Institut pour la Maîtrise des Risques
Sûreté de Fonctionnement - Management - Cindyniques

Analyse des retours d'expérience en clientèle par des méthodes de traitement automatique du langage naturel pour identifier les modes de défaillance

Customer feedback analysis using natural processing methods to identify failure modes

Dr Gwenaël EDELINE
SOM LIGERON
Toulouse
gwenael.edeline@ortec.fr

Résumé — Les méthodes d'intelligence artificielle (IA), notamment le traitement automatique du langage (NLP), offrent des possibilités révolutionnaires pour l'analyse des retours d'incidents en clientèle, qui permettent en particulier d'évaluer la fiabilité réelle d'un système en clientèle et de faire une projection de fiabilité à partir de l'identification préalable d'une loi de Weibull sur les incidents. Le présent article détaillera un outil d'IA développé pour attribuer automatiquement un mode de défaillance à chaque retour client, en exploitant des techniques de NLP et de modélisation par Machine Learning (ML). L'approche proposée de prétraitement des données et de classification des retours clients sera décrite. Les résultats obtenus démontreront l'efficacité de notre approche pour faciliter le travail des ingénieurs FMDS et fournir un traitement des réclamations clients précis et rapide.

Mots-clefs — Intelligence artificielle, Traitement automatique du langage, Analyse des retours clients, Modes de défaillance en clientèle, Apprentissage automatique.

Abstract — Artificial intelligence (AI) methods, particularly natural language processing (NLP), offer revolutionary opportunity for customer incident feedback analysis, especially for assessing the reliability of a system in the customer domain and to make a reliability projection through a Weibull law analysis. This article will detail an AI tool developed to automatically assign a failure mode to each customer feedback by leveraging NLP techniques and ML modelling. The proposed approach to data pre-processing and customer feedback classification will be described. The results obtained will demonstrate the effectiveness of the developed AI tool in facilitating the work of RAMS engineers and providing accurate customer processing in a short amount time.

Keywords — Artificial Intelligence, Natural Language Processing, Customer Feedback Analysis, Failures modes, Machine Learning.

I. INTRODUCTION

Les méthodes d'intelligence artificielle (IA) permettent aux entreprises d'analyser des données volumineuses et de faciliter leur exploitation. Le traitement automatique du langage (NLP : Natural Language Processing) est un sous-domaine de l'IA qui permet l'analyse rapide, précise et efficace des écrits, et permet aux entreprises de comprendre les données textuelles clients à grande échelle en temps réel (Ramaswany et al., 2018).

L'exploitation des données de retour d'expérience (REX), liées aux incidents (dans le sens d'une défaillance ici) subit par un système donné en clientèle, est nécessaire pour estimer (via une analyse de Weibull sur les incidents correspondants à chacun des modes de défaillance du système considéré) et améliorer la fiabilité dudit système, ainsi que celles des composants le

constituant. Si l'on considère le cas de l'industrie automobile, l'analyse multilinguistique des retours textuels clients et des diagnostics ateliers nécessite des ressources humaines particulièrement conséquentes, en particulier afin de mettre en évidence les différents modes de défaillance de chacun des composants constitutifs d'un véhicule. Pour l'exemple, rien que le groupe automobile Toyota a commercialisé 11,2 millions de véhicules dans le monde en 2023. Et chaque véhicule est composé de 30 000 à 90 000 composants.

Pour répondre à cette problématique, nous avons développé un outil d'IA qui attribue un mode de défaillance à chaque retour client, en analysant automatiquement le langage (en multilinguistique, via de la traduction automatisée) et le lexique. Notre objectif est de réduire considérablement le temps de traitement de ce type de base de données, ainsi que d'éviter les erreurs humaines inhérentes à ce type de classification.

L'exploration de la base de données synthétique considérée a permis d'identifier des caractéristiques particulières : la corrélation des défaillances observées avec le type de composants, la nature du moteur, la distance parcourue, ... Ensuite, des variables appropriées ont été sélectionnées à partir de critères métiers pour construire les modèles d'IA. Ceux-ci traitant des données numériques, les variables textuelles ont donc été numérisées, en procédant en plusieurs étapes qui tiennent compte de notre lexique spécifique automobile.

Les modèles de Machine Learning sélectionnés ont été entrainés sur des données prétraités afin d'obtenir des modèles performants. Il a été constaté que la variante du modèle pré-entrainé « Camembert de Bert » (Hu Y et al., 2022), pour la langue française, basée sur les méthodes « Transformers » permet le traitement automatique des retours clients. Cette approche permet de lier un incident en clientèle à un mode de défaillance à partir du contexte des mots dans les phrases des retours clients ou garagistes.

Il a été mis en évidence dans nos travaux qu'entrainer un modèle de type « Camembert » directement sur nos données augmentait considérablement les performances du modèle choisi. Cependant l'entrainement d'un modèle « Transformer », tel que « Camembert », demande une quantité importante de données. Lorsque ce n'est pas le cas, il est possible d'appliquer la méthode de Fine-Tuning (Chi Sun et al., 2019), en ajustant les poids du modèle en fonction des données.

A partir du modèle développé, une application de classification automatique des retours clients a été mise en place. Celle-ci permettra par la suite de réaliser des analyses de Weibull pour tous les modes de défaillance identifiés, puisque tous les commentaires clients, liés à un système considéré, ont été attribués au préalable à un mode de défaillance à partir du modèle d'IA développé.

Nos travaux permettront d'illustrer l'apport des modèles d'IA pour faciliter les tâches des ingénieurs FMDS en fournissant rapidement des résultats précis dans le traitement des retours clients. Nos travaux seront illustrés par le traitement d'une base de données synthétique du secteur automobile.

II. ETAT DE L'ART

L'analyse des retours textuels de clients par des méthodes de traitement automatique du langage naturel (NLP), afin d'automatiser l'identification des sentiments globaux, est un domaine de recherche et d'application de plus en plus important, en particulier dans les domaines du service client, de la gestion de la qualité et de l'amélioration des produits. Cette technologie est largement utilisée dans divers secteurs industriels, notamment les technologies de l'information, les télécommunications, la santé, etc… Son application permet d'aider les entreprises à acquérir des informations sur la satisfaction des clients, d'identifier les domaines à améliorer, ainsi qu'à prendre des décisions éclairées, basées sur des données factuelles, pour améliorer la qualité des produits et de l'expérience client globale (Ganesan et al., 2023). Ces méthodes sont également utiles pour suivre les tendances et surveiller l'e-réputation d'une marque en temps réel.

Les entreprises reçoivent souvent des retours d'expérience clients signalant des problèmes récurrents ou des modes de défaillance spécifiques. Les méthodes de NLP peuvent être utilisées pour analyser ces retours et identifier les tendances ou les schémas communs. Par exemple, une étude (Deepa et al., 2015) a utilisé des techniques d'extraction d'informations pour identifier les problèmes courants signalés par les clients et mieux comprendre les émotions des clients, grâce à l'analyse des sentiments.

Les problèmes signalés par les clients peuvent évoluer avec le temps en raison de modifications dans les produits, les politiques de l'entreprise, etc… Les méthodes NLP peuvent être utilisées pour suivre l'évolution de ces problèmes au fil du temps, en analysant les retours d'expérience des clients à intervalles réguliers. Par exemple, une étude (Min et al., 2012) a utilisé des techniques d'analyse de texte pour suivre les tendances et les changements dans les retours d'expérience des clients de produits électroniques au fil du temps.

De nombreuses entreprises collectent des avis clients sur des sites Web, des forums ou des plateformes de médias sociaux. Les méthodes de NLP sont utilisées pour analyser ces avis en identifiant les points positifs et négatifs mentionnés par les clients. (Chin Chen et al. 2011) proposent une méthode pour évaluer la qualité des informations contenues dans les avis sur des produits. L'article traite de l'évaluation de la qualité des revues comme un problème de classification et utilise un cadre efficace d'analyse de qualité de l'information pour extraire les caractéristiques représentatives des revues.

Ces exemples illustrent comment les méthodes de traitement automatique du langage naturel sont utilisées pour analyser les retours d'expérience client. Ces techniques peuvent aider les entreprises à améliorer leurs produits et services, à mieux comprendre les besoins des clients et à prendre des décisions plus éclairées en matière de gestion de la qualité.

Dans nos travaux, nous souhaitons utiliser ces méthodes NLP pour l'analyse et la classification des retours clients et des commentaires ateliers liés à des défaillances de sous-systèmes ou de composants d'une automobile. Bien évidemment, ces travaux peuvent être étendus via quelques adaptations à d'autres secteurs industriels. En effet, les analyses de fiabilité en vie-série et la consolidation du REX en clientèle nécessitent la bonne compréhension des défaillances en clientèle et l'attribution de chacune d'elle à un mode de défaillance déterminé.

III. METHODOLOGIE

Notre méthodologie de recherche (Fig.1) s'est articulée autour de cinq étapes principales, chacune jouant un rôle crucial dans le processus de classification des modes de défaillance à partir des retours des clients et des ateliers.

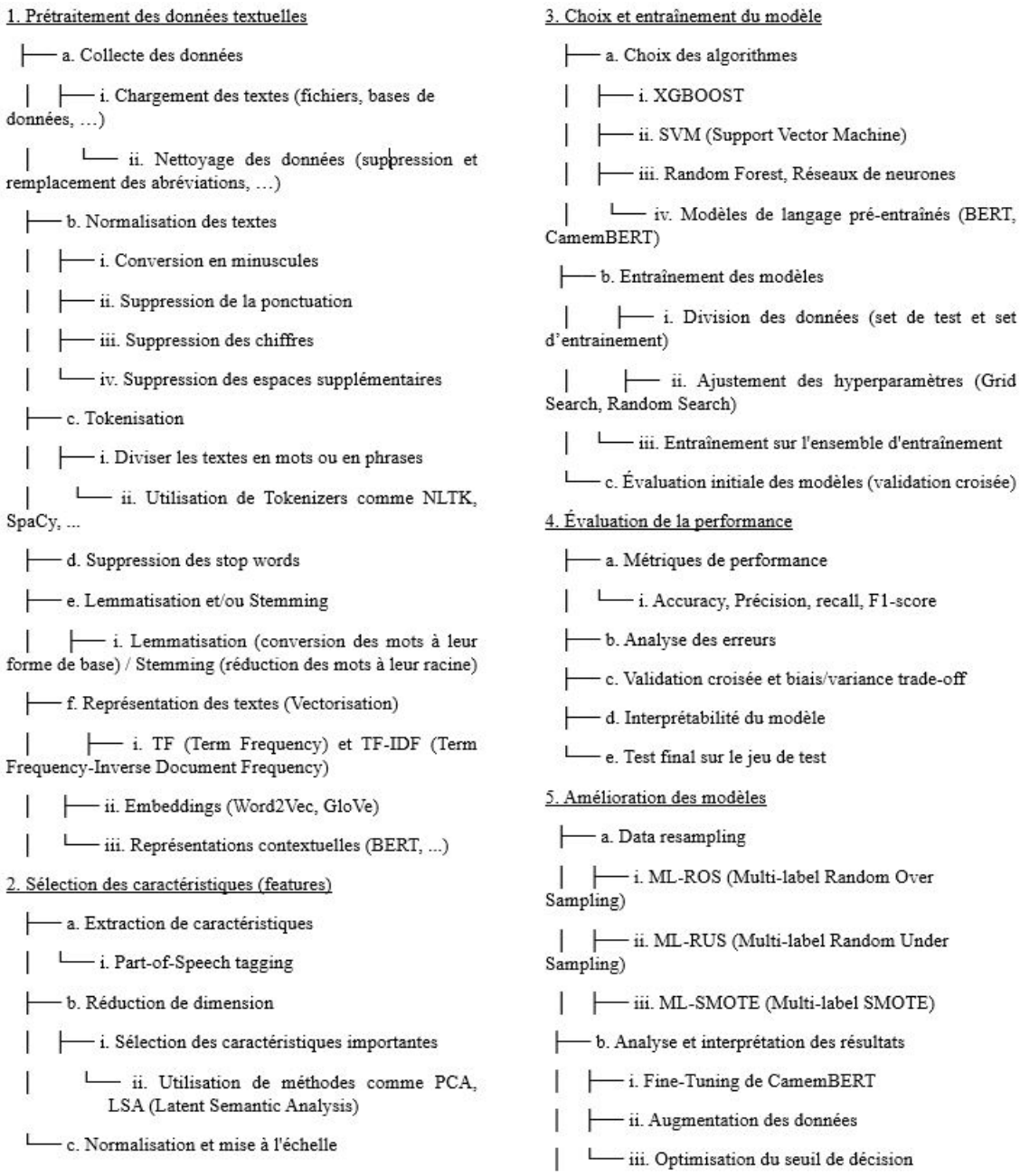

Fig. 1. Synthèse de notre processus de classification supervisé

Tout d'abord, une attention particulière a été consacrée au prétraitement des données textuelles. Ce processus a impliqué une série d'étapes visant à nettoyer, normaliser et préparer les données pour l'entraînement des modèles d'IA qui seront appliqués.

Parmi ces étapes, la suppression des caractères spéciaux, la correction des fautes d'orthographe, la lemmatisation et la vectorisation des données ont été effectuées afin de garantir la qualité et la cohérence des données utilisées.

Ensuite, les variables à inclure dans le modèle d'IA ont été soigneusement sélectionnées. Cette étape a été guidée par une analyse approfondie des critères de sélection, en mettant en évidence les informations les plus pertinentes pour la classification des modes de défaillance. Des facteurs, tels que la fréquence des termes, la spécificité des mots-clés et la pertinence contextuelle, ont été pris en compte pour identifier les variables les plus significatives.

Le choix et l'entraînement du modèle ont constitué la troisième phase de la méthodologie. Pour cela, plusieurs modèles de Machine Learning ont été explorés, notamment les Machines à Vecteurs de Support (SVM), les « Random Forests » et « XGBOOST ». Chaque modèle a été entraîné sur les données (80% des données de REX sont utilisées) à l'aide de techniques d'apprentissage supervisé, en ajustant les hyper-paramètres et en optimisant les performances pour maximiser la précision de la classification.

Enfin, la performance de chaque modèle d'IA étudié a été évaluée à l'aide de métriques : plus un modèle est précis (mesure le pourcentage des classifications correctes parmi les classes positives détectées par rapport aux classes positives totales), moins il a de chances de détecter l'ensemble des classes positives ; plus il a un rappel (pourcentage de classes positives trouvées) élevé, moins il a des chances d'être précis. A cet effet, nous avons fait le choix d'une métrique, construite à partir de la précision et du rappel et qui permet d'avoir le meilleur compromis entre les deux : le « F1-Score » (Powers, 2011). Celui-ci a été utilisé pour mesurer la qualité des prédictions et l'efficacité de l'approche. Ces évaluations ont permis de valider la robustesse du modèle sélectionné et d'identifier les potentielles améliorations nécessaires pour de futures itérations du système de classification des modes de défaillance.

A. Présentation des données

Les données étudiées comprenaient 11 434 lignes d'observations de REX dans des langues différentes, sur des pannes de véhicules automobiles. Ces données ont également été classifiées manuellement par des ingénieurs fiabilistes en trois familles de verbatims : Effet(s) Client, Mode(s) de défaillance (une dizaine sont identifiés) et Solution(s) appliquée(s).

Les variables dans les bases de données fournissent des informations concernant les éléments suivants : identifiant du type de véhicule, identifiant commercial, identifiant moteur, identifiant véhicule, centre de production du véhicule, date d'assemblage de la voiture, date de début et de fin de garantie, date d'intervention en atelier, coût de l'intervention, pays où la panne a eu lieu, commentaires émis par le client, commentaire émis par l'atelier de réparation, commentaire indiquant la solution à la panne, catégorie prédéfinies de pannes, identification des composants ayant été remplacés, fournisseur de la pièce défectueuse, kilométrage parcouru au moment de la panne, …

Dans la suite, les variables qui ont été sélectionnés sur la base de critères métiers pour la modélisation sont décrits dans Tab.I.

TABLE I. VARIABLES CHOISIES POUR LA MODELISATION : VARIABLES EXPLICATIVES ET CIBLES

Variables déjà existantes dans les bases de données : variables explicatives	**Variables ajoutées manuellement par les ingénieurs fiabilistes : variables cibles (labels)**
Commentaire émis par le client	Effet(s) client
Commentaire émis par l'atelier de réparation	Mode(s) de défaillance
Commentaire indiquant la solution à la panne	Solution(s) appliquée(s)

Le modèle de traitement automatique du langage naturel choisi a été entraîné sur ces dernières variables labélisés par les variables : Effet(s) client, Mode(s) de défaillances et Solution(s) appliquée(s) pour ensuite classer automatiquement de nouveaux commentaires.

B. Prétraitement des données

Avant d'entreprendre toute analyse, une étape cruciale consiste à prétraiter les données pour les rendre adaptées à une utilisation par un modèle d'intelligence artificielle (IA). Dans cette section, les différentes étapes de prétraitement des données effectuées sont décrites.

Dans les deux bases de données utilisées (chacune correspondant au retour d'expérience clients pour un ensemble de composants et de pays), la langue française ne représente qu'environ 37% des variables choisies sur la base de critères métiers pour la modélisation. Pour les commentaires dans d'autres langues, un processus automatisé a été mis en place pour les traduire dans une langue de référence choisie (en l'occurrence, la langue française), en utilisant les outils suivants :

- Sélénium / Python (Selenium) : Un bot a été développé pour automatiser les traductions, en utilisant le Framework Selenium de Python, qui a navigué sur le site web « DeepL Translator » et récupéré les traductions ;

- DeepL Translator (Deepl) : Ce service de traduction en ligne a été utilisé pour traduire les commentaires textuels présents dans notre base de données.

Cette approche a permis de préparer les données pour l'analyse ultérieure en NLP et de surmonter les défis liés à la variété des langues et à la diversité des verbatims. L'impact d'une erreur de traduction par l'outils Deepl ne sera pas étudié ici.

La mise en commun des verbatims (Tab.II) des différentes bases de données (correspondant à des sous-systèmes différents) a été réalisée par un Data Scientist à l'aide de tables de mappings Excel. Par ailleurs, nous avons identifié les verbatims synonymes et les verbatims isolés, qui existaient uniquement dans une seule des bases de données disponibles :

- Verbatims synonymes : Des tables de mappings ont été créées pour regrouper les verbatims ayant des noms différents ;
- Verbatims isolés : Les verbatims uniques ont été soit regroupés avec des verbatims similaires, soit simplement ajoutés comme nouveaux verbatims aux tables de mappings.

TABLE II. EXEMPLE DE MISE EN COMMUN DES VERBATIMS

Base de données 1	Base de données 2	Mise en commun
Echec de démarrage véhicule bloqué	Véhicule ne démarre pas	Véhicule ne démarre pas
Arrêter/verrouiller/bloqué le véhicule pendant la conduite	Arrêt moteur	Arrêt moteur
Pas de puissance	Perte de puissance	Pas de puissance

La concaténation des bases de données a été faite à l'aide des tables de mappings crées lors de l'étape de mise en commun des verbatims. Des scripts Python ont été écrits pour manipuler les données (à l'aide des bibliothèques pandas (Pandas), et Numpy (Numpy)) et automatiser la concaténation sur la base des mappings faits dans les tables de mappings.

C. Vectorisation des données textuelles

La vectorisation des données dans les méthodes de traitement automatique du langage naturel (NLP) est un processus essentiel pour transformer des données textuelles en représentations numériques exploitables par les modèles d'apprentissage automatique. Les étapes réalisées automatiquement (après une phase d'expertise) avant la vectorisation des données sont la suppression des abréviations et des « stopswords », ainsi que la lemmatisation.

Les commentaires provenant des bases de données contenaient diverses abréviations. Ainsi, un même mot pourrait être représenté par plusieurs variables après l'étape de vectorisation. Tous les commentaires des clients et ceux des ateliers de réparation, ainsi que les solutions apportées à la panne, ont été étudiés : les abréviations récurrentes ont été regroupées dans un fichier Excel, ainsi que les mots complets associés. Enfin dans le script de prétraitement, ledit fichier Excel a été importé pour faire correspondre et remplacer les abréviations détectées. La présence des abréviations diminuait l'importance des mots qui pourraient être utiles à la classification par les algorithmes d'apprentissage automatique. Cette étape a permis de réduire la dimensionnalité (nombre de variables après vectorisation) de nos données, sans perte d'information.

Les « stopswords » ont été supprimés du corpus des commentaires. Ce sont des mots très courants qui n'apportent généralement pas beaucoup de valeur sémantique à un texte donné lors de l'analyse ou du traitement automatique du langage naturel (NLP). Ces mots sont souvent omis lors de la vectorisation des données textuelles car ils peuvent introduire du bruit et rendre les modèles moins performants. Ces « stopwords » comprennent généralement les articles, les prépositions, les conjonctions et d'autres mots très fréquents (par exemple l', c', m', est, ai …) qui sont utilisés pour former la structure grammaticale des phrases, mais qui ne portent pas le sens principal du texte.

Une lemmatisation des commentaires a été réalisée. La lemmatisation désigne un traitement lexical apporté à un texte et consiste à transformer les mots en leur forme canoniques. Tout comme les deux étapes précédentes, ceci a pour effet de réduire la dimensionnalité en préparation de la phase de vectorisation. La bibliothèque Spacy-lefff de Python (Spacy-lefff) a été utilisée pour faire cette lemmatisation. Pour donner un exemple de lemmatisation :

- Luisent, luisant ----------> luire
- Manger, mangeant, mangez, mangeons ----------> mange

Pour la vectorisation des commentaires (Tab.I), la méthode « TF-IDF » a été choisie. La fréquence de terme-inverse fréquence de document (TF-IDF) est une statistique numérique utilisée en traitement automatique du langage naturel pour évaluer

l'importance d'un mot dans un document par rapport à une collection de documents, souvent un corpus. Cette méthode « TF-IDF » est composé de deux composantes principales :

- La Fréquence de Terme (TF) : mesure à quelle fréquence un terme apparaît dans un document. Elle est calculée en divisant le nombre de fois qu'un terme apparaît dans un document par le nombre total de termes dans ce document. L'idée est que les mots qui apparaissent plus fréquemment dans un document sont plus importants ;

$$TF(t,d) = \frac{Nombre\ de\ fois\ que\ le\ terme\ t\ apparaît\ dans\ le\ document\ d}{Nombre\ total\ de\ termes\ dans\ le\ document\ d} \quad (1)$$

- L'Inverse Document Frequency (IDF) : mesure l'importance d'un terme dans l'ensemble du corpus en pénalisant les termes qui apparaissent dans de nombreux documents. Elle est calculée en divisant le nombre total de documents par le nombre de documents contenant le terme, puis en prenant le logarithme de ce quotient. L'idée est que les termes rares sont plus informatifs que les termes communs.

$$IDF(t,D) = \log\left(\frac{Nombre\ total\ de\ documents\ dans\ le\ corpus\ D}{Nombre\ de\ documents\ contenant\ le\ terme\ t}\right) \quad (2)$$

Une fois que les « TF » et « IDF » sont calculés, ils sont combinés pour obtenir le score « TF-IDF » pour chaque terme dans chaque document selon la formule :

$$TF - IDF(t,d,D) = TF(t,d) \times IDF(t,D) \quad (3)$$

Plus le score « TF-IDF » d'un terme dans un document est élevé, plus ce terme est important pour le document par rapport au reste du corpus. Celui-ci est couramment utilisé dans diverses tâches de NLP, telles que la recherche d'informations, la classification de texte et la modélisation de sujets, pour identifier les mots-clés, classer les documents et extraire les caractéristiques pertinentes.

D. Vectorisation des données en utilisant des modèles « Transformers » pré-entraînés

Les outils de traitement automatique du langage naturel (NLP) ont connu une évolution constante, passant des modèles de machine learning traditionnels aux « embeddings » de mots non contextuels, tels que « GloVe » (Pennington et al., 2014) et « Word2Vec » (Mikolov et al., 2013), et aux réseaux de neurones récurrents (RNN). Plus récemment, une avancée révolutionnaire dans ce domaine est survenue avec l'introduction des modèles basés sur les mécanismes d'attention, connus sous le nom de « Transformers ». Parmi ces modèles, « Camembert », une adaptation française du modèle de « BERT » pour Bidirectional Encoder Representation of Transformers (Devlin et al., 2018), se distingue en capturant le contexte des mots au sein des phrases, transformant ainsi les mots en vecteurs. Contrairement aux approches antérieures, telles que « TF » et « TF-IDF », les commentaires sont désormais représentés par des vecteurs de vecteurs de nombres.

Cependant, entraîner un modèle « Camembert » spécifique à un domaine nécessite une quantité de données considérable, ce qui n'était ici pas le cas dans l'exemple que nous avons considéré. Pour cette raison, des modèles pré-entraînés disponibles sur la plateforme « Hugging Face » (Hugging Face), qui offrent une variété de données textuelles, ont été utilisés. De plus, l'implémentation d'un modèle « Camembert » demande des ressources importantes en termes de mémoire vive (RAM). Pour surmonter cette contrainte, une version allégée de « Camembert », nommée « cmarkea/distilcamembert-base » (Delestre et al., 2022), a été choisie pour numériser les commentaires. Cependant, même cette version allégée nécessitait plus de ressources mémoire qu'un PC standard est en mesure d'en allouer. Pour résoudre ce problème, un traitement par lot, traitant 50 lignes à la fois, a été mis en place. En effet, nous souhaitons que nos outils, destinés à un ingénieur FMDS, puissent être utilisable avec un ordinateur portable standard.

E. Modélisation et classification automatique des commentaires

Le problème étudié concerne une classification en classe multiple. Il existe deux types de classification en classes multiples : la classification multi-classe, où chaque observation est assignée à une seule classe ; la classification multi-label, où chaque observation peut être attribuée à plusieurs classes simultanément (ces classes sont alors appelées labels). Avant de modéliser un problème de classification, il est important de déterminer le type de classification auquel nous sommes confrontés. Dans le cadre de cette étude, il s'agit d'une classification multi-label, car les observations dans les données peuvent appartenir à plusieurs verbatims d'une même famille de verbatim à la fois. En fait, certains commentaires peuvent contenir plusieurs types de défaillances et donc plusieurs verbatims peuvent y être associés.

Dans la littérature, différentes approches sont proposées pour traiter le problème de la classification multi-label. Parmi les plus couramment utilisées, on peut mentionner :

- L'approche "Binary Relevance" : Cette technique consiste à entraîner un modèle pour chaque classe, puis à agréger les prédictions de ces modèles pour obtenir la prédiction finale ;
- L'approche "Problem Transformation" : Cette technique implique de transformer le jeu de données en un problème de classification multi-classe, puis d'entraîner un seul modèle. Les prédictions obtenues sont ensuite transformées inversement pour retrouver la forme multi-label. Le choix entre ces approches dépend de la base de données. En général, si une corrélation significative existe entre les labels, l'approche "Problem Transformation" est préférée, sinon l'approche "Binary Relevance" donne de meilleurs résultats. Une métrique utilisée pour évaluer la corrélation entre les labels est le « SCUMBLE » (Francisco et al., 2019). Un « SCUMBLE » supérieur à 0,1 indique des classes ou labels fortement corrélés.

Les valeurs de « SCUMBLE » appliquées à notre jeu de données donnent les résultats suivants :

- « SCUMBLE_Effets_clients » = 0,056
- « SCUMBLE_Modes_defaillance » = 0,063
- « SCUMBLE_Solutions » = 0,067

Toutes ces valeurs sont inférieures à 0,1. Ces résultats nous ont conduits à choisir l'approche "Binary Relevance" plutôt que l'approche "Problem Transformation". Ainsi, lors de la phase d'entraînement des modèles, un modèle a été entraîné pour chaque label. Pour cela, la classe « multioutputclassifier » de « Scikit learn » (Scikit-Learn) a été utilisée.

F. Jeu de données déséquilibré

En plus de déterminer s'il s'agit d'un problème multi-label ou multi-classe, il est essentiel de vérifier l'équilibre des différents labels présents dans les données. Les données sont considérées comme équilibrées lorsque les labels ont approximativement le même nombre d'occurrences. En revanche, une grande disparité dans le nombre d'occurrences des labels indique des données déséquilibrées. La métrique utilisée pour évaluer l'équilibre ou le déséquilibre des données est le « MeanIR » (Herrera et al., 2015). Plus le « MeanIR » se rapproche de 1, plus les données sont équilibrées. En revanche, un « MeanIR » supérieur à 1 indique un déséquilibre des données. Une piste d'amélioration des modèles consiste à ré-échantillonner les données pour les équilibrer.

Pour résoudre le déséquilibre des labels (« MeanIR » à 330 sur les données d'effets clients), la méthode de « sur-échantillonnage aléatoire » (Ying, 2019) a été utilisée. La méthode de rééchantillonnage appelée "Random Over Sampling" (sur-échantillonnage aléatoire) est une technique utilisée pour équilibrer les classes dans un ensemble de données déséquilibré, en particulier dans le cadre de problèmes de classification. L'objectif est de générer des données synthétiques en sur-échantillonnant les classes minoritaires de manière aléatoire jusqu'à ce que toutes les classes soient équilibrées.

L'application de la méthode ML-ROS a permis de diminuer la valeur du « MeanIR » à 3 sur les données d'effets clients. Néanmoins, peu importe les techniques de rééchantillonnage utilisées (ML-ROS, ML-SMOTE, ML-RUS), les performances des modèles sont dégradées lorsqu'elles sont utilisées.

G. Modèles de machine learning utilisés

Les modèles employés pour la classification sont des modèles d'apprentissage automatique supervisés. Les données sont réparties entre les variables explicatives et les variables cibles (les labels), comme illustré dans le Tab.I : 80% des données sont utilisées pour entraîner les modèles, tandis que les 20% restants sont réservés à l'évaluation de l'efficacité de la classification après l'entraînement.

Trois modèles ont été sélectionnés pour la modélisation : les Machines à Vecteurs de Support ou SVM (Cortes et al., 1995), « Random Forest » (Breiman, 2001) et « XGBOOST » (XGBOOST). Ces choix ont été motivés par les avantages spécifiques de chaque algorithme :

- « SVM » : Ce modèle est réputé pour sa précision élevée, notamment dans les espaces de haute dimension. Dans le cadre de notre problème NLP, le texte après vectorisation est projeté dans un espace vectoriel de dimension élevé. Il est également résistant au sur-apprentissage ;
- « XGBOOST » : Comme « SVM », « XGBOOST » offre une précision élevée, notamment dans les espaces de haute dimension. De plus, « XGBOOST » propose diverses techniques de régularisation pour contrer le sur-apprentissage ;
- « Random Forest » : Cet algorithme a été sélectionné en raison de sa capacité à éviter le sur-apprentissage grâce à son mécanisme de vote.

Ces algorithmes ont été évalués sur les trois techniques de vectorisation différentes mentionnées ci-dessus (« TF », « TF/IDF » et « Camembert Transformer Embeddings ») afin de déterminer la combinaison la plus performante entre l'algorithme et la technique de vectorisation.

Pour évaluer les performances, le « F1-Score » (Powers, 2011) a été utilisé, représentant la moyenne harmonique entre la précision et le rappel.

Un fine-tuning sur les hyper-paramètres des modèles (Max_depth, learning_rate, n-estimator pour XGBOOST ; Random_state, max_iter, calss_weight, C, tol pour SVM ; N_estimator, max_depth,max_features_boostrap pour Random forect) utilisés a été effectué automatiquement. Le fine-tuning des hyperparamètres consiste à ajuster minutieusement les paramètres d'un modèle de Machine Learning pour obtenir les meilleures performances possibles sur un ensemble de données spécifique. Ce processus est souvent itératif et implique généralement l'utilisation de techniques, telles que la recherche par grille (Claesen et al., 2015) pour explorer l'espace des hyperparamètres et trouver la combinaison optimale. L'objectif du fine-tuning des hyperparamètres est d'optimiser les performances du modèle en trouvant la meilleure configuration possible, ce qui peut se traduire par une meilleure précision, un temps d'entraînement plus court ou une meilleure généralisation aux données de test. Cela peut également aider à éviter le surapprentissage en ajustant les paramètres de régularisation.

IV. RESULTATS

Le Tab.III regroupe les « F1-Score » obtenus par les différents modèles de machine learning et les différentes méthodes de vectorisation qui ont été testés, associés aux trois variables cibles (Tab.I). La durée d'apprentissage avec nos données était inférieure à 2 heures sur un ordinateur portable standard par modèle.

TABLE III. « F1- SCORE » POUR LES MODELES DE MACHINE LEARNING ET DE VECTORISATION TESTES

	Effet(s) client			**Mode(s) de défaillance**			**Solution(s) appliquée(s)**		
	SVM	XGB	RF	SVM	XGB	RF	SVM	XGB	RF
TF	0,744	**0,768**	0,611	0,841	**0,843**	0,785	**0.920**	0,908	0,795
TF/IDF	0,689	**0,751**	0,615	0,824	**0,834**	0,774	0.878	**0,895**	0,793
Camembert	**0,737**	0,638	0,602	**0,840**	0,813	0,747	**0.881**	0,878	0,763

V. DISCUSSION ET PERSPECTIVES

L'analyse des résultats obtenus au Tab.III permet d'en déduire les modèles les plus adaptés au traitement de chacune des variables : effet(s) client, modes de défaillance et solution(s).

La combinaison « XGBOOST » / « TF » produit de meilleurs résultats pour les verbatims "Effet(s) client". De plus, le modèle « XGBOOST » se distingue comme le plus performant avec les techniques de vectorisation « TF » et « TF/IDF ». En revanche, pour la technique de vectorisation basée sur le « transformeur Camembert », c'est le modèle SVM qui obtient le meilleur résultat. Quelle que soit la technique de vectorisation utilisée, le modèle « Random Forest » se révèle être le moins performant parmi les trois modèles. Cependant, tous les scores obtenus jusqu'à présent pour les "Effet(s) clients" ne sont pas satisfaisants et nécessiteront des améliorations. Les verbatims "Effet(s) client" sont principalement basés sur les commentaires des clients, où l'on trouve un langage informel, des abréviations et différents styles d'écriture. En raison de ces spécificités, les techniques de vectorisation « TF » et « TF/IDF » pourraient ne pas être efficaces à long terme, surtout lorsque la classification doit être effectuée sur un volume de données beaucoup plus important. En revanche, les modèles « transformeurs » sont capables de comprendre le contexte des phrases et pourraient s'avérer mieux adapté.

De manière similaire aux verbatims "Effet(s) client", le modèle « XGBOOST » affiche une performance supérieure avec les techniques de vectorisation « TF » et « TF/IDF » pour les verbatims « Mode(s) de défaillance ». En revanche, pour la technique de vectorisation basée sur le « transformeur Camembert », c'est encore une fois le modèle SVM qui obtient le meilleur résultat. Le modèle « Random Forest » se révèle encore être le moins performant parmi les trois modèles.

Pour les verbatims « Solution(s) appliquée(s) », le modèle SVM affiche une performance supérieure avec les techniques de vectorisation « TF » et le « transformeur Camenbert ». Quant au modèle « XGBOOST », il affiche la meilleure performance pour la technique de vectorisation « TF/IDF » sur ces verbatims « Solution(s) appliquée(s).

Les variables « commentaires atelier » et « solutions apportées » utilisés pour la vectorisation étaient plus riches en termes techniques et plus courts. Ainsi, les techniques de vectorisation « TF » et « TF/IDF » pourraient être viables à long terme. Par conséquent, pour les verbatims "Mode(s) de défaillance", la combinaison « XGBOOST » / « TF » obtient le meilleur score. Le modèle « Random Forest » se révèle encore être le moins performant parmi les trois modèles.

L'application à la base de données considérée d'un modèle d'IA et vectorisation sélectionnés permet l'attribution de chaque verbatim client à un mode de défaillance donné. Cette classification est un élément préliminaire essentiel à la réalisation d'une analyse de fiabilité opérationnelle : l'identification des paramètres d'une loi statistique de Weibull (en temporel ou en kilométrique) sur des données de retour d'expérience en clientèle, à partir d'une méthode de maximum de vraisemblance par exemple, doit être réalisé sur un ensemble de donnée homogène à un unique mode de défaillance. Cette identification des deux

ou trois paramètres d'une loi de Weibull correspondant à un mode de défaillance unique peut être aisément réalisée en automatique, à partir du moment où les données clients sont classées. En général, la partie la plus fastidieuse du travail d'un ingénieur fiabiliste, qui doit réaliser une analyse de fiabilité opérationnelle, correspond au nettoyage de la base de données considéré et à l'attribution de chaque retour client multilinguistique à un mode de défaillance. L'intérêt des outils d'IA que nous proposons devient alors évident.

VI. CONCLUSION

En conclusion, cette étude démontre clairement l'efficacité et la pertinence des méthodes d'intelligence artificielle, en particulier en ce qui concerne le traitement automatique du langage, pour l'analyse des retours d'expérience en clientèle correspondants aux défaillances d'un système. En développant un outil d'IA capable d'attribuer automatiquement un mode de défaillance à chaque retour client, il a été possible de réduire considérablement le temps nécessaire au traitement de ces données volumineuses tout en minimisant les erreurs humaines associées à cette tâche complexe.

L'utilisation de modèles de Machine Learning, notamment le modèle « XGBOOST » avec une méthode de vectorisation « TF » s'est avérée particulièrement efficace dans ce contexte, permettant une analyse multilingue des retours clients et des diagnostics d'ateliers. De plus, l'application de techniques, telles que le Fine-Tuning a permis d'adapter ces modèles à nos données spécifiques, même en l'absence d'un volume de données massif.

En déployant cette application de classification automatique des retours clients, la réalisation d'analyses de fiabilité opérationnelle (par exemple, via des analyses de Weibull sur les incidents attribués automatiquement par un modèle « XGBOOST » avec une méthode de vectorisation « TF » à chacun des modes de défaillance du système. Ces analyses de Weibull ne faisant pas l'objet du présent article), a été facilitée, offrant ainsi aux ingénieurs FMDS des résultats précis et rapidement disponibles pour la bonne compréhension des modes de défaillance en clientèle et de leur origine, ainsi que l'amélioration continue des systèmes et des composants. Ces travaux, axés sur une base de données synthétique de l'industrie automobile, illustrent la valeur ajoutée des approches d'IA dans la gestion des retours d'expérience client et ouvrent la voie à de futures applications dans divers secteurs industriels. En définitive, l'intégration de l'IA dans les processus d'analyse des retours clients représente un véritable atout pour les entreprises cherchant à améliorer la fiabilité de leurs produits et à répondre efficacement aux besoins des consommateurs.

REMERCIEMENTS

Les auteurs tiennent à remercier Dr Tonia-Maria ALAM et M. Walter SONGO pour leurs contributions conséquentes à ces travaux, ainsi que Lilia BEKDA.

Ces travaux ont été financés à partir des budgets R&D internes à SOM LIGERON.

BIBLIOGRAPHIE

Breiman, (2001). Random Forests. *Machine Learning* 45, 5–32.

Chi Sun et al., (2019). How to Fine-Tune BERT for Text Classification*, Chinese Computational Linguistics, Lecture Notes in Computer Science*, *Springer*, volume 11856, p.194-204**.**

Chin Chen et al., (2011). Quality evaluation of product reviews using an information quality framework, *Decision Support Systems,* volume 50, Issue 4, Pages 755-768.

Claesen et al., (2015). "Hyperparameter search in machine learning." *arXiv preprint arXiv:1502.02127* .

Cortes et al., (1995). Support-Vector Networks. *Machine Learning* 20, 273–297.

Cyrile et al., (2022). une distillation du modèle français CamemBERT. *CAp (Conférence sur l'Apprentissage automatique)*, Vannes, France. ⟨hal-03674695⟩

Deepa et al., (2023). Sentimental analysis recognition in customer review using novel-CNN, *2023 International Conference on Computer Communication and Informatics, ICCCI, IEEE*, pp. 1-4.

Deepl Translator, https://www.deepl.com/fr/translator?referrer=https%3A%2F%2Fwww.google.com%2F

Delestre et al., (2022). DistilCamemBERT : une distillation du modèle français CamemBERT. *CAp (Conférence sur l'Apprentissage automatique)*, Vannes, France.

Devlin et al., (2018). "Bert: Pre-training of deep bidirectional transformers for language understanding." *arXiv preprint arXiv:1810.04805*.

Francisco et al., (2019). Dealing with difficult minority labels in imbalanced mutilabel data sets, Neurocomputing, Volumes 326–327, Pages 39-53.

Ganesan et al. (2023). Deep learning approaches for accurate sentiment analysis of online consumer feedback, *2023 International Conference on Computer Communication and Informatics, ICCCI, IEEE,* pp. 1-5.

Herrera et al., (2015). Addressing Imbalance in multi-label classification: Measures and random resampling algorithms. Neurocomputing, Volume 163.

https://huggingface.co/

Hu Y et al., (2022). Short-Text Classification Detector: A Bert-Based Mental Approach*, Computational Intelligence and Neuroscience,* volume 2022, published online.

Hugging Face, https://huggingface.co/

Martin et al., (2019). "CamemBERT: a tasty French language model." *arXiv preprint arXiv:1911.03894*.

Mikolov et al., (2013). "Efficient estimation of word representations in vector space." *arXiv preprint arXiv:1301.3781*.

Min et al. (2012). Identifying helpful reviews based on customer's mentions about experiences, *Expert Syst Appl,* volume 39, pp. 11830-11838.

Numpy, https://numpy.org/

Pandas, https://pandas.pydata.org/

Pennington et al., (2014). "Glove: Global vectors for word representation." *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*.

Powers, (2011). "Evaluation: From Precision, Recall and F-Measure to ROC, Informedness, Markedness & Correlation". *Journal of Machine Learning Technologies*. 2 (1): 37–63.

Ramaswamy et al. (2018). Customer Perception Analysis Using Deep Learning and NLP, *Procedia Computer Science, volume 140*, p. 170-178.

Scikit-Learn, Scikit Learn, https://scikit-learn.org/stable/

Selenium, https://selenium-python.readthedocs.io/

Spacy-lefff, https://pypi.org/project/spacy-lefff/

XGBOOST https://github.com/dmlc/xgboost

Ying, (2019). An Overview of Overfitting and its Solutions. Journal of Physics: Conference Series. 1168. 022022. 10.1088/1742-6596/1168/2/022022.