Ecole nationale de la statistique
et de l'analyse de l'information

Projet statistique de 2ème année

Etudiant 2A

Analyse d'avis client

Équipe :
Joslin Martinez
Nicolas Bourgain
Ayoub Ben Chaliah
Hela Dellagi

Tuteur :
Emmanuel Haback
Responsable :
Youssef ESSTAFA

2020 - 2021

Table des matières

1 Introduction **1**

2 Problématiques organisation **2**
2.1 Prendre du recul sur les données brutes . . . 2
2.1.1 Trustpilot.com . . . 2
2.1.2 Réflexions sur les données brutes . . . 3
2.2 La création d'un dashboard pertinent . . . 4
2.2.1 Qu'est-ce qu'un dashboard ? . . . 4
2.2.2 Quels indicateurs ? . . . 4
2.3 Organisation du travail . . . 5
2.3.1 Organisation du groupe . . . 5
2.3.2 Diagramme de Gantt . . . 6

3 Les données **7**
3.1 Extraction des données . . . 7
3.2 Les variables . . . 8
3.3 Nettoyage des avis clients . . . 8

4 Les analyses **9**
4.1 Présentation, traitement du langage naturel . . . 9
4.2 Utilisation de dictionnaires . . . 10
4.2.1 Prédiction du sentiment positif, neutre ou négatif d'un avis . . . 10
4.2.2 Les dictionnaires sont peu appropriés à la détection de sujets . . . 11
4.3 Apprentissage automatique . . . 11
4.3.1 Modèle LDA . . . 12
4.4 Sentiment de l'avis selon le topic . . . 13
4.5 Sentiment de l'avis selon le fournisseur . . . 13
4.6 Continuité de l'analyse et temporalité . . . 14
4.7 Deep learning . . . 14
4.7.1 Introduction aux réseaux neuronaux artificiels . . . 15
4.7.2 Présentation du modèle BERT . . . 17
4.7.3 Intégration de poids entraînées . . . 17
4.7.4 Intégration en fonction de nos données . . . 17
4.7.5 échantillon représentatif . . . 17

5 L'intégration dans un dashboard interactif **19**
5.1 Introduction . . . 19
5.2 Présentation de l'API . . . 19
5.3 Présentation du dashboard . . . 20

6 Conclusion **24**

7 Annexes **25**
7.1 Bibliographie 25

Table des figures

2.1 Exemple d'avis sur Trustpilot.com . . . 3
2.2 Diagramme de Gantt du groupe . . . 6

4.1 Modèle comportant 1 couche de 2 neurones artificiel simple . . . 15
4.2 Modèle comportant 2 couche de 8 et 2 neurones artificiel simple . . . 16

5.1 Vue d'ensemble du dashboard . . . 20
5.2 Relancement du processus . . . 20
5.3 Choix de l'année . . . 20
5.4 Formats de téléchargement des figures . . . 20
5.5 Focalisation sur un seul fournisseur . . . 21
5.6 Elimination des fournisseurs sélectionnés du graphe . . . 21
5.7 Graphe de Favorabilité basée sur le nombre d'étoiles . . . 21
5.8 Graphe de nombre de critique par sujet : . . . 22
5.9 Graphe de proportion des critiques (avec focalisation sur Engie) . . . 22
5.10 Hystogramme Positif/négatif par sujet (avec Total Direct Energie éliminé) . . . 22
5.11 Graphiques en radar . . . 23
5.12 Filtres appliqués sur les données . . . 23

Résumé

Posting reviews about goods and services on specialized website is quite common on the internet nowadays. It is therefore a tremendous source of data about the customers' opinions and experiences with companies. Therefore, naturally many companies would ask consulting firms to gather and analyse these data.

For our 2nd year statistics' project at ENSAI, Lead Data Scientist Emmanuel Haback at TNP Consulting assigned ourselves with the task of collecting and analysing the reviews on Trustpilot.com that relate to France's energy providers. The resulted dataset, then must be used in order to make a dashboard for benchmarking and drawing conclusions about the main customers' concerns.

At first, we discussed the difficulties that we may encounter while dealing with data that is programmatically collected via scrapping and making a dashboard that will best highlight the most useful extracted information to the companies. The second step was choosing the set of tools and frameworks that will best serve the tasks in question and provide us with a viable and elegant dashboard that is fully automated and as accurate as possible. After setting a clear plan of how the project should progress, the first technical step after scrapping was cleaning the dataset by removing punctuations, links, emojis ... in order to remove nuance from the data that will negatively impact the precision of the future models that will be using it.

At this point, we used different corpora from the library NLTK and deep learning model (BERT) for sentiment analysis of the reviews. We also used the LDA model for topic-modeling, which will make the dashboard automatically determine the main topics discussed.

Afterwards, we provided the dashboard with the resulted dataset that included the outputs of the aforementioned models. Since the dashboard was made using JavaScript, the resulted data were fed to the dashboard via an API made using a flask.

Chapitre 1

Introduction

Dès les années 2000, la banalisation d'internet a profondément changé les habitudes de nos sociétés modernes. Il n'est en effet plus rare aujourd'hui d'acheter des biens et services en ligne. Mais Il ne faut pas oublier l'aspect participatif et communautaire sur le web : échange de contenus divers, participation à une encyclopédie, partage des idées sur des forum de discussion . . . Parmi ces activités il y en a une qui fait le lien entre les entreprises et les consommateurs : les sites de recueil de notes et d'avis.

Ces derniers sont devenus de plus en plus populaires car ils permettent aux clients d'orienter leurs choix d'achat dans une démarche que Franck Cochoy appelle le « calqul ». Par ce néologisme l'auteur explique que les internautes copient leurs décisions d'achats sur les autres. Ils ont en effet davantage confiance dans leurs pairs que dans la publicité traditionnelle. L'information sur les sites d'avis y est perçue comme objective et vérifier les avis avant achat est vite devenu un réflexe pour beaucoup de consommateurs. On estime ainsi que plus de 2000 milliards d'avis ont circulé en France en 2016.

Les entreprises ne sont pas en reste car les sites d'avis et de notations sont aussi un formidable outil d'analyse marketing : satisfaction client, perception de la marque, benchmarking, qualité perçue du B2C . . . Néanmoins ces pages web sont aussi une arme à double tranchant qui peuvent réduire à néant la réputation d'une entreprise. Les commentaires négatifs ou les faux avis déposés par un concurrent ou un employé revanchard doivent ainsi être pris très sérieusement. Analyser ces sites n'est donc pas une activité anodine et constitue au contraire un enjeu majeur pour les entreprises au XXI siècles.

Dans le cadre de notre projet statistique de deuxième année, il nous a été demandé d'analyser et de comparer les avis des fournisseurs d'énergie publiés sur le site trustpilot.com. Le fruit de notre travail sera ensuite présenté sous la forme d'un dashboard.

Mais comment peut-on extraire les avis sur ces sites et les convertir en une base de données utilisable ? Comment déterminer le sujet principal d'un avis et savoir s'il est positif, négatif ou neutre ? Comment présenter de manière efficace et pertinente nos résultats ?

Afin de répondre à ces questions, le rapport sera organisé en 4 parties. La première partie exposera les différentes problématiques soulevées par le sujet. Puis nous parlerons de la collecte des données avec la méthode que nous avons mis en oeuvre pour rassembler les informations. Le traitement de ces données afin de tirer des analyses sera ensuite abordé dans la troisième partie. La présentation de notre dashboard ferra l'objet de la dernière partie avant de conclure notre rapport.

Chapitre 2

Problématiques organisation

Le sujet de notre étude consiste dans la réalisation d'une application permettant de visualiser et d'analyser des avis sur les fournisseurs d'énergie publiés sur le site trustpilot.com. Pour cela il faut passer par plusieurs étapes. Tout d'abord il faut réaliser un scraping des avis pour se constituer une base de données. La quantité importante d'information va ensuite nécessiter de mettre en place différentes méthodes pour analyser les avis. Enfin les informations devront être présentées à travers un dashboard.

Si cela semble trivial à première vu, il n'en est rien pour plusieurs raisons. D'une part rien ne nous laisse entendre que les données seront exploitables directement à travers un simple scraping. D'autre part la réalisation des analyses et du dashboard mérite réflexion pour que l'application soit utile aux futurs utilisateurs. Enfin dans le cadre d'un travail de groupe, il est aussi important de bien s'organiser pour pouvoir répondre au mieux à la demande de notre tuteur.

2.1 Prendre du recul sur les données brutes

Avant d'exploiter les données brutes, il faut réfléchir sur certaines choses. La provenance des données n'est en effet pas anodine. Trustpilot.com, de par son fonctionnement, est sujet à certaines limites dont il faut avoir conscience. Ensuite les avis en eux-mêmes sont rarements exploitables directement et demande aussi une précaution supplémentaire.

2.1.1 Trustpilot.com

Créé en 2007, Trustpilot.com est un site web recueillant les avis des internautes sur des entreprises partout dans le monde. Tout le monde peut s'inscrire gratuitement sur le site et poster des avis à l'aide d'un mail ou d'un compte Facebook. On estime ainsi qu'un million d'avis sont déposés chaque mois sur le site Trustpilot.com en 2015.

Le site n'est évidemment pas excempt de défauts et de critiques. Ainsi on accuse le site de propager de faux avis malgré la politique de vérification mis en avant par les gestionnaires du site d'avis. La facilité d'inscription sur le site permet en outre de créer de faux comptes et donc de contourner les analyses de trustpilot et d'influencer les résultats. Même si le service est gratuit, les entreprises peuvent payer le site pour accéder à diverses fonctionnalitées et surtout avoir une influence sur les avis. Les firmes peuvent ainsi demander à TrustPilot de n'afficher que certains avis selon des critères qu'on imagine favorable. Elles peuvent aussi soliciter l'aide du site pour les aider à améliorer leur score.

Enfin il paraît nécessaire de rappeler qu'en terme d'avis, les consommateurs sont nettements plus enclins à commenter leurs expériences lorsqu'elles sont négatives que lorsqu'elles sont positives. Outre

FIGURE 2.1 – Exemple d'avis sur Trustpilot.com

le besoin d'exprimer sa frustration, donner un avis négatif permet aussi d'attirer l'attention des entreprises afin d'obtenir un traitement de faveur ou de débloquer une situation personnelle avec le SAV. Le site Trustpilot permet en effet aux entreprises d'envoyer des messages aux utilisateurs suite à leurs avis. Il ne faudra donc pas s'étonner de voir une proportion extrêmement importante d'avis négatifs.

En définitif bien que Trustpilot soit une source pratique pour obtenir une base de données d'avis il ne faut pas oublier certaines des limites énoncées avant de réaliser des analyses et de des conclusions.

2.1.2 Réflexions sur les données brutes

Comme le montre la figure 2.1 chaque avis est présenté d'une manière standardisée. Il y a tout d'abord le nom de l'utilisateur, le nombre d'avis qu'il a posté ainsi que le lieu de résidence. Puis vient ensuite la note sur cinq étoiles suivie du nombre de jours où l'avis a été mis en ligne. Après une courte phrase faisant office de titre il y a enfin l'avis en lui même qui fait plusieurs lignes. Pour terminer le site affiche la date d'envoi de l'avis. Les autres utilisateurs ont ensuite l'option d'approuver l'avis et ils peuvent partager l'avis sur les réseaux sociaux.

A la vue de ces informations plusieurs réflexions viennent à l'esprit. Tout d'abord on pourrait imaginer que certaines informations pourraient rentrer en contradiction. Par exemple les notes est un sujet hautement subjectif et un texte noté 3 étoiles pourrait en valoir 4 ou 2 selon la personne.

Ensuite il y a des problèmes liés au corps de l'avis. La richesse de la langue française fait qu'il y a un nombre conséquent de mots à analyser. Il y a aussi la ponctuation et l'utilisation de smiley qui complexifient les opérations de traitement. Mais le plus problématique reste l'utilisation de l'ironie ou du sarcasme dans la rédaction de l'avis. Il est en effet difficile de pouvoir reconnaître et traiter automatiquement un commentaire en apparence positif mais dont le message est tout sauf des louanges.

L'analyse des avis n'est donc pas une tâche aisée. Il faut avoir en tête les limites d'un site comme TrustPilot ainsi que les difficultés liés à la langue qu'un traitement automatique ne pourrait pas forcément détecter. Il faut donc se préparer et sans doute se résoudre à avoir recours à certaines concessions dans le traitement de l'information pour pouvoir l'analyser.

2.2 La création d'un dashboard pertinent

La finalité du projet est de créer un Dashboard destiné au management pour l'analyse des avis clients et la comparaison de leur service avec celui de la concurrence. Après avoir défini ce qu'est un dashboard, nous discuterons des points importants à prendre en compte pour en créer un.

2.2.1 Qu'est-ce qu'un dashboard ?

Le dashboard, plus connu sous le nom de tableau de bord, est défini par Henri Bouquin comme :

"Un ensemble d'indicateurs peu nombreux (cinq à dix) conçus pour permettre aux gestionnaires de prendre connaissance de l'état et de l'évolution des systèmes qu'ils pilotent et d'identifier les tendances qui les influenceront sur un horizon cohérent avec leurs fonctions"

Un indicateur est une donnée qui représente l'état ou l'évolution d'un système. Dans le cadre de notre projet, cela pourrait être l'opinion d'un utilisateur à propos d'un fournisseur d'énergie selon une échelle à déterminer : une note sur dix, un classement positif/négatif/neutre ... Cela signifie donc que le dashboard n'est pas unique et doit être au contraire adapté à la situation de l'entreprise et aux buts qu'elle recherche. Les informations fournies par le tableau devront aussi aider le manager à prendre des décisions. Par exemple si les avis sont très négatifs et rapportent souvent des problèmes liés au SAV alors le management choisira sans doute d'améliorer les prestations dudit service par différents moyens : formation des employés, prime de performance, recrutement, restructuration du service ...

Par ailleurs il faut différencer le tableau de bord du tableau de reporting. le premier vise à accompagner une stratégie d'entreprise tandis que le second vise à agréger des informations à un instant t pour pouvoir effectuer des comparaisons avec les concurents. Dans le cadre de notre projet, il nous semble que nous devons réaliser un tableau mixte. Nous allons en effet remonter des informations à un instant t comme un tableau de reporting. Mais il y a aussi l'introduction d'une échelle temporelle dans la demande du projet qui impose de faire des comparaisons à différentes périodes. On pourrait alors dire que l'entreprise cherche à connaître les évolutions des avis afin de mieux connaître les leviers pour s'améliorer. Notons pour la suite du rapport que nous parlerons de dashboard ou de tableau de bord afin de ne pas désorienter le lecteur et par commodité.

2.2.2 Quels indicateurs ?

Après avoir parlé du tableau de bord. Nous allons abordé les indicateurs qui consituent le coeur de celui-ci. Dans son livre "Mesurer et piloter la performance" Nicolas Berland énumère les caractéristiques d'un bon indicateurs :

- Objectif : il ne doit pas dépendre de l'interprétation du manager ou de l'instrument de mesure.
- Quantifiable : Il faut pouvoir disposer d'une mesure absolue ou relative de ce qu'on cherche à analyser.
- Fidèle : L'indicateur doit toujours donner la même réponse lorsque les situations mesurées sont strictement identiques. Il faut donc une certaine rigueur lors la création d'un indicateur.
- Simple et compréhensible : Il faut que l'information rapporté soit comprise et facile à interpréter pour l'utilisateur. Il faut donc trouver un juste milieux entre pas assez et trop d'informations.
- Sensible : Il doit pouvoir réagir rapidement aux évènements qu'il souhaite mesurer. Il faut néanmoins faire attention à chercher à mesurer en temps réel notamment pour des indicateurs qui

ne prennent du sens que sur le long terme.

Dans le cadre de notre projet il nous est demandé de pouvoir connaître l'avis des clients et de comparer les indicateurs entre les différents fournisseurs d'énergie (Engie, Direct Energie, Edf ...) sur différentes périodes. Il faut donc pouvoir créer des indicateurs :

- qui mettent en évidence les différents type d'avis avec par exemple une échelle positif/négatif/-neutre.
- qui expliquent quel thème ressort le plus souvent des avis pour mieux comprendre ce qui est reproché ou apprécié.
- qui présentent des valeurs numériques sur les avis (ratio, nombres ...) pour en tirer une information rapidement
- qui incluent une échelle temporelle pour comparer sur des durées
- qui peuvent se mettre rapidement à jour et faire des comparaisons entre fournisseurs d'énergie

En outre comme pour toute application, il faut que le tableau de bord puisse être agréable à l'oeil et à l'utilisation. L'apparence de l'interface n'est pas anodine et permet au contraire de renforcer son utilité. Un tableau de bord difficile d'utilisation et extrêmement basique ne donnerait pas envie d'être utilisé, même si son contenu est de qualité.

2.3 Organisation du travail

Notre projet s'est déroulé du 18 Janvier au 25 Avril. Afin de concilier la charge de travail habituelle de l'Ensai (cours, examens ...) avec le projet, nous nous sommes organisés. Nous avons tout d'abord partagé les tâches dans les groupes et réalisé un diagramme de Gant pour avoir une représentation du temps et des tâches à effectuer.

2.3.1 Organisation du groupe

Notre groupe s'est organisé selon les spécialités et les affinités de chacun afin d'être le plus efficace possible.

- Joslin : nettoyage des données, analyse statistiques, rédaction du rapport
- Nicolas : rédaction du rapport et du rapport intermédiaire, répartition des tâches via le diagramme de Gantt, aide à la création d'une API qui restructure les données et connecte le tableau de bord à l'ensemble de données
- Ayoub : Extraction des données, analyse des sentiments, programmation et supervision de la programmation
- Hela : Réalisation du Dashboard, rédaction du rapport

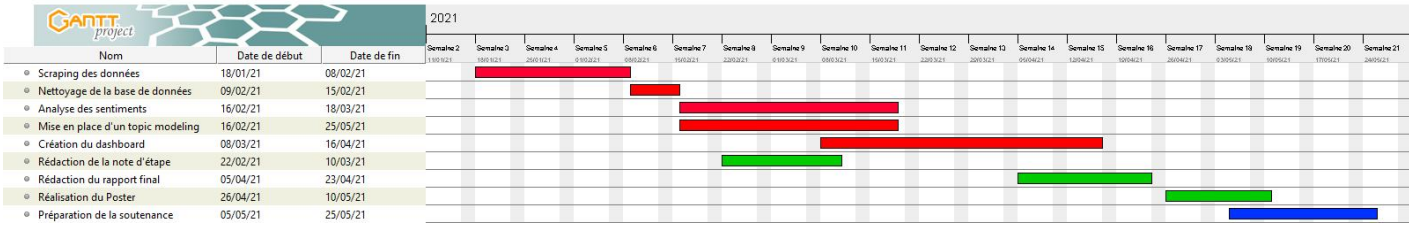

FIGURE 2.2 – Diagramme de Gantt du groupe

2.3.2 Diagramme de Gantt

Le planning que nous avons prévu via un diagramme de Gantt réparti notre travail en plusieurs étapes :

Le rassemblement des avis ainsi que leur nettoyage a été fait sur un mois. Un autre mois a été nécessaire pour l'analyse des sentiments. Puis la mise en place d'un topic modeling a duré 2 semaines. En parallèle, le dashboard a nécessité un peu plus de mois pour être finalisé. La première ébauche n'ayant pas convaincu notre tuteur, nous avons du le retravailler.

En plus des tâches prévu par le diagramme de Gantt, des réunions ont été réalisé sur Teams. Ces dernières ont pour but de faire le point sur l'état d'avancement du projet et d'échanger sur nos parties respectives. Si l'avancement n'est pas significative nous échangions rapidement sur l'application Messenger. Les conditions sanitaires actuelles n'ont pas permis de nous rencontrer directement à l'école et cela a pu avoir un impact sur la qualité des échanges.

Chapitre 3

Les données

3.1 Extraction des données

Le scraping est une technique permettant d'extraire des données provenant d'un ou plusieurs sites web, de façon automatique. Les avis clients que nous devons analyser doivent être extraits de pages sur le site Trustpilot. Nous devons réaliser un scraping de l'ensemble des avis clients concernant les fournisseurs suivant : "edf.fr", "engie.fr" et "total.direct-energie.com".

Afin d'effectuer cette extraction de données, nous avons dans un premier temps utilisé la librairie Python requests pour extraire le script HTML des différentes pages web. Ensuite il a été nécessaire d'interpréter ce code HTML, afin d'obtenir in fine un document Excel contenant l'ensemble des données analysables dans le cadre du projet. Pour cela, nous utiliserons l'interpréteur syntaxique BeautifulSoup du package bs4, permettant de parser (c'est à dire analyser et vérifier la syntaxe d'un élément) le code HTML en Python. Cela va nous permettre d'intégrer les données à analyser dans un dictionnaire, afin d'organiser chaque colonne avec les données correspondantes. Les données que nous avons considérées comme importantes sont :

- "reviewId", l'ID de l'avis
- "consumerId", l'ID du client
- "consumerName" , le nom du client
- "socialShareUrl", l'adresse https de l'avis concerné
- "businessUnitDisplayName", le nom du fournisseur
- "reviewHeader", le titre de l'avis
- "reviewBody", le corps de l'avis
- "stars", les étoiles de l'avis, allant de 1 à 5

Un fichier .xls peut alors être extrait simplement. Celui-ci sera utilisé dans le cadre de l'analyse des avis client.

Ce script a été optimisé pour être modifiable et avoir les moyens d'intégrer facilement de nouveaux fournisseurs d'énergie. Il peut donc être relancé automatiquement lors d'un refresh de l'analyse de données afin de prendre en compte les nouveaux avis déposés par les clients, dans un but de continuité de l'application.

3.2 Les variables

Nous considérerons comme variables d'intérêts le fournisseur, le corps de l'avis, la date de l'avis ainsi que les étoiles de l'avis. Les corps des avis constituera les X_i que nous allons devoir traiter et analyser. Les fournisseurs et la temporalité seront principalement utilisés à des fins de segmentation :

- Segmenter par fournisseur permettra de faire apparaître les similarités et les différences entre fournisseurs.
- Segmenter par période permettra d'identifier l'évolution dans le temps des phénomènes étudiés concernant la satisfaction client.

Par ailleurs, les étoiles seront rattachés au dashboard, uniquement à des fins de segmentation. Le fait de ne pas les intégrer directement à l'analyse des avis sera discuté dans la suite de ce rapport. Pour éviter les biais, nous préfererons utiliser des prédictions basées sur le titre et le corps de l'avis quant aux sentiments qu'ils représentent, plutôt qu'utiliser les étoiles. Nous posons donc Y_i la classification de sentiments associée à l'avis X_i.

3.3 Nettoyage des avis clients

L'analyse du langage nécessite un travail préalable quant à la structure des phrases. En effet, la phrase "C'est dommage, ce fournisseur est médiocre" est bien moins facile à analyser pour un modèle d'apprentissage que la phrase "dommage fournisseur médiocre". Il est absolument nécessaire de "nettoyer" les avis clients afin de les rendre utilisable.

La ponctuation sera un des premiers éléments à supprimer des avis. Pour cela, nous avons utilisé les expressions régulières appelé aussi regex. Ce sont des chaînes de caractères, qui décrivent, selon une syntaxe précise, un ensemble de chaînes de caractères possibles *(source : wikipedia)*. Cela nous a aussi permis de supprimer les apostrophes et les émoticones de la base de données. Par ailleurs, tous les mots des avis seront mis en minuscule.

La deuxième étape importante est de définir des stopwords, c'est à dire des mots que l'on ne veut pas que le modèle considère comme important. Cela se présente sous la forme de dictionnaire où sont répertoriés, dans la langue considérée, les mots qui n'ont pas vocations à servir dans l'analyse. On peut imaginer un début de dictionnaire tel que : stopwords=["à", "a","donc","est" ...].
En complément, nous avons utilisé un *lemmatizer* pour considérer un nombre de stopwords bien plus conséquent. Celui-ci permet de retrouver une forme lexicale basique du mot en question. Par exemple le lemme du nom "chevaux" est "cheval". Cela marche aussi avec les structures verbales : si "est" est contenu dans les stopwords, alors "étaient" sera supprimer de tous les avis. On pourrait dire que ces modifications constituent une perte d'informations sur les avis, ce qui est sans aucun doute vrai. Cependant, nous ne pourrions pas extraire correctement les sujets, si une multitude de mots avec peu d'importance viennent interférer avec l'analyse.

La troisième étape est de transformer l'avis client, en une liste des mots restant dans l'avis. Ainsi, on peut regarder l'exemple suivant : les données stockées pour l'avis ["dommage fournisseur médiocre"] deviennent ["dommage","fournisseur","médiocre"]. Cette architecture est obtenu grâce à un *tokeniser* permettant de séparer les mots selon les espaces. Nous obtenons donc une liste de mots nettoyés, par avis clients.

La base de données est maintenant organisé comme un ensemble d'avis X, chaque avis X_i étant composé d'une liste des mots importants w qu'il contient dans le corps. S'ajoute à cela l'ensemble des autres variables d'intérêts défini précédemment.

Chapitre 4

Les analyses

4.1 Présentation, traitement du langage naturel

Le langage naturel est caractérisé par le langage utilisé entre les êtres humains, en opposition avec les langages "formel" comme les langages informatiques ou les mathématiques. Le traitement du langage naturel (ou *Natural language processing*(NLP)) correspond à l'ensemble des méthodes pour traiter et analyser le langage naturel, le but étant souvent d'en extraire les informations importantes. Les champs d'application sont multiples. On peut réaliser des classifications de documents, analyser le sentiment global d'un texte, la reconnaissance de l'écriture manuscrite ou encore faire de la fouille de texte (ou *Text mining*)

Cette section consiste à explorer les différents outils que nous avons mis en oeuvre dans la suite, pour analyser le langage naturel, dans le cadre de la modélisation de la satisfaction clients. Notre DataSet, maintenant composé de données nettoyées, peut être analysé avec des outils NLP.

On peut se poser beaucoup de questions :
Comment analyser des avis et par quels moyens ?
Quels sont les clefs pour produire une analyse de sentiment clients ?
Comment comprendre les avis clients, et en extraire des informations ?
Quelles serait les informations nécessaires à des professionnels de l'énergie quant à la connaissance de leurs client ?

Le plus pertinent serait en premier lieu de se demander si les avis sont plutôt positif, neutre ou bien négatif. En effet, on peut s'interroger sur le sentiment que transmet un avis. Dans ce but on pourrait tout simplement recourir au système de notation par étoile. Mais nous perdrions une quantité d'information importante quant aux sentiments des clients. Avoir un outil permettant de transcrire les informations contenu dans l'avis en une prédiction positif, neutre ou négatif, semble indispensable afin d'avoir une prédiction du sentiment général de l'avis. Pour détecter le sentiment positif, neutre ou négatif, nous utiliserons une sélection d'outils parmi lesquels Vader, un analyseur de sentiment de la librairie NLTK. Nous effectuerons aussi des vérifications quant à la qualité de ses prédictions.

Avoir une représentation catégorielle du sentiment de l'avis est un point de départ à l'analyse des avis clients. Cependant, il n'est pas très intéressant de savoir qu'un commentaire est positif ou négatif, si l'on a aucune idée de ce dont il parle. Avoir une vision globale des sujets les plus discutés est sans doute un vecteur d'amélioration important de la connaissance client. Pour cela, nous avons choisi d'extraire les sujets les plus importants directement à partir des avis, plutôt que d'utiliser des dictionnaires, puisque ceux-ci nous enferment dans certains sujets, sans en détecter de nouveaux. Ces sujets seront

donc extrait à partir d'un modèle génératif probabiliste non supervisé, à allocation latente : le modèle *Latent Dirichlet allocation* (LDA). Ce modèle bayésien est pertinent car il prend des documents en entrée, et trouve des sujets en sortie.

Ces deux composantes distinctes vont nous permettre de comprendre les sentiments qu'ont les clients à l'égard de chaque fournisseur, mais aussi de classifier les avis par sujet, et d'analyser les sentiments en fonction de chaque sujet. Ce principe d'extraction de sujet est particulièrement intéressant lorsque l'on considère une application ayant vocation à durer dans le temps, et être intégrée dans un dashboard interactif. En effet les sujets seront regénérés à chaque refresh du scraping, offrant la possibilité de prendre en compte la continuité de la collecte des avis. Par exemple, imaginons que les clients se plaignent d'un nouveau phénomène qui était jusqu'à lors inconnu du fournisseur, si ils sont assez nombreux à en parler, alors il y a de grande chance que le modèle LDA détecte le nouveau sujet, là où cela aurait été impossible avec des dictionnaires. Nous souhaiterons aussi offrir la possiblité à l'utilisateur de consulter de manière intégrée, les avis classifiés se rapportant à un sujet, en fonction du sujet choisi.

L'analyse des sentiments étant un instrument central de l'étude de la satisfaction client, nous tenterons d'améliorer nos résultats de détermination du ressentiment client par de l'apprentissage profond (ou *Deep learning*). En effet, nous utiliserons un algoritmhe d'apprentissage profond très récent, le modèle Bidirectional Encoder Representations from Transformers(BERT, 2018),un modèle de GoogleIA couplant différents types de réseaux de neurones. Les calculs pour l'entrainement du modèle étant extrèmement important et coûteux, nous avons utilisé un modèle pré-entrainé.

4.2 Utilisation de dictionnaires

4.2.1 Prédiction du sentiment positif, neutre ou négatif d'un avis

Comme nous l'avons vu, il semble nécessaire d'analyser les sentiments qu'un client laisse apparaître dans son avis. Pour cela, nous aurions pu utiliser les étoiles associées aux avis. Cependant, il est souhaitable d'utiliser au maximum les avis écrits par les clients pour capturer un maximum d'informations. Aussi, nous nous sommes rendu compte de plusieurs biais concernant ces mêmes étoiles. En effet, certains clients oublient de noter l'avis lors du dépot, ce qui implique que des avis positifs peuvent se retrouver avec une étoile.

De même, on trouvera des clients qui postent des commentaires négatifs sur des points considérés comme nécessitant une amélioration, mais qui laisse au moins 4 étoiles car il sont globalement satisfait des services du fournisseur. Il faut bien comprendre que dans notre cas, ce qui importe est le sentiment global de ce qui est écrit dans le corps de l'avis, car nous serons amené à utiliser ces prédictions de sentiment au regard des sujets discutés dans l'avis même. En outre, il constituera un bon indicateur global de l'état de la satisfaction client.

Nous utiliserons alors un lexique appelé Vader, de la librairie NLTK. C'est un lexique entraîné qui détecte la positivité, la neutralité et la négativité. Le lexique de Vader regroupe une très grande variété de mots associés à une valeur entre -4 et 4. Cette valeur représente l'intensité du caractère positif ou négatif du mot. Le programme fonctionne ainsi : Vader associe une distribution de probabilité à un avis X_i, en fonction des mots qu'il contient, tel que :

$$P(X_i \text{ est négatif}) + P(X_i \text{ est neutre}) + P(X_i \text{ est positif}) = 1$$

Un score entre -1 et 1 est alors associé à l'avis. A des fins d'analyse, ce score sera plus lisible que la distribution de probabilité. Nous pourrons nous permettre de segmenter les avis neutres ayant un score compris entre $-\alpha$ et α, en choisissant la valeur de α. Nous considérerons un α plutôt faible, afin

de capter un faible mécontentement, ou un avis légèrement positif.

Le lexique vader utilise une API pour traduire le lexique dans quelques autres langues, dont le français. Nous apliquerons ce lexique français.

Par la suite, nous tenterons d'améliorer les performances de la détection de la positivité ou la négativité d'un avis, par des méthodes d'apprentissage profond.

4.2.2 Les dictionnaires sont peu appropriés à la détection de sujets

Vient maintenant une question qui a nourri mûres réflexions. Comment savoir quels avis parlent de tel ou tel sujet ? Utiliser des dictionnaires pour classifier les avis par sujet nécessiterait d'associer au sujet choisi une multitude de mots en rapport avec le sujet. Cela semble possible, mais de nombreuses questions sans réponses convenables apparaissent.

D'abord, quels sujets choisir ? "service client", "relevé compteur", "panne" ? Choisirions nous plutôt le sujet "réseaux","internet" ou "wifi" ? Ou bien les trois, sachant qu'aucun ne se recoupe parfaitement, pourquoi pas après tout ? L'utilisation d'un sujet est-elle pertinente par rapport à l'ensemble des avis, ou un autre sujet pourrait-il être mieux adapté ? Qui va choisir quel sujets sont discutés dans les avis ? On constate par ces questionnements que faire intervenir une intelligence humaine pour établir les sujets discutés par des milliers de clients, est fortement biaisé.

Utiliser des études reflétant les principaux sujets discutés par les clients des fournisseurs d'énergies pourrait être une solution. Mais nous n'avons pas accès à ces études. Nous devons maintenant associer à ces sujets, un ensemble de mots. Les mêmes questions reviennent alors. Quels mots associer à tel ou tel sujet ? Qui décide de ces mots ? Le mot "service" ne viendrait-il pas qualifier le sujet "service client" ? Mais cela signifie qu'un avis traitant des services réseaux se retrouverait classé comme traitant du service client ? Nous sommes donc en face d'un dilemme. Soit on n'associe pas le mot "service" et on perd une information importante. Soit on associe "service" à tous les sujets auxquels ils se rapportent avec les risques qu'on connaît. Doit-on associer une intensité reflétant la degré d'appartenance d'un mot à un sujet, à la manière de Vader ? et comment ?

On voit bien que l'attribution manuelle des mots et des sujets auxquels ils s'y rapportent est extrêmement aléatoire et dépend principalement des appréciations faites lors de la création du dictionnaire. L'ensemble de ces contraintes nous conduira à abandonner l'idée d'utiliser des dictionnaires dans le cadre de la segmentation par sujet. Nous utiliserons donc un modèle d'apprentissage, pour déduire les sujets directement des avis clients. En effet, n'est-ce pas les avis eux-mêmes qui sont les plus appropriés pour définir les sujets à considérer ?

4.3 Apprentissage automatique

L'apprentissage automatique (ou *Machine learning*), est un champ de l'intelligence artificielle basé sur des principes statistiques et mathématiques permettant de donner à la machine la capacité d'apprendre à partir de données. C'est possible car la machine va tenter d'améliorer ses performances à chaque étape de l'apprentissage.

Il existe une distinction importante à faire entre apprentissage supervisé, qui examine un set d'apprentissage labelisé (c'est à dire avec des sorties Y connues représentant des étiquettes) et apprentissage non supervisé ou il n'y a aucune labelisation. Pour la détection des sujets importants, nous aurions pu utiliser des méthodes d'apprentissage supervisé, mais cela aurait nécessité de labeliser à la main un set d'apprentissage d'avis en fonction de catégorie de sujet auxquels ils appartiendraient. Cette tâche de

labelisation aurait été fortement biaisé, pour toutes les discussions évoquées précédemment. Comment labeliser les avis est une question qui reste encore en suspens.

Ces questions n'ayant pas de réponses exactes dans le cadre d'une labelisation manuelle, nous utiliserons l'apprentissage non supervisé dans le cadre de la détection des sujets.

4.3.1 Modèle LDA

Comme nous l'avons évoqué précédement, nous allons utiliser le modèle *Latent Dirichlet Allocation* (LDA) pour extraire les principaux sujets d'un point de vue probabiliste. C'est un modèle d'apprentissage non supervisé ayant pour principale utilisation la détection de la thématique d'un document, dans un corpus de documents. Lors de l'utilisation, on fixe K sujets et on cherche à inférer les thèmes que les documents représentent. Chaque mot de chaque document est alors associé à un thème.

Le modèle

Nous pouvons décrire le modèle de façon plus précise.

Soit M le nombre de documents, N_d le nombre de mots dans chaque document d et N le nombre total de mots dans tous les documents. Les observations considérées sont les mots w collectés au sein du corpus de texte X. Le modèle suppose que chaque document $X_{d=1,...,M}$ est un mélange probabiliste θ d'un petit nombre de sujets. Il suppose par ailleurs que la génération de chaque occurence d'un mot w est attribuable en terme probabiliste à un sujet t. Les mots w sont les seules variables observables, les autres étant des variables latentes. Cela signifie que ces variables latentes seront inférées directement d'autres variables. Ainsi, ce que le modèle considérera comme sujet t seront les mots w ayant la plus grande probabilité d'être un thème selon les termes du modèle.

On initialise ensuite le modèle en choisissant le vecteur de paramètres α de la priore sur les distributions de sujets t par documents d avec $\alpha_{k=1,..,K} < 1$, pour préférer les distributions de sujets éparses, c'est-à-dire avec peu de sujets par document. Nous obtenons le vecteur de paramètres $\theta_{d=1,...,M}$ représentant la distribution des sujets dans chaque document d. Il suit une loi de Dirichlet $\theta_{d=1,...,M} \sim D(\alpha)$.

Par ailleurs, le vecteur de paramètres β de la priore sur les distributions des mots w par sujet t avec $\beta_{d=1,..,M} << 1$, pour préférer les distributions de mots éparses, c'est-à-dire avec peu de mots par sujet. Nous obtenons le vecteur de paramètres $\phi_{k=1,...,K}$ représentant la distribution des mots dans chaque sujet t. Il suit aussi une loi de Dirichlet : $\phi_{k=1,...,K} \sim D(\beta)$.

L'apprentissage consiste à améliorer le modèle et se rapprocher des distributions recherchées. Celui-ci passe par la mise à jour du sujet associé à chaque mot en considérant que c'est le sujet qui aurait la plus forte probabilité de le générer dans ce document (on onsidère ici la probabilité $P(t|d)P(w|t)$, correspondant à la probabilité que le sujet t génère le mot w dans le document d). Lorsque que l'on répète cette étape un grand nombre de fois, les allocations tendent à se stabiliser.

Le modèle étant relativement complexe, nous souhaitions donner l'intuition des bases du modèle, et de son fonctionnement global, sans rentrer dans des détails trop complexes. Ce modèle reste reconnu et efficace dans le cadre où nous l'utilisons. In fine, nous obtenons les distributions de l'ensemble des thèmes, des probabilités de mots qui leur sont associées et du mélange particulier de thèmes de chaque document.

Implémentation

Nous allons implémenter ce modèle grâce à sklearn (*sklearn.decomposition.LatentDirichletAllocation*). Nous utiliserons le même corpus d'avis nettoyés X que pour la détection de sentiments. Afin d'effectuer une détection de sujet efficace, nous avons choisi d'utiliser $K = 10$ afin de trouver les 10 sujets les plus important dans le cadre du modèle. Nous avons décidé d'utiliser deux mots consécutifs pour qualifier un sujet. Cela permet d'améliorer la pertinence des sujets extraits, et fournir des sujets plus précis. Le fonctionnement décrit si dessus change très peu du fait de cette modification, les sujets étant seuelement considérés comme deux mots consécutifs.

Sélection des sujets et stopword

Les sujets étant générés par le contenu des avis eux-mêmes, ils pourront parfois avoir des titres peu intéressant, ou à analyser de plus près. Tous les sujets détectés ne seront pas forcément pertinent. Parfois, il sera utile d'ajouter certains mots que l'on ne souhaite pas en tant que stopword, afin que ces mots ne soient plus pris en compte lorsque l'on relance le modèle. Nous pourrons alors déduire d'autres sujets, parfois plus pertinent. Cependant, plus l'on ajoute des sujets aux stopwords, plus les nouveaux sujets ont une probabilité moindre d'être pertinents. Par la suite, nous souhaitons offrir à l'utilisateur la possibilité de consulter directement les avis associé à tel ou tel sujet, de façon à pouvoir juger et vérifier la pertinence du sujet, et ce à quoi il se rapporte.

Nuage de mots

Le modèle LDA peut être utilisé à d'autres fins : nous pouvons l'appliquer sur le corpus d'avis X, en ayant pour but de détecter un maximum de sujet dans l'ensemble des avis. Cela peut nous permettra d'extraire un nuage de mots, ordonnant les sujets détectés à partir de l'ensemble du corpus.

4.4 Sentiment de l'avis selon le topic

Nous disposons maintenant de deux composantes distinctes pour l'analyse. D'une part, nous avons pu extraire de façon générale ce qui est discuté dans les avis clients. D'autres part nous pouvons extraire les sentiments globaux qui se rapporte à un avis. Cela nous autorise à appliquer une analyse plus poussée, en liant ces deux caractéristiques. En effet, il est maintenant possible de segmenter les avis par sujet, et d'observer les sentiments que renvoient les avis en fonction d'un sujet particulier. A titre d'exemple, nous pourrions dire : dans tous les avis, 26% traitent du sujet "Service client" et parmi eux, 32% sont positifs, 12% sont neutres et 56% sont négatifs. Ces analyses peuvent donc nous permettre de juger de la qualité de certaines caractéristiques.

4.5 Sentiment de l'avis selon le fournisseur

Dans ce projet, nous analysons les avis déposé sur Trustpilot, pour trois fournisseurs d'énergie distincts : EDF, Engie et Total-direct-énergie. Nous pouvons donc appliquer toutes les analyses vues précédement, puis les segmenter par fournisseurs. Cela nous permettra d'avoir une vision globale pour chacun des trois fournisseurs.

En outre, nous avons remarqué que les clients discutent souvent des fournisseurs d'énergie concurrents. En effet, des sujets comme "Direct Energie" sont souvent extrait des avis clients. Lorsque l'on segmente par founisseur, si l'on regarde uniquement les avis déposé pour Total direct énergie, cela peut permettre d'exprimer les sentiment globaux envers le fournisseur d'énergie du client. Si l'on considère les avis déposé pour les deux autres fournisseur, cela peut signifier de liens entre fournisseurs. Nous devons tout de même faire attention quant à l'analyse de ces liens ils sont assez incertains : le sentiment positif ou négatif qui se dégagerait de l'avis ne peut être facilement associé à un des deux fournisseurs

présent dans l'avis. Ce que l'on sait, c'est qu'il parle du concurent. Dans ce cas là, il peut être nécessaire d'aller voir directement les avis concernés, afin de conmprendre la nature des liens entre les fournisseurs (par exemple si beaucoup de clients veulent changer pour un autre fournisseur, où si les clients sont plutôt déçu ou satisfait de leur nouveau fournisseur, par rapport à l'ancien)

4.6 Continuité de l'analyse et temporalité

Notre base de données contient aussi les dates auxquelles ont été saisie les avis. Cela nous permet d'appliquer les analyses vues précédement, puis segmenter les avis par période. A titre d'exemple, nous pourrons observer la nature des sentiments concernant le "service client" pour EDF, sur la période du mois de décembre 2019. Cette segmentation offre à l'analyste la possibilité de réaliser une multitude de graphiques, en fonction de ce qui doit être étudié.

Par ailleurs, cela nous offre la possibilité de relancer le modèle LDA sur des périodes données, afin d'extraire les sujets les plus en vogue selon la période. Aussi, nous pourrons intégrer les nouveaux avis à l'apprentissage automatique des sujets, et pourquoi pas considérer seulement les nouveaux avis segmenté pour détecter les sujets important à l'instant présent.

4.7 Deep learning

Dans cette section, nous allons tenter d'améliorer la performance de notre détection de sentiments. Nous nous étions basés sur Vader, un outil d'analyse de sentiments basé sur un lexique. Comme nous le savons, cette méthode est puissante, mais le fonctionnement est assez limité. Il existe aujourd'hui des outils considérablement plus efficaces pour comprendre le langage naturel. Le modèle que nous avons utilisé et qui sera présenté ici est un modèle de deep learning complexe de GoogleIA, datant de 2018. Il se nomme *Bidirectional Encoder Representations from Transformers* (BERT).

Ce genre de modèle comporte une quantité de paramètres assez exeptionnelle : près de 110 millions de paramètres. Cela permet alors une compréhension réelle du langage, en traitant les mots dans le contexte du langage plutôt qu'en traitant les mots un à un, sans lien entre eux. Nous avons choisi ce modèle car c'est aujourd'hui l'un des plus puissant pour comprendre le langage. Il est en effet utilisé pour les tâches les plus complexes dans le cadre du NLP, comme la traduction.

Il a de nombreuses applications, dont l'analyse des sentiments, les tâches de "Question Answering" et la reconnaissance des entités nommées (NER). Ce modèle est reconnu et a amélioré considérablement les outils d'analyse du langage. Il aurait probablement pu aussi être utilisé pour l'extraction des sujets importants provenant des avis clients, mais la tâche s'est avéré trop complexe pour que nous puissions la mettre en place. Etant extrêmement complexe, nous ferons une description du modèle dans ses généralités, pour donner l'intuition de pourquoi le modèle est efficace, puis nous verrons les méthodes d'implémentation.

Ce modèle est un modèle contextuel bidirectionnel transformers, composé de 12 couches de 768 neurones et de 12 têtes d'attention. L'entraînement d'un modèle comme celui-ci nécessite une base de données énorme et est bien trop coûteux dans le cadre de ce projet, c'est pourquoi nous laissons la tâche d'entrainement de côté, et nous utilisons un modèle "pretrained" en lui ajoutant des couches supplémentaires durant l'étape de fine-tuning. Nous utiliserons des poids basés sur un apprentissage global fourni initialement par Hugging Face (modèle CamemBERT), que nous compléterons avec des données labelisées de la base de données d'avis de allociné (en français) pour l'étape de *fine-tuning*. Cette étape permet de compléter l'apprentissage avec des données qui correspondent au domaine du langage étudié. L'apprentissage étant basé sur deux catégories, positif et négatif, nous ne détecterons donc pas les avis neutres. Cela contribue aussi à une précision supérieure.

4.7.1 Introduction aux réseaux neuronaux artificiels

Cette première partie consiste à présenter une architecture simple d'un réseau de neurones artificiels, dans le cadre d'une classification de données. Considérons une variable X avec deux composantes $X1$ et $X2$. Une loi régit la valeur de Y (0 ou 1) en fonction des valeurs de X. On va chercher à apprendre une méthode de détermination de la valeur de Y à partir d'une nouvelle donnée X compte-tenu d'un apprentissage sur des données labelisées (X, Y). Voyons ici un problème simple de classification.

Voyons maintenant comment nous pouvons résoudre ce problème grâce à un réseau neuronal. Un neurone artificiel, inspiré du fonctionnement du neurone biologique, prend en entrée des valeurs X et associe une sortie Y. Une fonction d'activation détermine si le neurone va s'activer, ce qui correspond à une sortie égale à 1 pour le neurone en question. Si l'on utilise ici deux neurones distincts, nous pouvons optimiser les poids associés à chaque entrée et sortie des neurones en maximisant la fonction de perte associée à l'apprentissage. A chaque étape de l'apprentissage, la fonction de perte est optimisée par des moyens d'approximation (souvent l'algorythme de descente de gradient dans des cas simple) en fonction du taux d'apprentissage défini par l'utilisateur. On peut voir un premier exemple sur la figure ci dessous :

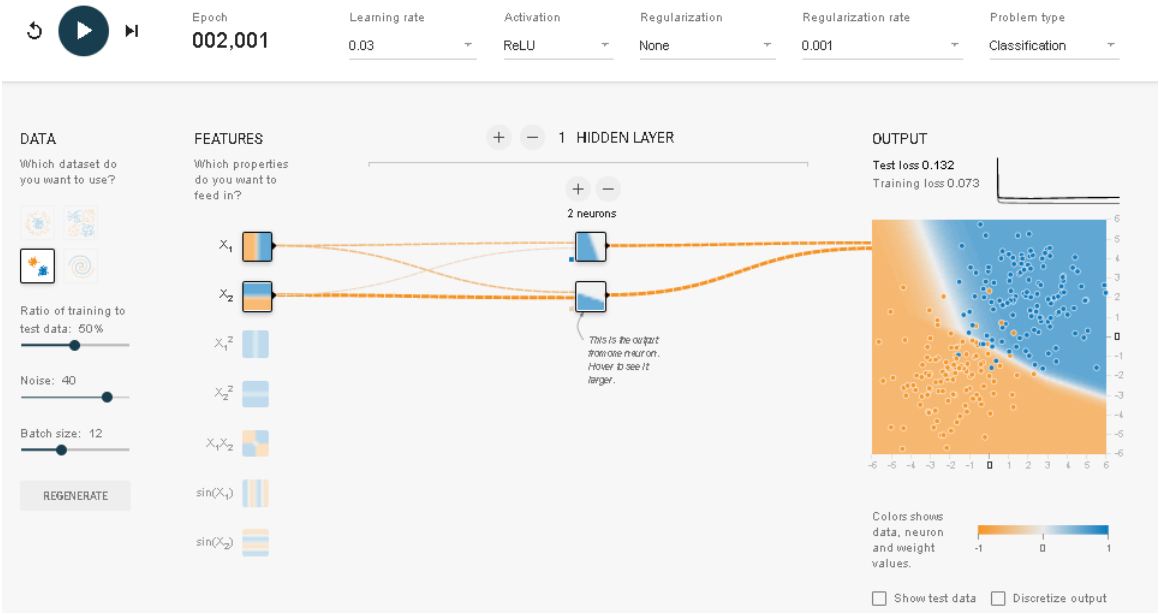

FIGURE 4.1 – Modèle comportant 1 couche de 2 neurones artificiel simple

Les zones bleus et orange sur le graphique correspondent aux sorties offertes par le modèles de neurones en fonction de nouvelles données X. Les points bleu et orange sont les données d'apprentissage labelisées. Comme on peut le voir sur cette simulation, les 2001 époques ont permi l'apprentisage de poids par le modèle, pour coller au mieux aux données d'apprentissages présentes. En effet, chacun des neurones a maintenant un poids précis affecté à ses entrées et ses sorties. On voit que dans le cadre de cette distribution de données, un seul neurone aurait pu suffire. A la manière d'une régression linéaire, une neurone aurait fourni une droite de régression. Sur les petits graphiques fournis au centre des deux neurones, nous pouvons voir qu'ils se complètent et fournissant chacun une droite de régression différente, qui comme nous l'avons vu, est associé à des poids. Nous pouvons alors mieux nous rapprocher des données d'apprentissage.

Pour comprendre la puissance de ces réseaux neuronaux, nous allons voir une seconde simulation,

dans le même cadre, mais avec une loi sur les données (X, Y) différente.

Voyons maintenant l'intérêt d'utiliser les neurones en couche. Une architecture en couche permet de prendre les sorties des neurones de la première couche en tant qu'entrée des neurones de la seconde. Cela ajoute des paramètres au modèle, et il est possible de résoudre des problèmes plus complexes.

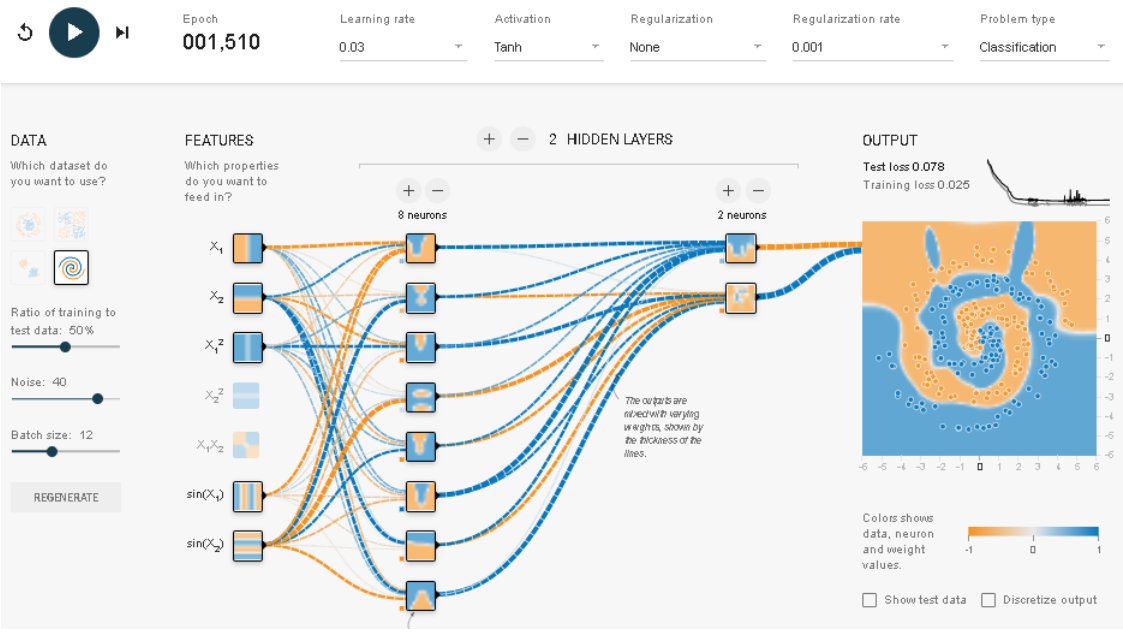

FIGURE 4.2 – Modèle comportant 2 couche de 8 et 2 neurones artificiel simple

Dans la simulation ci-dessus, les données ont été généré par une loi plus complexe que le précédent modèle de neurones qui n'était pas capable de comprendre. Nous avons toujours deux neurones avant la sortie, mais cette fois-ci, ils prennent comme entrée les sorties de la couche neuronale précédente, composée de 8 neurones artificielles. Ici, nous avons aussi effectué des transformations des variables en entrée, pour que les neurones de la première couche puissent capter le maximum de caractéristiques des données. Dans la pratique, ces structures sont extrêmement efficaces car chacun des neurones va se spécialiser, au cours de l'apprentissage, dans la détection d'un patern particulier.
Ce que nous avons vu est la base d'une architecture neuronale simple. Il est alors possible de multiplier les neurones et les couches de neurones pour obtenir des modèles efficaces dans un certain cadre analytique. De nombreuses innovations ont aussi vu le jour concernant la structure même des neurones. On peut penser à des réseaux de neurones récurrents (*Recurrent Neural Networks RNN*) qui permettent d'établir des connexions entre les neurones, pas seulement dans un sens, mais aussi dans les deux sens y compris des couches profondes à celles du dessus. Ce mécanisme permet alors la prise en compte du contexte : dans le cadre de la compréhension du langage par exemple, un mot sera étudié en fonction des mots qu'il y a autour. Pour reprendre un exemple utilisé plus tôt, le mot "service" dans "service client" sera analysé en prenant en compte qu'après le mot "service" se trouve le mot client.

Les améliorations apportées aux méthodes de construction d'un réseaux de neurones sont très nombreuses et répondent à des problématiques différentes. Nous aurions pu voir les réseaux de neurones à convolution (*Convolutional Neural Network CNN*) particulièrement utilisé dans le cadre de classification d'images, ou encore les cellules *Long Short Term Memory LSTM* visant à améliorer grandement les possibilités consernant les architectures RNN. Dans la suite nous apporterons quelques précisions sur les innovations qu'utilise le modèle BERT.

4.7.2 Présentation du modèle BERT

BERT est une amélioration significative quant aux méthodes utilisées pour la compréhension du langage. Il tend à remplacer les RNN et LSTM dans le cadre du NLP. Il est basé sur le modèle Transformer présenté en 2017 qui est un modèle qui apprend les relations contextuelles entre les mots dans un texte donné. Les modèles Transormer lisent le texte en une seule fois et non pas séquentiellement comme les modèles directionnels (d'ou l'appellation bidirectionnel, même si non-directionnel serait plus fidèle à cette description).

Un Transformer inclut un encodeur qui lit l'entrée et un décodeur qui donne la prédiction. Le modèle BERT n'utilise qu'un encodeur puisqu'il est destiné à générer un modèle de langage. La meilleure façon de le décrire est une pile d'encodeurs où l'output de l'un est l'input de l'autre. BERT utilise deux stratégies de training, *Masked ML* MLM et *Next sentence prediction* (NSP)

Dans la première stratégie, une partie des mots est remplacée par un masque avant d'être présentée comme input du modèle. Le modèle tente alors de prédire ces mots en se basant sur les mots non masqués.

Dans la seconde, le modèle prend des paires de phrases dans le processus de training et prédit si la seconde vient après la première dans le document ou non. La moitié de l'entrée est en paires et l'autre est aléatoire à partir du corpus. Bert assure une rapidité de fine-tuning, ce qui aidera probablement à trouver nombreuses applications dans le futur.

4.7.3 Intégration de poids entraînées

Ce modèle n'est pas entraînable à notre niveau compte-tenu de la puissance de calcul nécessaire. Nous avons donc procédé autrement. Nous utiliserons des poids basés sur un apprentissage global en français fourni initialement par Hugging Face (modèle *CamemBERT*). Les poids fournis par ce modèle ont été appris à l'aide d'une base de données de 138 GB de textes en français. Nous avons donc pu importer ces poids dans le modèle.

4.7.4 Intégration en fonction de nos données

Afin d'optimiser notre modèle dans le cadre particulier de l'analyse de sentiments d'avis clients, il est nécessaire d'ajouter des données spécifique pour compléter l'apprentissage. Afin d'avoir une base de données d'apprentissage conséquente et labelisée, nous avons utilisé les avis provenant d'un autre site français où se recoupe des avis : Allociné. En effet, ce choix peut sembler pertinent car la critique de film et plus généralement l'expression de sentiment envers les films ressemble en beaucoup de points à la critique d'une entreprise. En effet, les avis décrivent ce qui va ou ne va pas dans le film, ainsi le modèle pourra apprendre les architectures de langages qui critiquent ou honorent le film, ou dans notre cas, le fournisseur d'énergie. Nous avons aussi choisi cette base pour des questions pratiques car d'une part, nous voulions utiliser un set conséquent d'apprentissage ce qui n'était pas possible avec les données de Trustpilot et d'autres part car nous voulions des données labelisées. D'autres part, les avis étant reliés à des étoiles allant de 1 à 5, ces avis peuvent être labelisé de façon assez convenable : on supprime tous les avis neutres (c'est à dire ayant 3 étoiles) et on considère comme négatif les avis avec 1 ou 2 étoiles et positif ceux avec 4 ou 5 étoiles. Cette suppression est obligatoire afin d'avoir une précision plus pertinente par rapport à la labellisation.

4.7.5 échantillon représentatif

Afin d'évaluer la performance du modèle, nous avons sélectionné un échantillon représentatif des données parmi les avis clients, pour comparer la classification effectuée par le modèle et celle que ferait un être humain. Nous chercherons alors à déterminer la proportion de faux négatif (évalué négatif

mais positif) et de faux positif (évalué positif mais négatif). Nous procéderons par un échantillonnage aléatoire simple.
Soit :
- N, la population constituée de 2669 avis client
- Z, le niveau de confiance associé à ((1-alpha)/2)
Nous prendrons un seuil de confiance de 95% par rapport à une loi normale
- M, la marge d'erreur associée à la taille de l'intervalle
Nous nous autoriserons une marge d'erreur égale à 8 %.
- P et 1-P les proportions d'avoir un avis négatif ou positif. N'ayant pas d'apriori compte tenu de l'absence de labels explicites sur nos données, nous choisirons p = 0.5
La formule pour calculer la taille minimale de l'échantillon nécessaire pour obtenir les caractéristiques ci-dessus est :

$$n = (z^2 * p(1-p))/m^2$$

On obtient $n = 150$. On tire aléatoirement 150 avis de la base de données et on les labelise manuellement on affectant des 0 et des 1 aux avis. On peut alors comparer avec les prédictions du modèle. Nous avons trouvé les résultats suivants :
Taux de vrais positifs : 96%
Taux de vrais négatifs : 96%

Cela nous rassure quant à la pertinence de l'utilisation de ce modèle.

Chapitre 5

L'intégration dans un dashboard interactif

5.1 Introduction

L'intégration des résultats obtenus dans un dashboard interactif permet de mieux visualiser et comparer les résultats obtenus pour chaque fournisseur d'énergie. Nous avons choisi **VueJS** comme framework pour des raisons à la fois fonctionnelles et esthétiques. VueJS est un framework JavaScript pour les applications monopages et les interfaces utilisateur.

5.2 Présentation de l'API

Pour pouvoir restructurer les données et connecter le tableau de bord à l'ensemble de données, nous avons creé une API en utilisant le microframework Flask.
Le décorateur @route() indique à Flask vers quelle fonction il faut diriger les requets.
L'endpoint /reset déclenche la fonction reset_func() qui relance tout le code et remplace les anciens résultats par les nouveaux.

```
@app.route('/reset', methods=["GET"])
def reset_func():
    subprocess.check_output("python3 inferBERT.py && mv new_analysed_energy.xls analysed_energy.xls", shell=True)
    df = pd.read_excel('analysed_energy.xls')
    return 'OK', 200
```

L'endpoint /reviews déclenche la fonction reviews_table() qui récupère les avis à afficher dans une table du dashboard et /portal déclenche la fonction basic_func() qui restructure les résultats pour qu'ils soient intégrés ultérieurement dans tous les graphiques.
Le retour de toutes ces fonctions sera intégré dans le dashboard via JavaScript.

5.3 Présentation du dashboard

FIGURE 5.1 – Vue d'ensemble du dashboard

Les résultats obtenus sont importés dans le script VueJS et utilisés pour générer tous les graphiques. Si l'utilisateur du dashboard veut voir les graphiques représentant les données actuelles, nous avons ajouté un bouton **Deep Learning** dans le menu qui déclenche l'ensemble du processus (passage par le scrapping, le nettoyage puis le script de machine learning) et renvoie les nouveaux résultats. Cela prend un certain temps, il faut donc patienter un peu pour avoir les nouveaux graphiques.

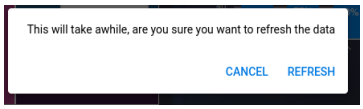

FIGURE 5.2 – Relancement du processus

Vous Pouvez choisir l'année de publication des critiques sinon l'année actuelle sera choisie par défaut. Vous pouvez également télécharger chaque graphique dans l'un de ces formats : SVG, PNG et CSV.

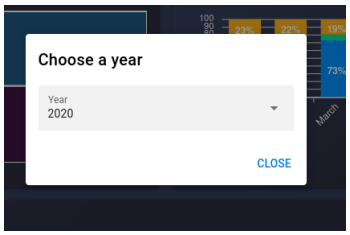

FIGURE 5.3 – Choix de l'année

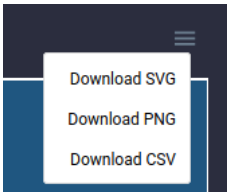

FIGURE 5.4 – Formats de téléchargement des figures

— **Favorabilité (basé sur les critiques) :**

L'axe des abscisses représente les mois de l'année que vous avez choisi et l'axe des ordonnées représente le pourcentage du ratio positif/négatif.

Les graphiques des trois fournisseurs d'énergie sont affichés mais vous pouvez vous focaliser sur le

graphique d'un fournisseur particulier en passant la souris sur son icône comme nous pouvons le voir dans ce graphique, vous pouvez également faire disparaître un ou plusieurs fournisseurs d'énergie du graphique en cliquant sur leurs icons.

FIGURE 5.5 – Focalisation sur un seul fournisseur

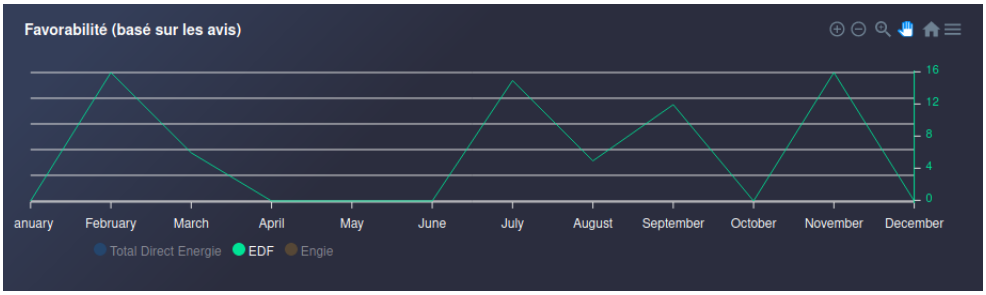

FIGURE 5.6 – Elimination des fournisseurs sélectionnés du graphe

Chaque fournisseur d'énergie a sa propre échelle sur l'axe des ordonnées. Plusieurs outils supplémentaires sont également disponibles : vous pouvez zoomer et dézoomer (par rapport au milieu du graphique) avec les icônes **-** et + ou zoomer dans une zone de votre choix que vous sélectionnez dans le graphique avec l'outil **loupe**. Vous pouvez également translater les graphiques avec l'outil **main** et revenir à la forme initiale du graphique en cliquant sur le boutton **home**.

— **Favorabilité (basé sur le nombre d'étoiles) :**

Contrairement au graphique précédent qui utilise des données estimées pour l'axe des ordonnées, celui-ci utilise simplement les données réelles. Il présente les mêmes caractéristiques que le précédent, mais pour le nombre d'étoiles.

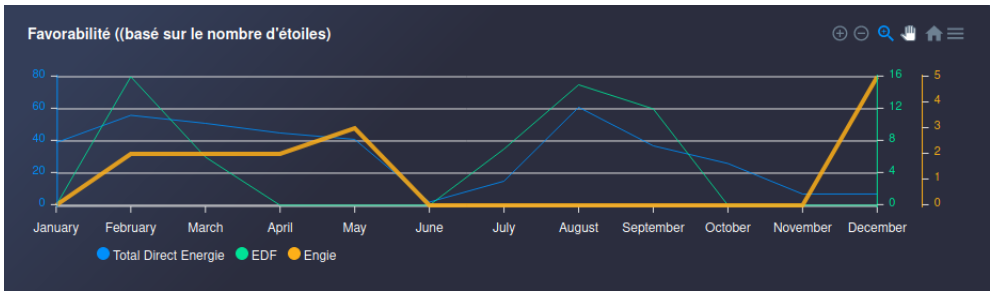

FIGURE 5.7 – Graphe de Favorabilité basée sur le nombre d'étoiles

Ces graphiques peuvent aider à comparer la popularité des fournisseurs d'énergie et à visualiser ses fluctuations au fil des mois d'une année donnée.

— **Nombre de critique par sujet**

Ce diagramme représente la proportion de chaque sujet par rapport au total des critiques (pour les trois fournisseurs d'énergie).

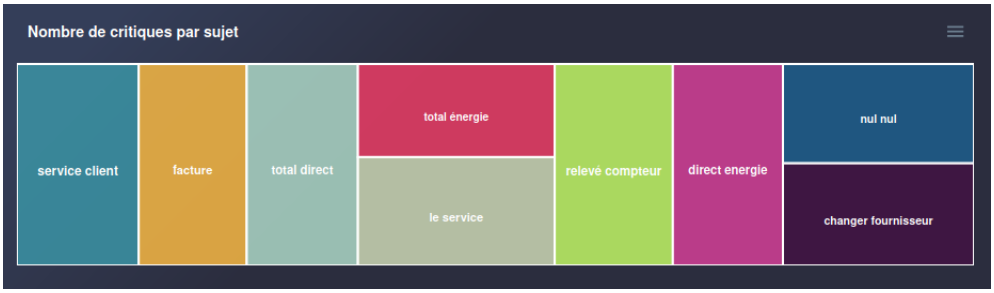

FIGURE 5.8 – Graphe de nombre de critique par sujet :

— **Proportion des critiques (Stacked Chart) :**

Ce graph représente le nombre d'critiques que chaque fournisseur d'énergie a reçu par mois (l'axe des abscisses est le même que celui des graphiques de tendance de favorabilité).

Vous pouvez focaliser sur un seul fournieur ou eliminer des fournisseurs dans ce graphes exactement comme ceux de Favorabilité.

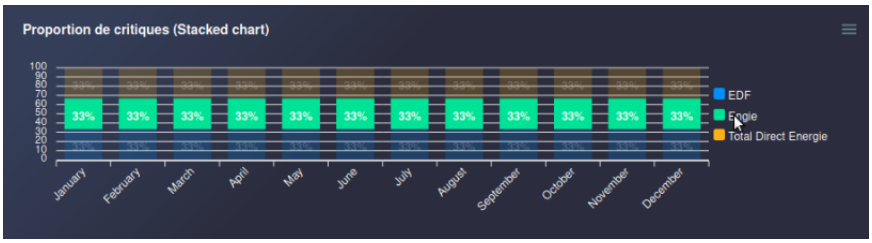

FIGURE 5.9 – Graphe de proportion des critiques (avec focalisation sur Engie)

— **Hystogramme Positif/Total par sujet :**

L'axe des x contient les sujets détectés par l'algorithme ML et l'axe des y est le pourcentage du ratio positif/total. Le graphique illustre les 3 fournisseurs d'énergie Total Direct Energie, EDF et Engie (chacun est représenté par une couleur spécifique) ce qui permet de comparer les trois en fonction des sujets. Vous pouvez faire disparaître un ou plusieurs fournisseurs d'énergie du graphique en cliquant sur leurs incons.

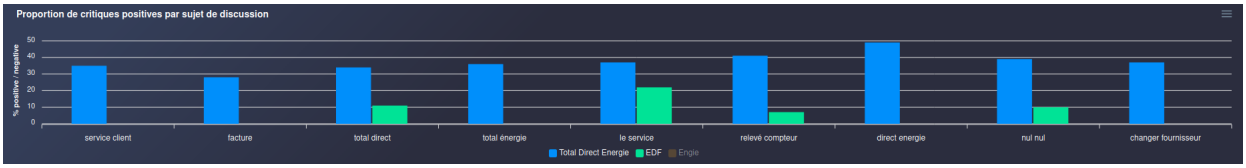

FIGURE 5.10 – Hystogramme Positif/négatif par sujet (avec Total Direct Energie éliminé)

— **Graphiques en radar :**

Les 3 graphiques en radar représentent les sujets par fournisseur d'énergie et combien de fois chaque sujet a été abordé dans les commentaires, ce qui offre la possibilité de comparer davantage les fournisseurs d'énergie. Parfois, le nom d'un autre fournisseur d'énergie figure parmi les sujets, ce qui indique généralement une comparaison entre les deux dans les commentaires (elle peut être positive ou négative).

FIGURE 5.11 – Graphiques en radar

— **Visualisation des données :**
Nous avons ajouté une fonctionnalité supplémentaire du dashboard : la possibilité de visualiser les données ; les données scrappées : les critiques par fournisseur d'énergie, les étoile et la date et l'heure de l'critiques ainsi que les résultats estimés : sujets LDA et sentiment. Nous avons également ajouté une sélection de filtres pour faciliter la consultation des données. Vous pouvez sélectionner les fournisseurs d'énergie que vous souhaitez afficher,choisir ceux qui sont dans un intervalle de nombre d'étoile de votre choix ainsi que ceux qui sont spécifiques à un sujet LDA (1). Vous pouvez rechercher des mots figurant à la fois dans les sujets LDA et dans les critiques (2).vous pouvez trier les critiques par avis, sentiment et étoiles (3). Vous pouvez également choisir le nombre de critiques affichés sur chaque page et naviguer facilement dans les pages grâce aux boutons que nous avons ajoutés (4).
Remarque : Les numeros **1**, **2**, **3** et **4** font référence à ceux de la figure ci-dessous.

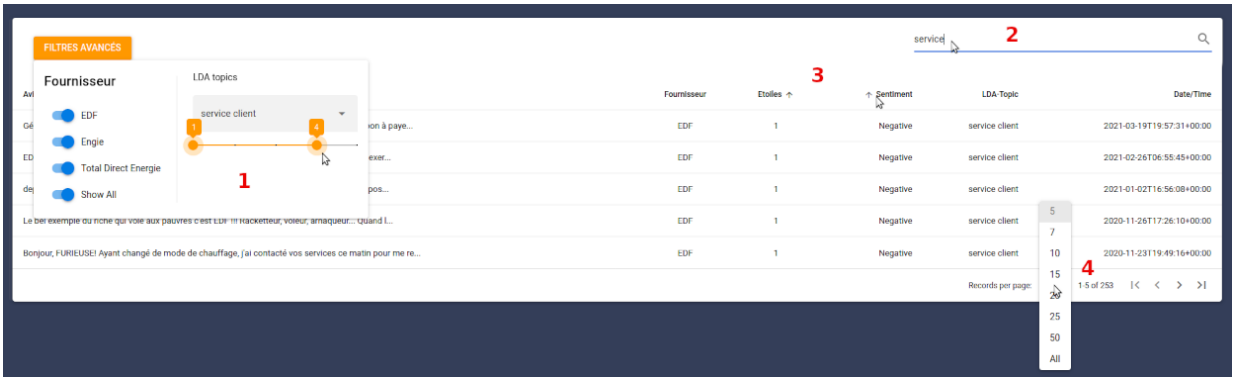

FIGURE 5.12 – Filtres appliqués sur les données

Chapitre 6

Conclusion

"Ce qui manque aujourd'hui ce n'est pas l'information mais la capacité à la traiter". Cette phrase prononcée par le prix Nobel d'économie Herbet Simon en 1950 peut sembler désuète mais reste tout de même d'actualité. Car malgré les progrès liés à l'informatique, notre projet a bien mis en évidence toutes les difficultés liés à l'analyse de l'information et à sa mise en valeur.

Tout d'abord nous avons rappelé à quel point les informations de la source d'origine, le site Trustpilot.com, pouvaient être biaisés. Nous avons aussi mis en avant les difficultés pour analyser du texte et détecter l'information qui nous intéresse. On ne peut donc pas analyser hativement les données sans une prise de recul et un minimum d'organisation dans le groupe.

Le data scientist a heureusement de nombreux outils à sa disposition. Il peut aisément collecter l'information grâce à une requête Python et la mettre en forme avec des scripts. Les données sont ensuites épurées à l'aide de regex et de stopwards pour faciliter et accélérer l'analyse. Cette dernière s'effectue à l'aide du Deep learning à l'aide du modèle BERT. Enfin nous avons synthétisé les informations au sein d'un tableau de bord mettant en évidence ce qui pourrait être utile aux futurs utilisateurs.

Toutefois parvenir à ces résultats ne s'est pas fait sans certaines concessions. Nous avons par exemple ignoré une partie des informations que nous n'avons pas jugé fiables comme la notation par étoile. Nous avons simplifié les textes d'avis avec les risques de perte d'information que l'on connaît. De plus le fait de se concentrer principalement sur les textes d'avis fait que nous accordons autant d'importance à un défaut mineur qu'à un défaut majeur aux yeux des consommateurs. Le fait de traiter les avis de manière globale sans chercher à priviligier un sujet précis peut conduire à de mauvaises interprétations. Enfin, le modèle a obtenu un score de précision d'environ 95%, qui peut potentiellement augmenter si les critiques non sensées ont été filtrées de l'ensemble de données au début.

Faut-il pour autant douter de la fiabilité de nos travaux et de notre tableau de bord ? Bien que perfectible, les outils utilisés permettent d'avoir une idée générale des avis des clients. Par ailleurs il faut rappeler que le tableau de bord est avant tout un instrument d'aide à la décision et non pas un oracle tout puissant qui reflète la réalité à 100%. Il incombe donc à l'utilisateur d'orienter ses recherches et d'en tirer les bonnes conclusions, tout en ayant conscience des limites, afin d'exploiter le dashboard le plus efficacement possible.

Chapitre 7

Annexes

7.1 Bibliographie

Hutto, C.J. Gilbert, E.E. (2014). VADER : A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text. Eighth International Conference on Weblogs and Social Media (ICWSM-14). Ann Arbor, MI, June 2014.

David Blei, Andrew Ng et Michael I. Jordan, « Latent Dirichlet Allocation », Proceedings of the 14th International Conference on Neural Information Processing Systems : Natural and Synthetic, Vancouver, British Columbia, Canada, MIT Press, no NIPS'01, 2001, p. 601-608)

Daniel Smilkov and Shan Carter Outil de simulation d'architecture de réseau de neurones simple, et de visualisation des résultats.
http ://playground.tensorflow.org

SIMONNET E., DELÉGLISE P., CAMELIN N. ESTÈVE Y. (2015) Exploring the use of attention based recurrent neural networks for spoken language understanding. In NIPS.

Google Research BERT : Pre-training of Deep Bidirectional Transformers for Language Understanding BERT-Base, Multilingual Cased : 104 languages, 12-layer, 768-hidden, 12-heads, 110M parameters, github.com/google-research/bert

CamemBERT model in CamemBERT : a Tasty French Language Model by Louis Martin, Benjamin Muller, Pedro Javier Ortiz Suárez, Yoann Dupont, Laurent Romary, Éric Villemonte de la Clergerie, Djamé Seddah, and Benoît Sagot. It is based on Facebook's RoBERTa model released in 2019. It is a model trained on 138GB of French text.

Berland Nicolas,"Mesurer et piloter la performance" Editions de la performance

Cochoy Frank, Une sociologie du packaging ou l'âne de Buridan face au marché, Paris, PUF