresden, Germany, 8-12 March 2010, p. 1761-1766.

Hirsch, V., Reimann, P., & Mitschang, B., “Data-driven fault diagnosis in end-of-line testing of complex products”, *2019 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*. IEEE, Washington, DC, USA, 5-8 Oct. 2019, p. 492-503.

Jordan, M. I., Mitchell, T. M., “Machine learning: Trends, perspectives, and prospects”. *Science*, vol. 349, no 6245, p. 255-260, 2015.

Kane, P., Andhare, A., “Application of Psychoacoustics for Gear Fault Diagnosis Using Artificial Neural Network”. *Journal of Low Frequency Noise, Vibration and Active Control*, vol. 35, no. 3, p. 207–20, 2016.

Kulin, M., Kazaz, T., De Poorter, E., et al., « A survey on machine learning-based performance improvement of wireless networks: PHY, MAC and network layer”, Electronics, vol. 10, no 3, p. 318, 2021.

Lad, P., Somani, A., Krishnan, K. E., Gupta, A., & Kartik, V., "High-throughput shape classification using support vector machine". *2016 IEEE International Conference on Industrial Technology (ICIT)*. IEEE, Taipei, Taiwan, 14-17 March 2016, p. 854-859.

Lejeune, M. A. "Measuring the impact of data mining on churn management", *Internet Research: Electronic Networking Applications and Policy*, vol.11, p. 375-387 2001.

Lantz B., "Machine Learning with R". *Packt Publishing,* 2013.

Laxman S., Shadid B., Sastry P.S., and Unnikrishnana K.P., "Temporal data mining for root-cause analysis of machine faults in automotive assembly lines", 2009.

Li, L., Ota, K., Dong, M., "Deep Learning for Smart Industry: Efficient Manufacture Inspection System with Fog Computing", *IEEE Transactions on Industrial Informatics,* vol. 14, no. 10, p. 4665–73, 2018.

Mangal, A., Kumar, N., "Using Big Data to Enhance the Bosch Production Line Performance: A Kaggle Challenge", *2016 IEEE International Conference on Big Data*. Washington, DC, USA, 5-8 Dec. 2016 pp. 2029-2035.

Manns, M., Wallis, R., & Deuse, J., "Automatic proposal of assembly work plans with a controlled natural language", *9th CIRP conference on intelligent computation in manufacturing engineering,* Capri, Italy, Dec. 2015, pp. 345–350.

Manyika, J., Chui, M., Brown, B., Bughin, J., Dobbs, R., Roxburgh, C., & Hung Byers, A., "Big data: The next frontier for innovation, competition, and productivity", *McKinsey Global Institute,* 2011.

Maurya, A., "Bayesian Optimization for Predicting Rare Internal Failures in Manufacturing Processes". *2016 IEEE International Conference on Big Data, Big Data 2016*. Washington, DC, USA, 5-8 Dec. 2016, p. 2036-2045).

Moeuf, A., Pellerin, R., Lamouri, S., Tamayo-Giraldo, S., & Barbaray, R., "The industrial management of SMEs in era of Industry 4.0"*, International Journal of Production Research*. vol. 56, no 3, p. 1118-1136, 2018.

Mohri, M., Rostamizadeh, A., & Talwalkar, A., "Foundations of machine learning", *MIT press,* 2018.

Mueller, T., Greipel, J., Weber, T., and Schmitt, R.H., "Automated root cause analysis of non-conformities with machine learning algorithms", *Journal of Machine Engineering*, vol. 18, No. 4, p. 60–72, 2018.

Nedelkoski, S., Stojanovski, G., "Machine Learning for Large Scale Manufacturing Data with Limited Information", *13th IEEE International Conference on Control and Automation*, Ohrid, Macedonia, 3-6 July 2017, p. 70-75.

Abdelrahman, O., & Keikhosrokiani, P., "Assembly line anomaly detection and root cause analysis using machine learning", *IEEE Access*, vol. 8, p. 189661-189672, 2020.

Patnaik, S. K., Sahoo, S., & Swain, D. K., "Clustering of Categorical Data by Assigning Rank through Statistical Approach", *International Journal of Computer Applications*, vol. 43, no 2, p. 1-3, 2012.

Pavlyshenko, B., "Machine Learning, Linear and Bayesian Models for Logistic Regression in Failure Detection Problems", *2016 IEEE International Conference on Big Data* (Big Data), Washington, DC, USA, 5-8 Dec. 2016, pp. 2046-2050.

Rahmatov, N., Paul, A., Saeed, F., Hong, W., Seo, H., Kim, J., "Machine learning–based automated image processing for quality management in industrial Internet of Things", *International Journal of Distributed Sensor Networks*, vol. 15, 2019.

Ravikumar ,S., Ramachandran, K.I., Sugumaran, V., "Machine learning approach for automated visual inspection of machine components", *Expert Systems with Applications*, vol. 38, no 4, p. 3260-3266, 2011.

Rodriguez, A., Bourne, D., Mason, M., Rossano, G. F., & Wang, J., "Failure detection in assembly: Force signature analysis", *2010 IEEE International Conference on Automation Science and Engineering,* Toronto, ON, Canada, 21-24 Aug. 2010, p. 210-215.

Sand, C., Kunz, S., Hubbert, H., & Franke, J., "Towards an inline quick reaction system for actuator manufacturing using data mining", *2016 6th International Electric Drives Production Conference (EDPC),* Nuremberg, Allemagne, 30 Nov. – 1 Dec., 2016, p. 74-79.

Sarkar P., "Clustering of Event Sequences for Failure Root Cause Analysis", *Quality engineering*, vol. 16, No. 3, pp. 451–460, 2004.

Sassi, P.,Tripicchio, P., Alberto Avizzano, C., "A Smart Monitoring System for Automatic Welding Defect Detection", *IEEE Transactions on Industrial Electronics,* vol. 66, no. 12, pp. 9641–50, 2019.

Shih-Yang, L., Yun, D., Po-Chang, K., Tzu-Jung, W., Ping-Tsan, H., Sivakumar, V., Rama subbareddy, "Fog Computing Based Hybrid Deep Learning Framework in Effective

Inspection System for Smart Manufacturing", *Computer Communications*, vol. 160, p. 636–642, 2020.

Silva Peres, R., Barata, J., Leitao, P., Garcia, G., "Multistage Quality Control Using Machine Learning in the Automotive Industry", *IEEE Access*, vol. 7, p. 79908–79916, 2019.

Soares, S. G., "Ensemble learning methodologies for soft sensor development in industrial processes". *Ph.d. thesis Computer engineering, faculty of sciences and technology*. Coimbra, Portugal, 2015.

Soares, S. G., Araujo, R., "AN on-line weighted ensemble of regressor models to handle concept drifts", *Engineering Applications of Artificial Intelligence*, vol. 37, p. 392-406, 2015.

Tao, F., Qi, Q., Liu, A., & Kusiak, A., "Data-driven smart manufacturing", *Journal of Manufacturing Systems*, vol. 48, p. 157–169, 2018.

Efraim, T., Jay, E. A., Tin-peng, L. & Ramesh, S., "Decision support and business intelligence systems", *Pearson Education*, 2007.

Töpfer, A., "Six Sigma: Projektmanagement für Null-Fehler-Qualität in der Automobilindustrie", *ZfAW*, vol. 2, p. 13-24, 2004.

Wang, J., Hong, H., Long, C., Caiying, H., "Assembly defect detection of atomizers based on machine vision"*, Proceedings of the 2019 4th International Conference on Automation, Control and Robotics Engineering,* Chine, Shenzhen, Jui. 2019, p.1-6.

Wang, Y., Sun, Y., Lv, P., Wang, H., "Detection of line weld defects based on multiple thresholds and support vector machine", *NDT&E International*, vol. 41, no 7, p. 517-524, 2008.

Xu, C., & Zhu, G., "Intelligent manufacturing lie group machine learning: Real-time and efficient inspection system based on fog computing", *Journal of Intelligent Manufacturing*, vol. 32, no 1, p. 237-249, 2021.

Xu, Z., Dang, Y., & Munro, P., "Knowledge-driven intelligent quality problem-solving system in the automotive industry", *Advanced Engineering Informatics,* vol. 38, p. 441-457, 2018.

Ye, R., Pan, C., Chang, M., Yu, Q., "Intelligent defect classification system based on deep learning", *Advances in Mechanical Engineering 2018*, vol. 10, no 3, p. 1687814018766682, 2018.

Yuniarto, H., "The shortcomings of existing root cause analysis tools", *Proceedings of the World Congress on Engineering*, Vol. 3, pp. 186-191, 2012.

Zhang, Y., Luo, B., “Parallel classifiers ensemble with hierarchical machine learning for imbalanced classes”, *Proceedings of the Seventh International Conference on Machine Learning and Cybernetics*, Chine, Kunming, 12-15 July 2008, Vol. 1, pp. 94-99.

Zhu, Q., Ai X., “The Defect Detection Algorithm for Tire X-ray Images Based on Deep Learning”, *2018 3rd IEEE International Conference on Image, Vision and Computing*, Chongqing, China, 27-29 June 2018, pp. 138-142.

Zikopoulos, P., & Eaton, C., “Understanding big data: Analytics for enterprise class hadoop and streaming data”, 1st. ed., McGraw-Hill Osborne Media, 2011.

ANNEXE A TAUX DE DÉFAUTS DES VOITURES EN FONCTION DES LABELS

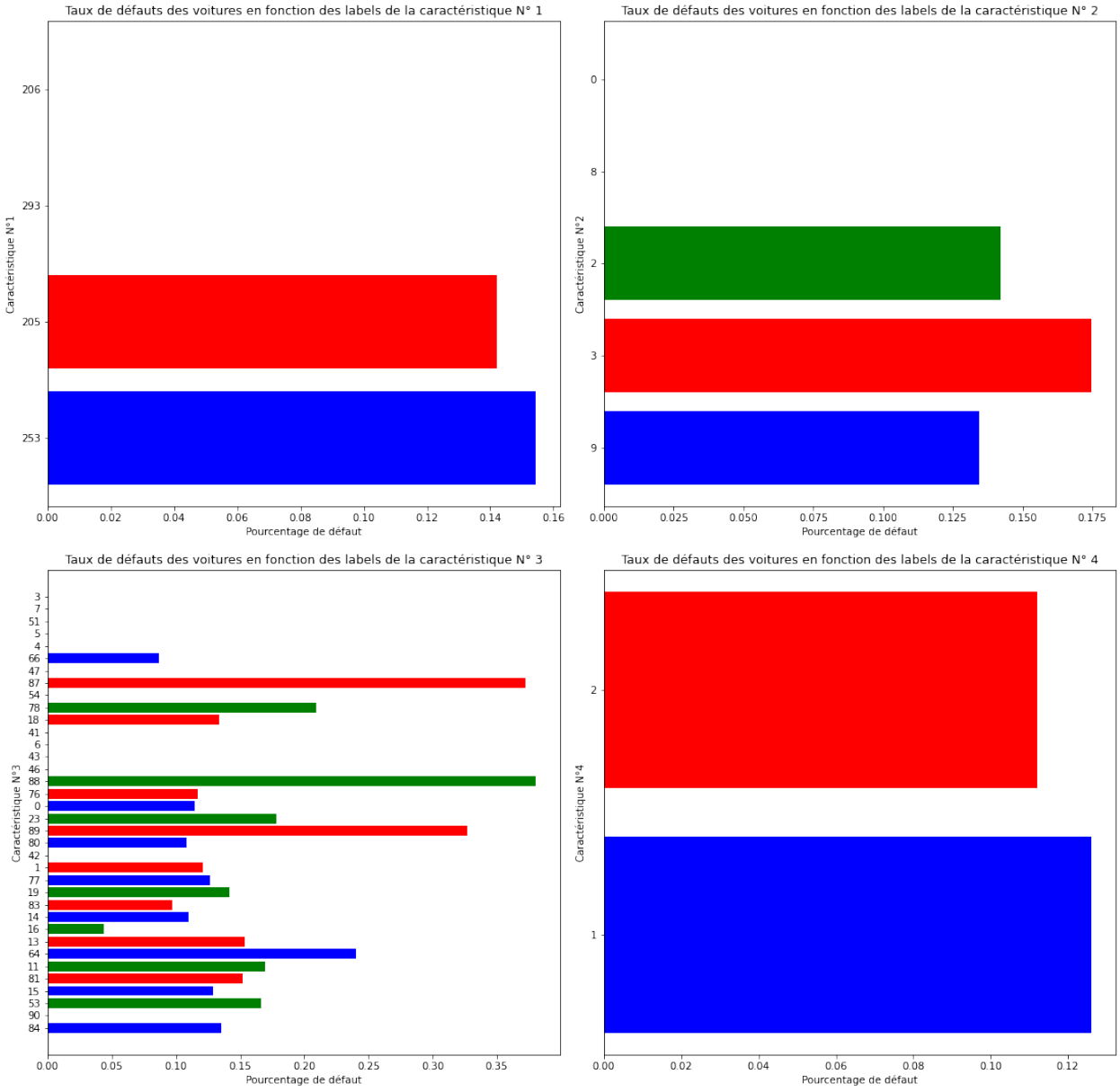


Figure A.1 Pourcentage de défaut des voitures selon les caractéristiques des labels 1 à 4

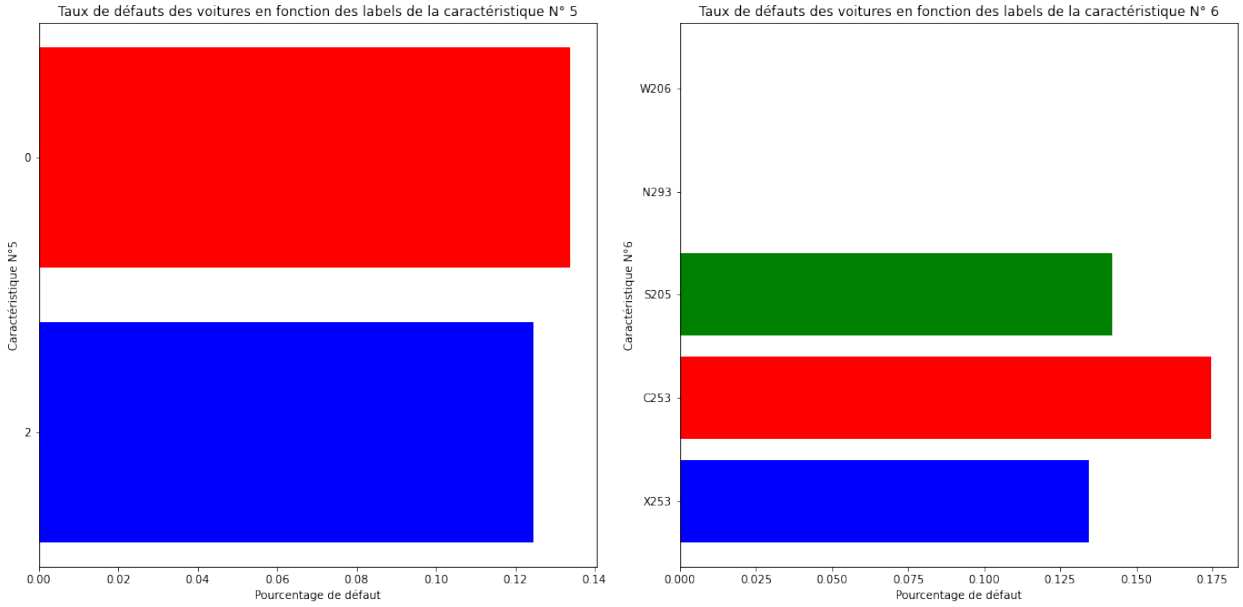


Figure A.2 Pourcentage de défaut des voitures selon les caractéristiques des labels 5 et 6

ANNEXE B TAUX DE DÉFAUTS DES VOITURES PERTINENTES EN FONCTION DES LABELS

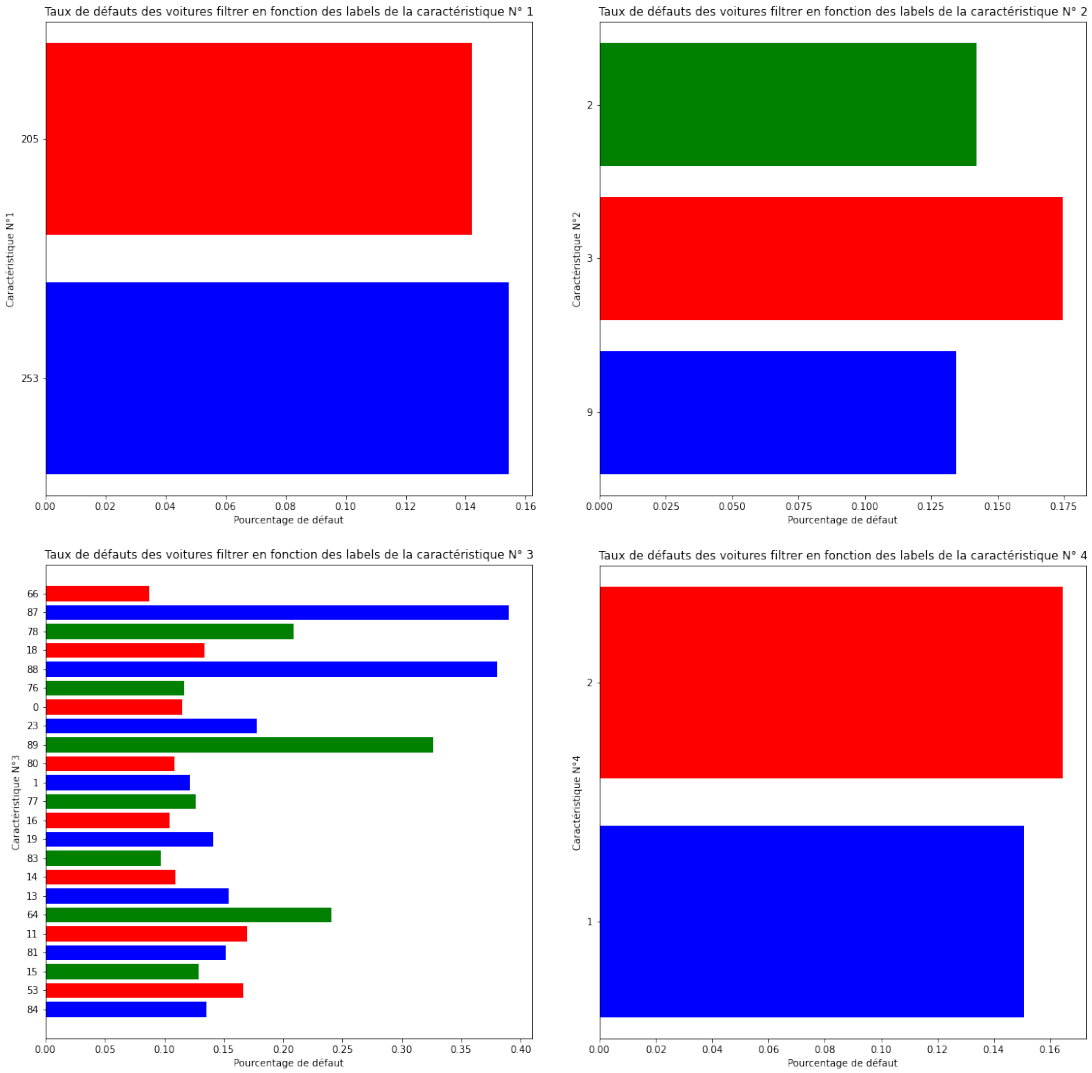


Figure B.1 Pourcentage de défaut des voitures filtrer selon les caractéristiques des labels 1 à 4

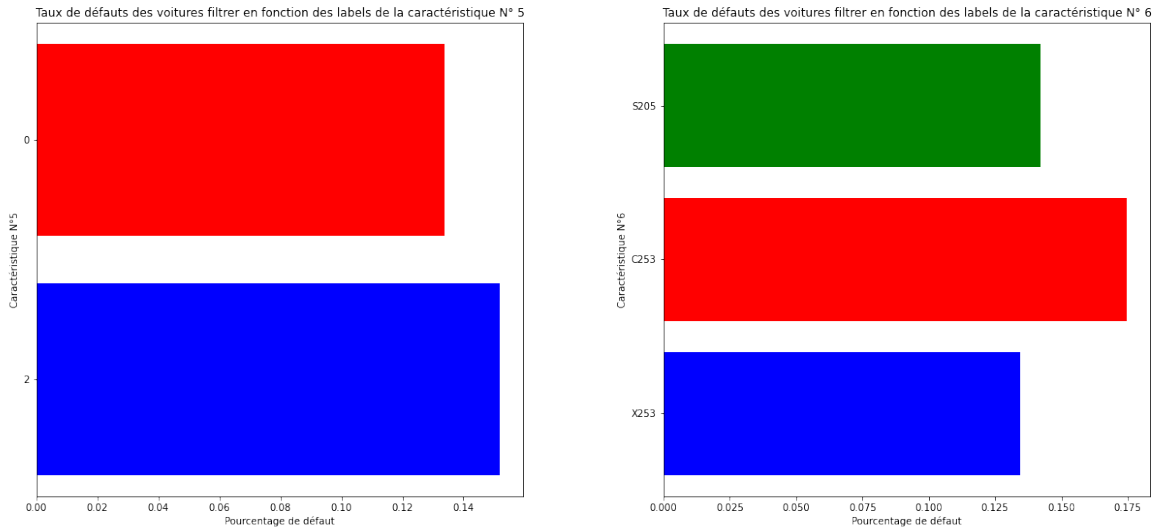


Figure B.2 Pourcentage de défaut des voitures filtrer selon les caractéristiques des labels 5 et 6

ANNEXE C MÉTRIQUE M DES VOITURES EN FONCTION DES LABELS

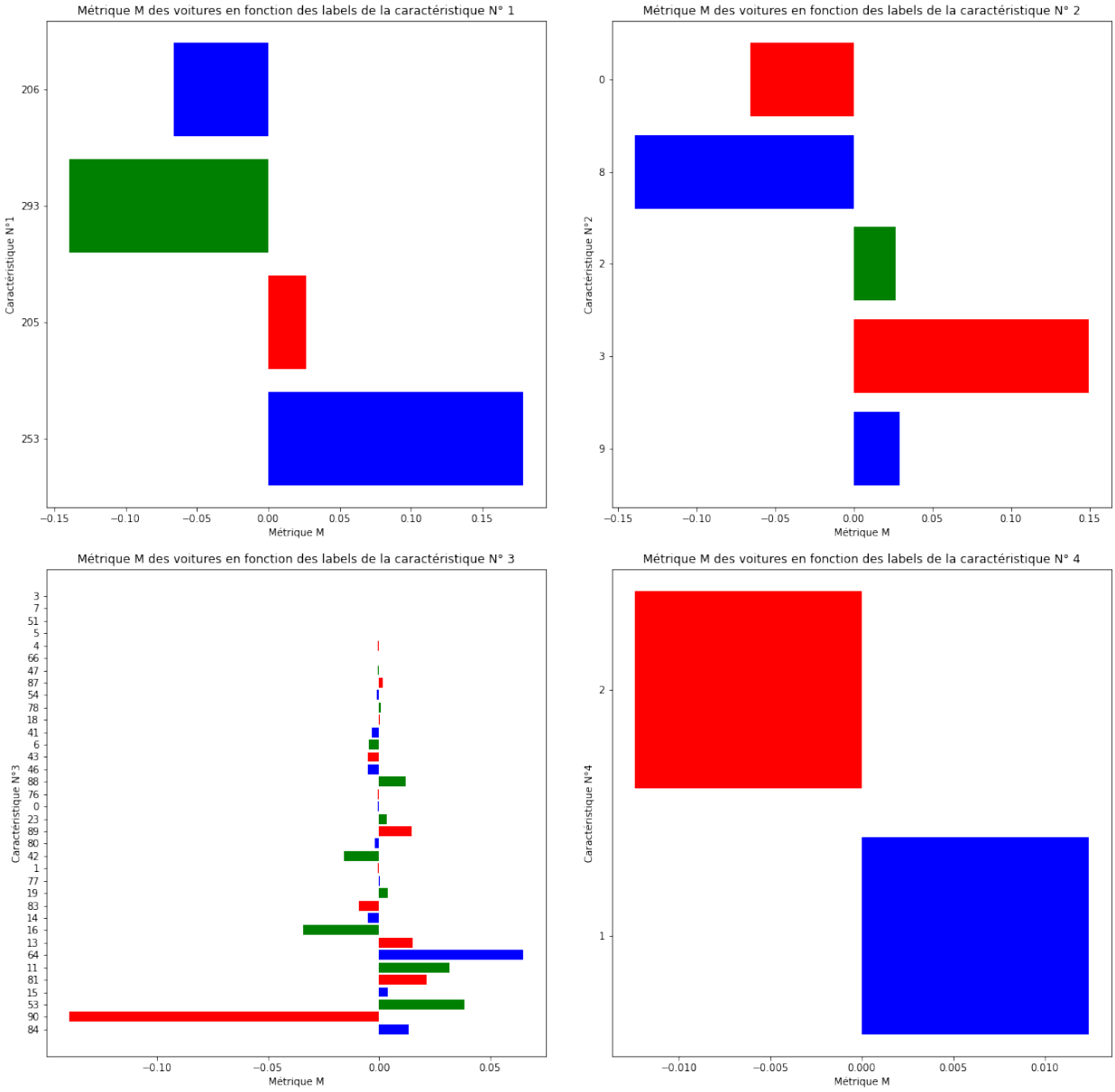


Figure C.1 Métrique M des voitures selon les caractéristiques des labels 1 à 4

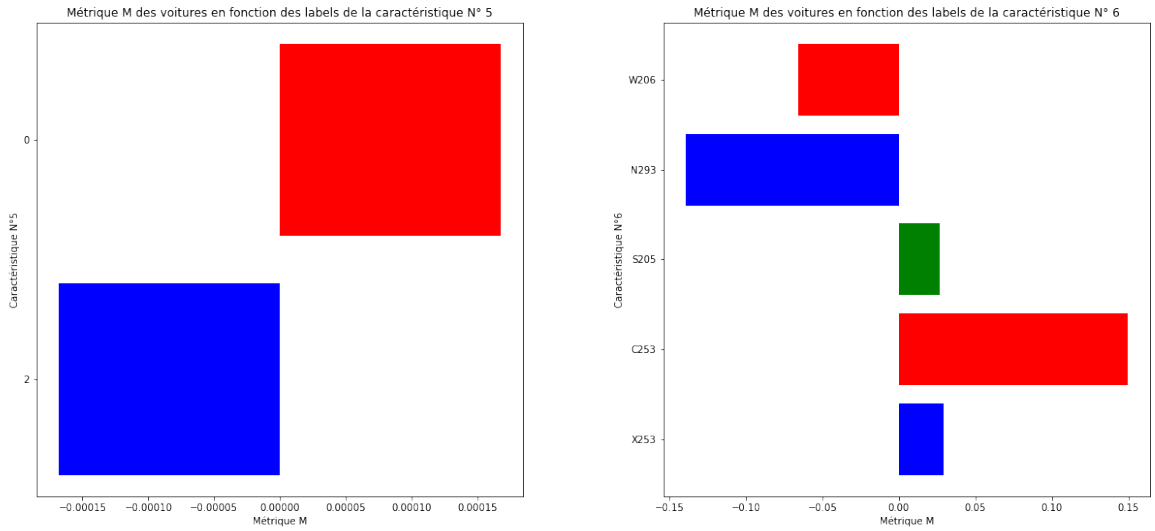


Figure C.2 Métrique M des voitures selon les caractéristiques des labels 5 et 6

ANNEXE D MÉTRIQUE M DES VOITURES PERTINENTES EN FONCTION DES LABELS

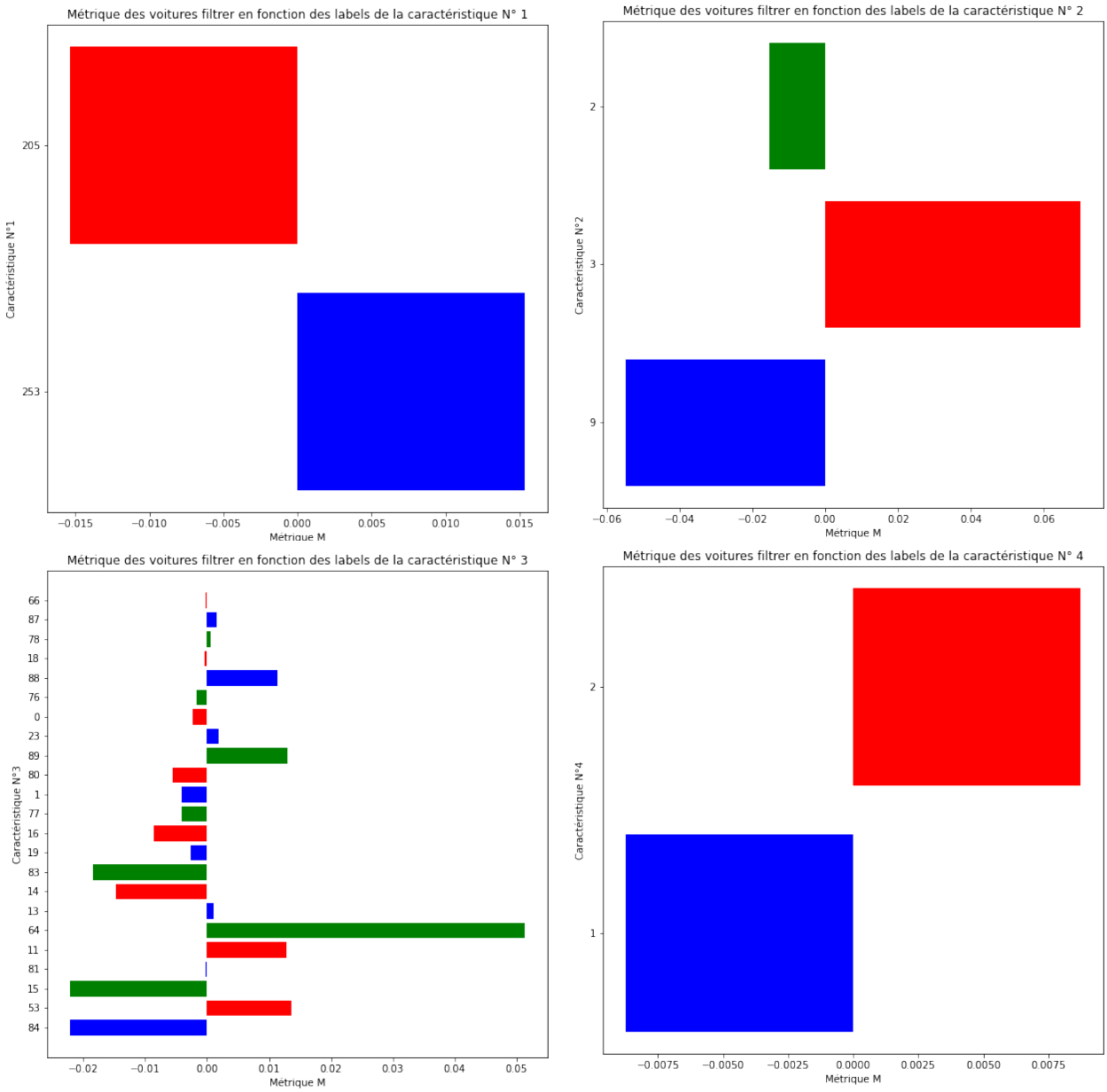


Figure D.1 Métrique M des voitures pertinentes selon les caractéristiques des labels 1 à 4

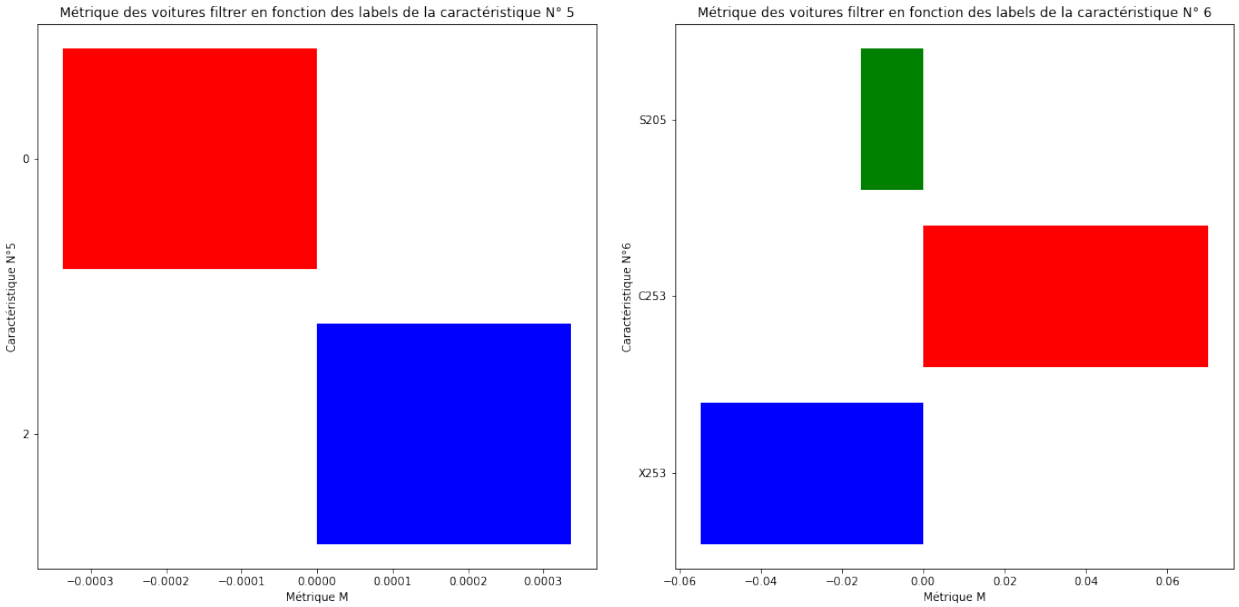


Figure D.2 Métrique M des voitures pertinentes selon les caractéristiques des labels 5 et 6

ANNEXE E ARBRE DE DÉCISION COMPLETS

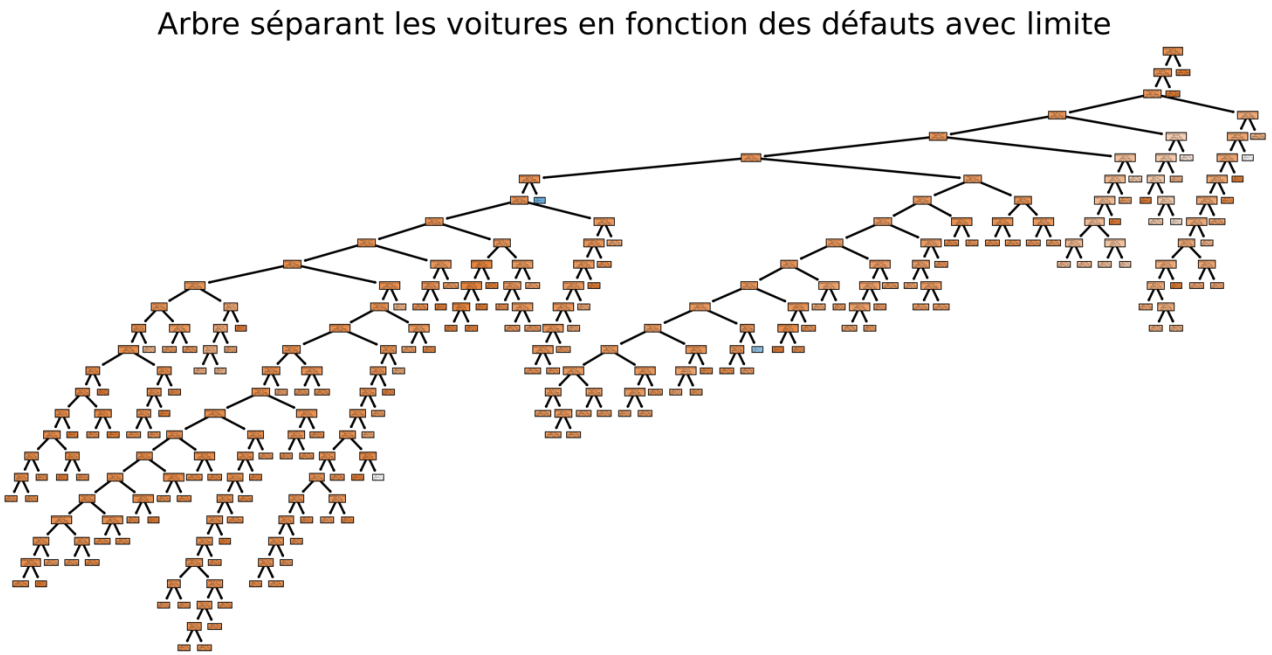


Figure E.1 Arbres de décisions de la phase d'explorations des données complets

ANNEXE F GRAPHIQUE DES MÉTHODES DE LA SILHOUETTE DE L'ÉTUDE DE CAS 1

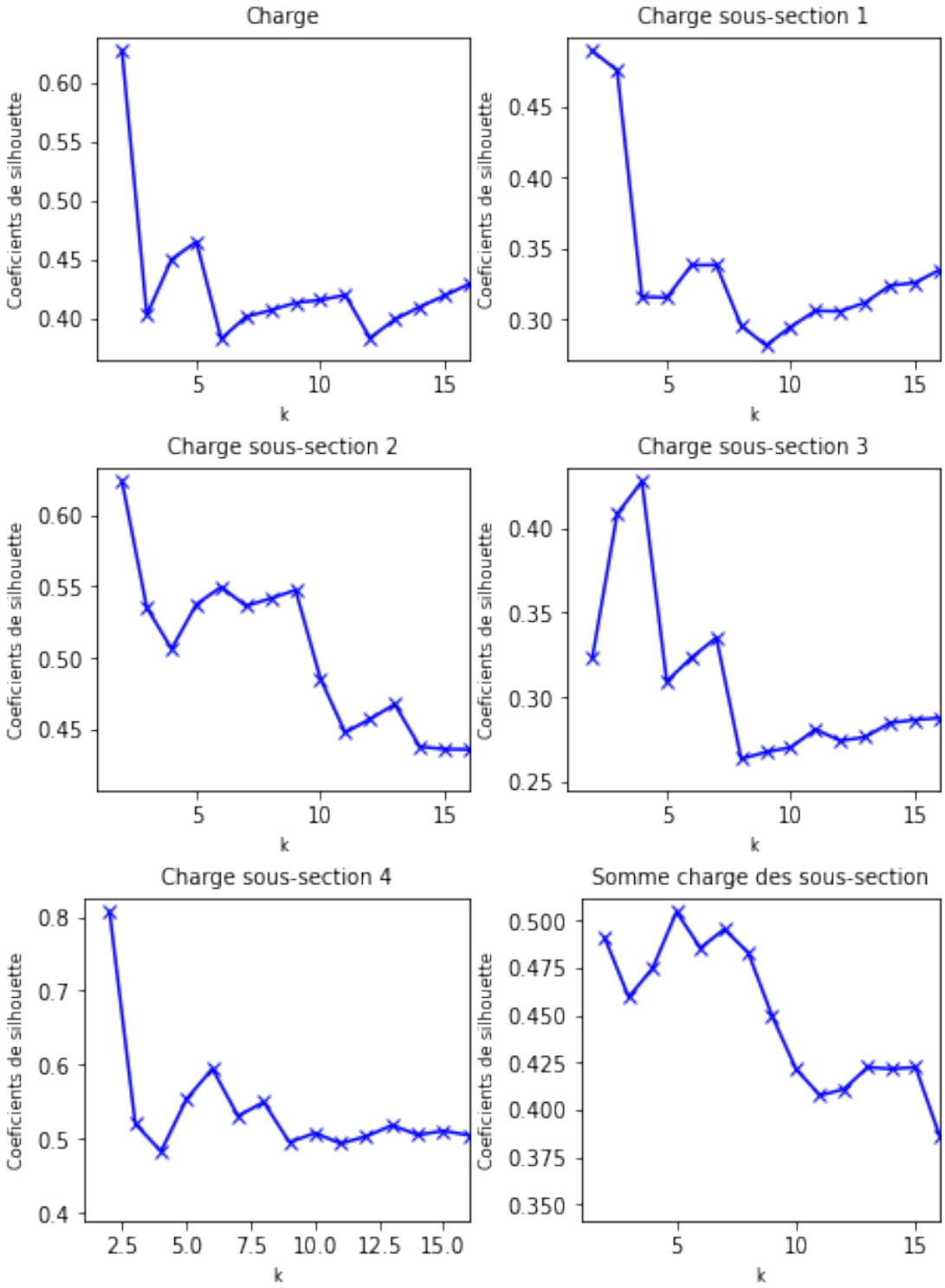


Figure F.1 Courbe des coefficients de silhouette pour le partitionnement

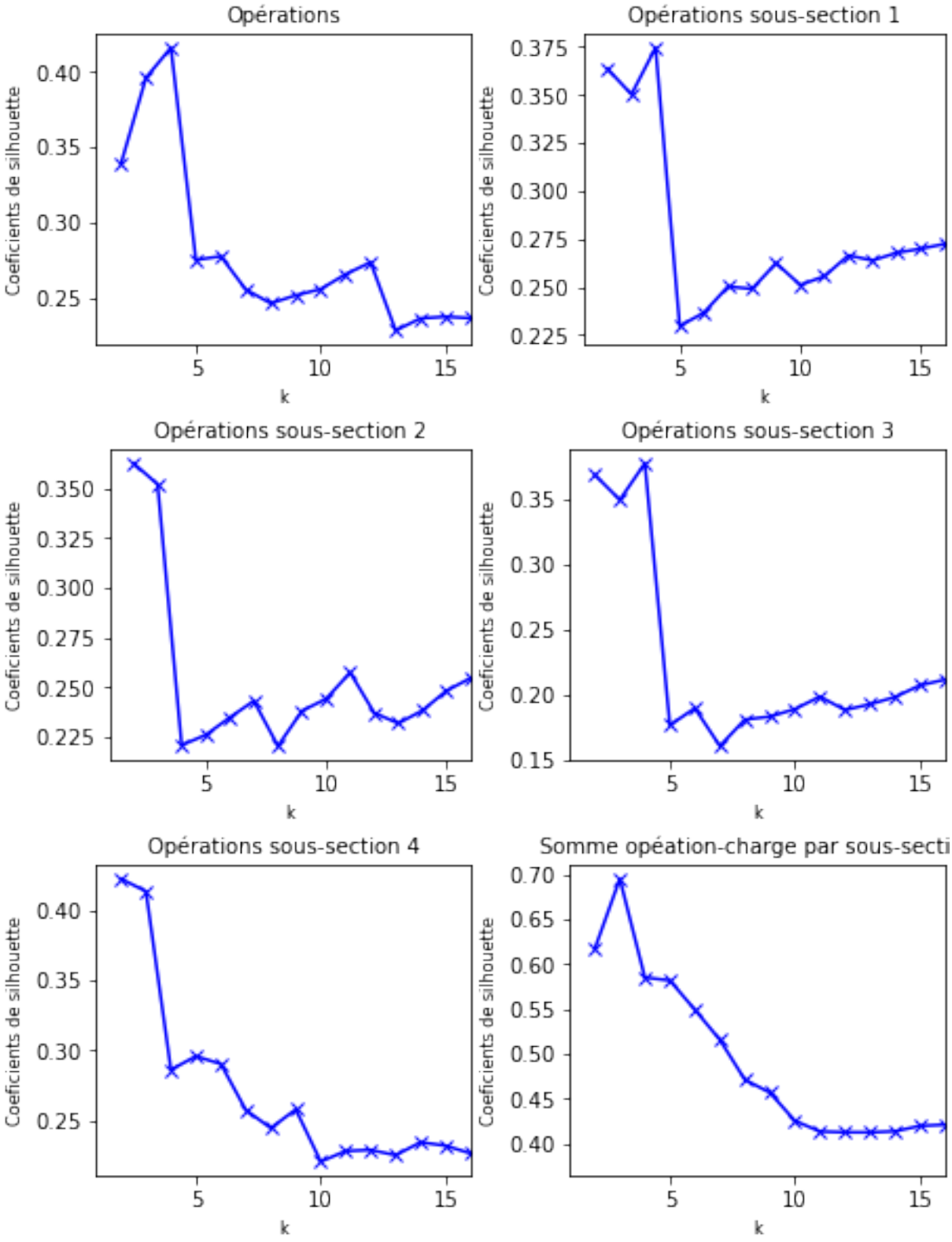


Figure F.2 Courbe des coefficients de silhouette pour le partitionnement

ANNEXE G : COUPLES DE CARTES DE DENSITÉ DES TAUX DE DÉFAUTS DE L'ÉTUDE DE CAS 1

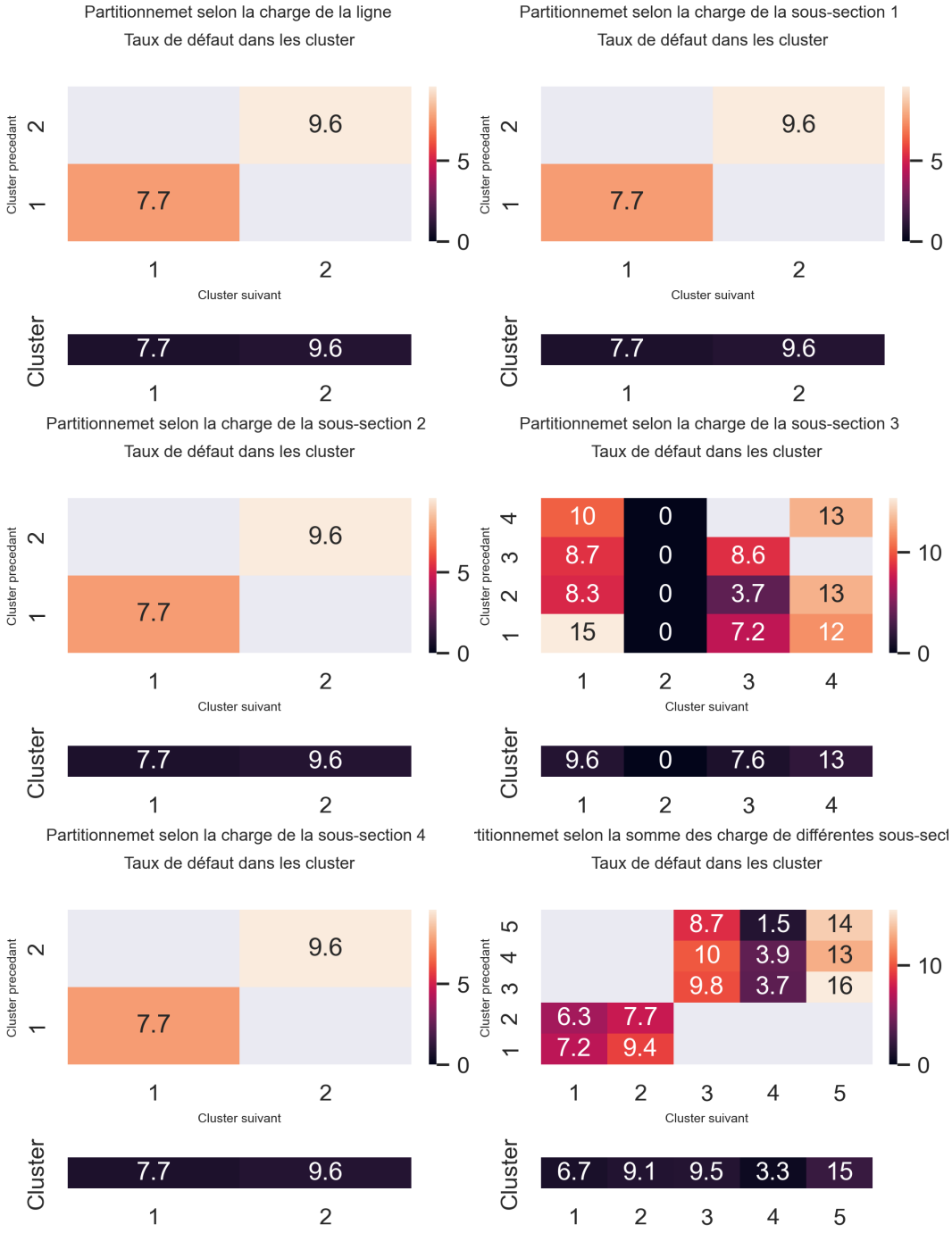


Figure G.1 Couples de carte densité

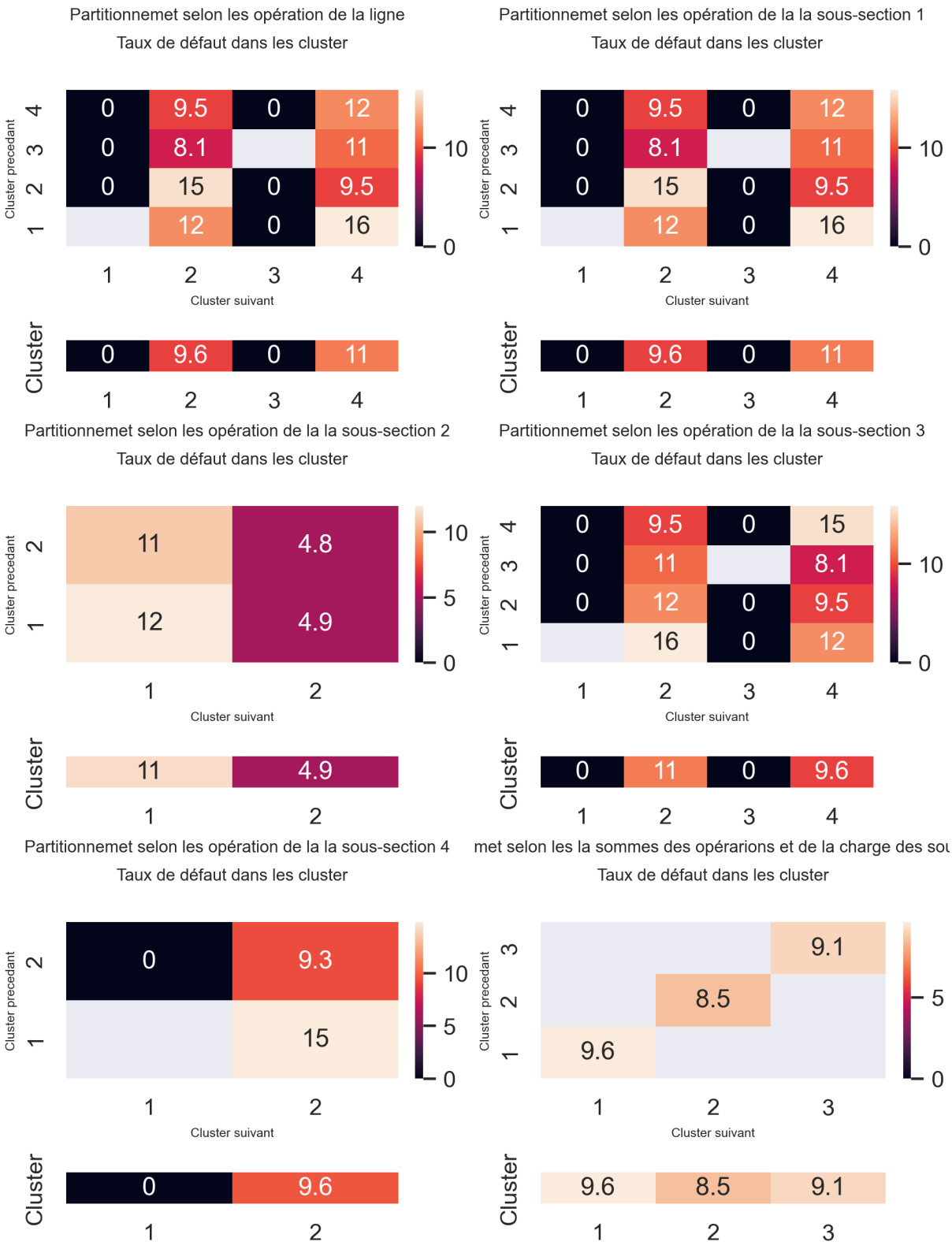
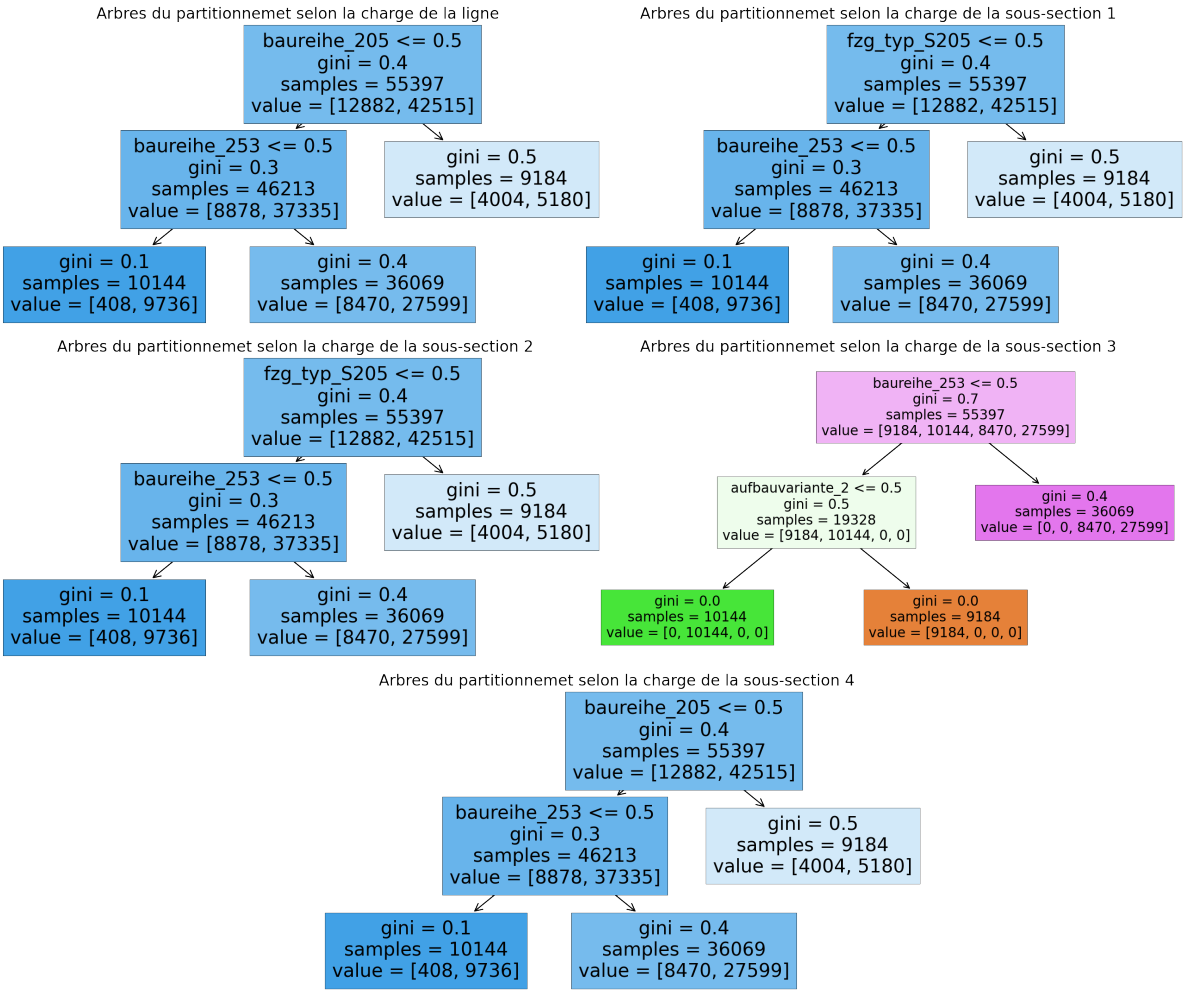


Figure G.2 Couples de carte densité

ANNEXE H ARBRES DE DÉCISION DE L'ÉTUDE DE CAS 1



Arbres du partitionnemet selon la somme des charge de différentes sous-section

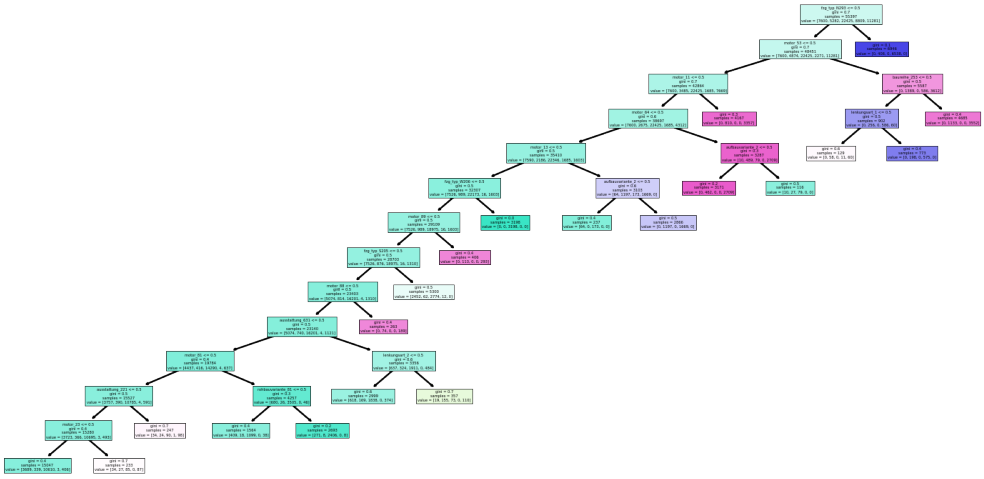


Figure H.1 Arbres de décision de l'étude de cas 1

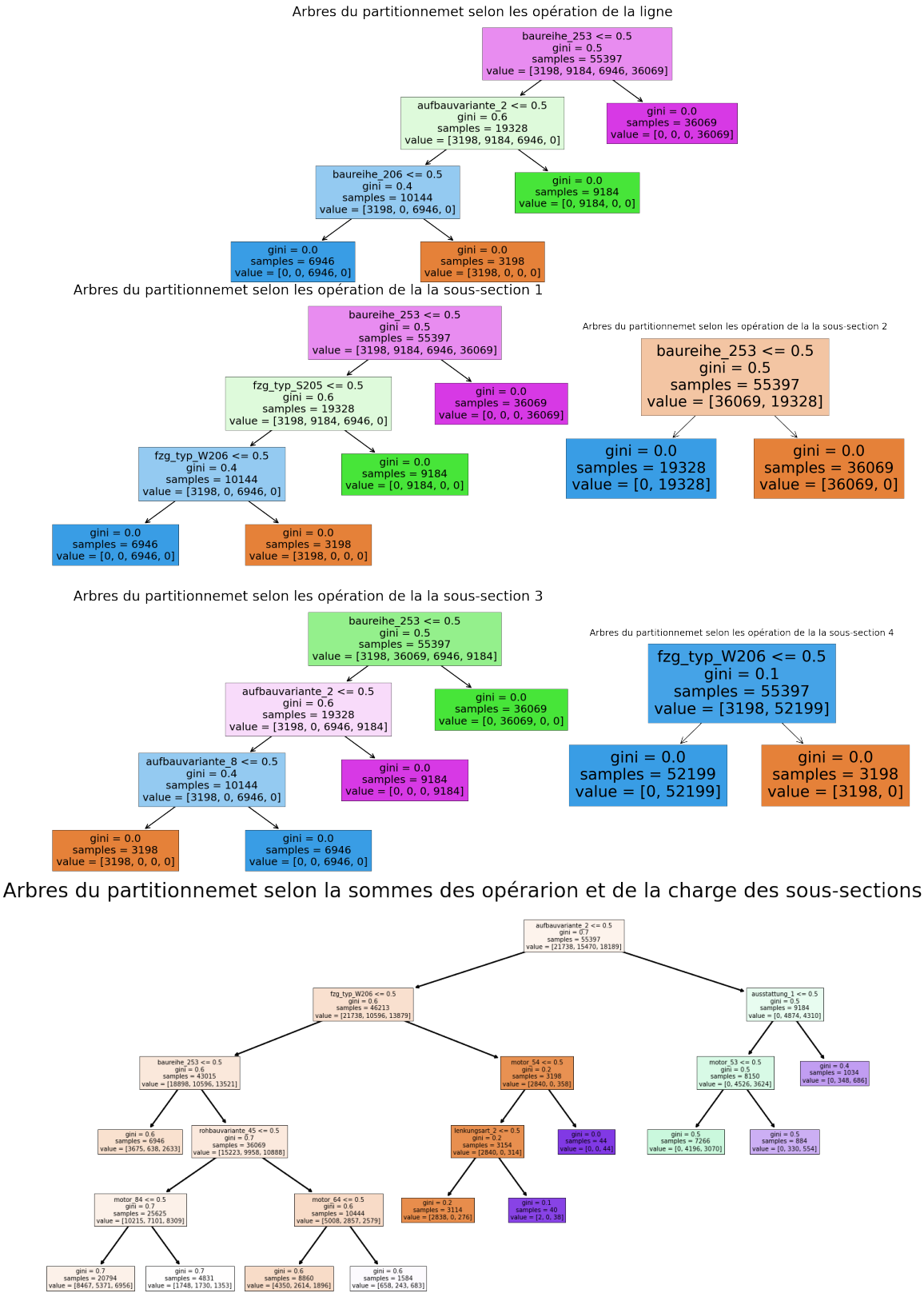


Figure H.2 Arbres de décision de l'étude de cas 1

ANNEXE I ARBRES DE DÉCISION DE L'ÉTUDE DE CAS 2

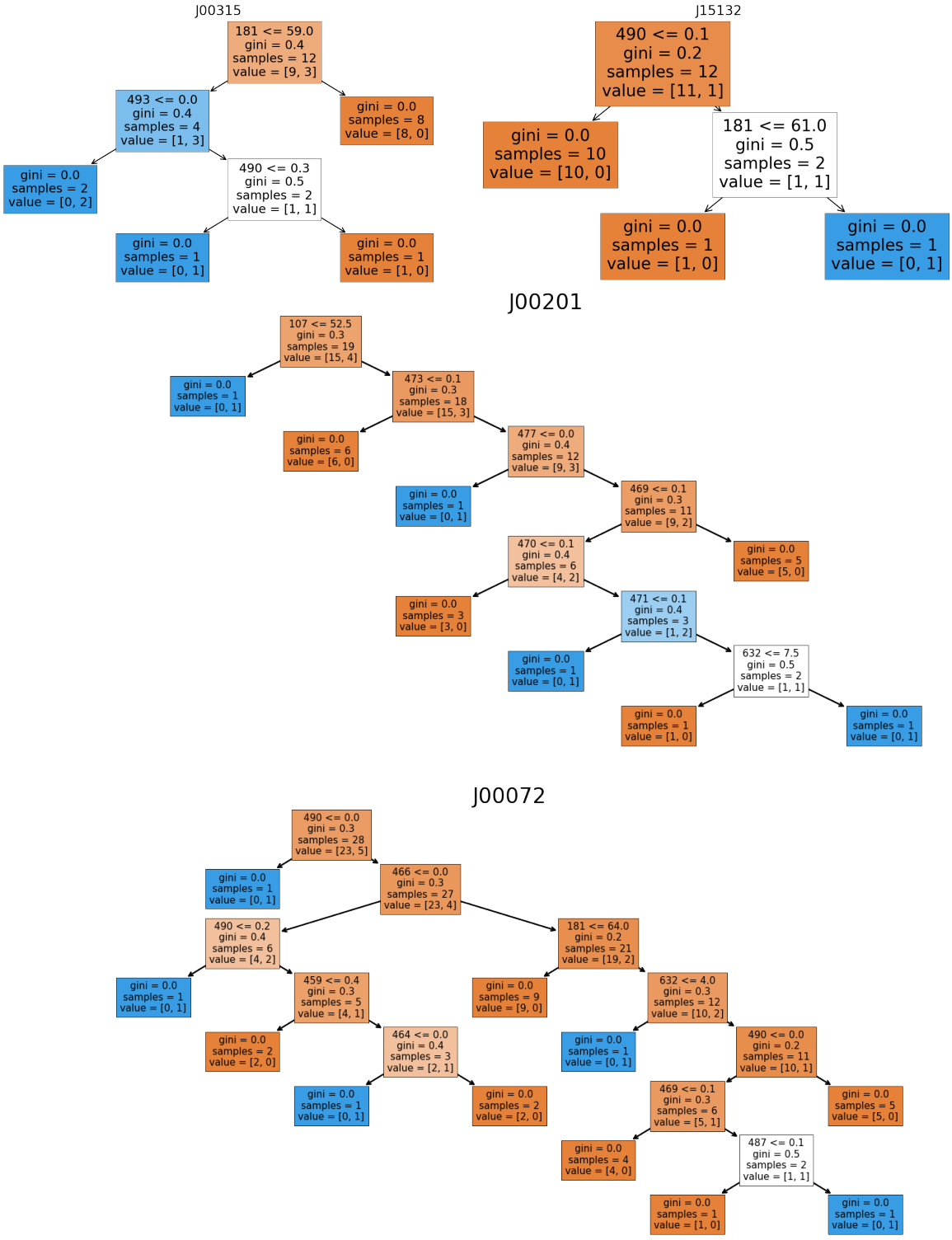


Figure I.1 Arbres de décision de l'étude de cas 2

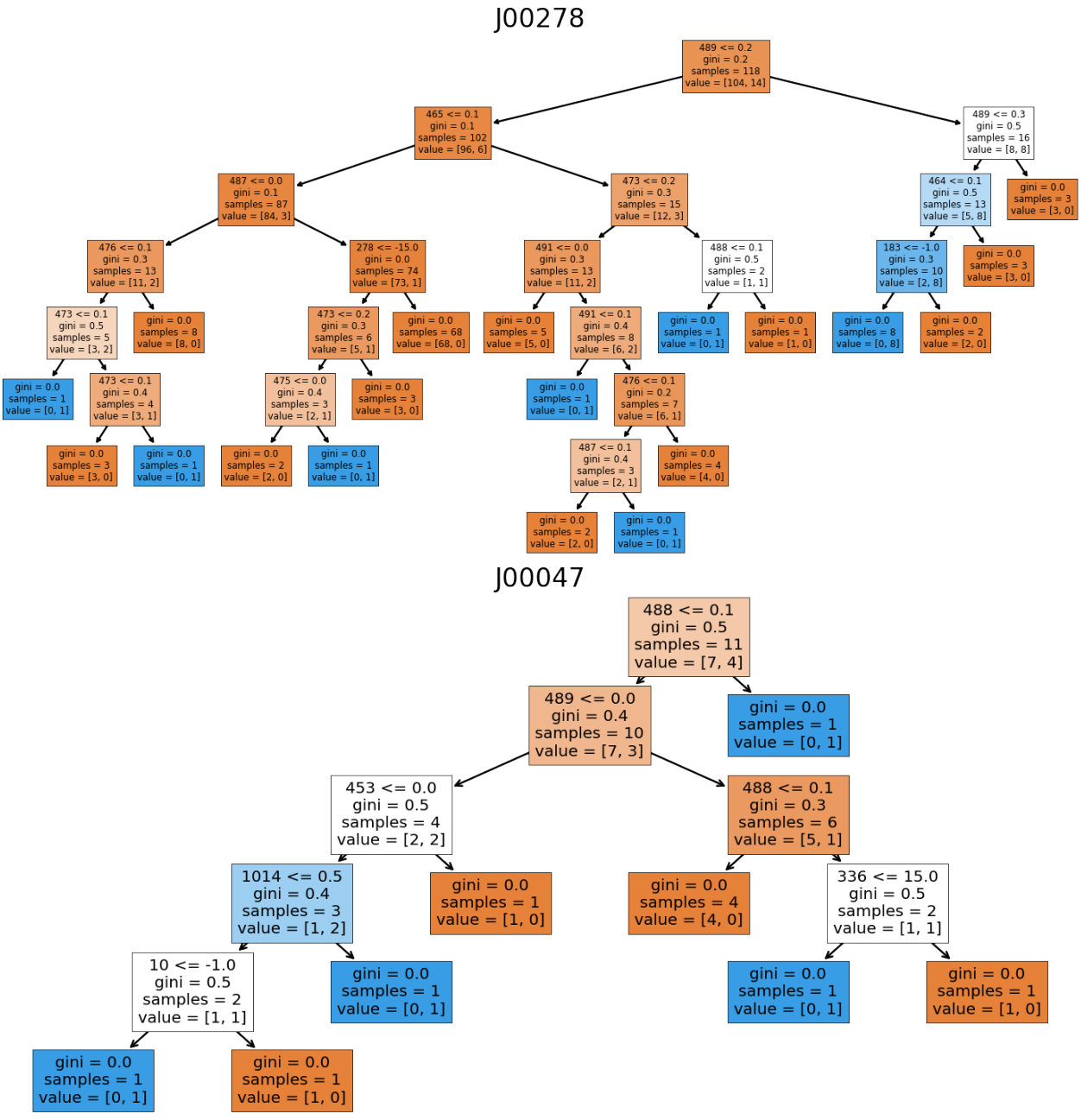


Figure I.2 Arbres de décision de l'étude de cas 2

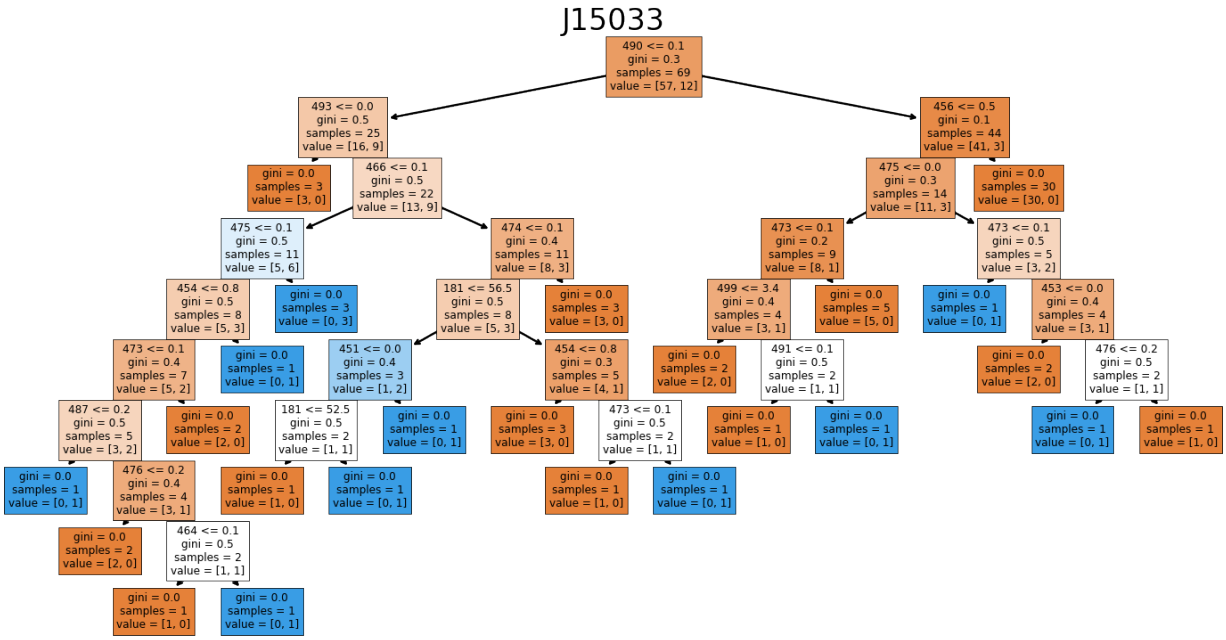


Figure I.3 Arbres de décision de l'étude de cas 2