



Université Cadi Ayyad

Faculté des sciences Semlalia

Département d'informatique

Rapport de projet de fin d'études

Pour l'obtention du diplôme : Licence Fondamentale

Option : Mathématiques et Informatiques

Aspect based sentiment analysis
applied to aviation industry

Réalisé par :

LABIAD Salah Eddine

ZIRY Asmaa

Encadré par :

Dr. ZAHIR Jihad

Remerciements

Au terme de la réalisation de ce projet, c'est un devoir agréable d'exprimer en quelques lignes la reconnaissance que nous devons à tous ceux qui ont contribué de loin ou de près à l'élaboration de ce travail, qu'ils trouvent ici nos vifs respects et notre profonde gratitude.

Nous présentons nos profonds respects et nos reconnaissances à Dr. Jihad ZAHIR notre encadrante et notre professeur, pour sa patience, sa disponibilité et surtout ses judicieux conseils, qui ont contribué à alimenter notre réflexion pour pouvoir accomplir notre stage au sein de cet établissement dans les meilleures conditions.

Votre sujet a été une opportunité précieuse qui nous a permis d'acquérir des connaissances indispensables pour les fonctions auxquelles nous nous destinons. Cette expérience constituera un atout important en terme de notre futur parcours professionnel et académique.

Nous souhaitons adresser nos remerciements les plus sincères au corps professoral et administratif de la Faculté des Sciences Semlalia, pour la richesse et la qualité de leur enseignement et qui déploient des efforts considérables pour assurer à leurs étudiants une formation actualisée.

Table des matières

INTRODUCTION GÉNÉRALE.....	1
1. PRÉSENTATION GÉNÉRALE DU PROJET	4
1.1 MOTIVATIONS ET OBJECTIFS.....	4
1.2 INTRODUCTION	4
1.2.1 Comprendre l'énoncé du problème.....	4
1.2.2 Analyse des sentiments	5
1.3 PROBLEMATIQUE	7
1.4 SOLUTION PROPOSEE	7
2. ANALYSE ET CONCEPTION	8
2.1 METHODOLOGIE KDD MODIFIEE	8
2.2 VUE D'ENSEMBLE	9
2.3 ANALYSE EXPLORATOIRE DES DONNEES	9
2.3.1 Reconstitution de dataset.....	11
3. OUTILS ET TECHNOLOGIES UTILISÉS	14
3.1 PYTHON & FLASK.....	14
3.1.1 Python	14
3.1.2 Django vs Flask.....	14
3.2 NATURAL LANGUAGE PROCESSING.....	15
3.2.1 La bibliothèque NLTK.....	15
3.3 VISUALISATION	15
3.4 ALGORITHMES D'APPRENTISSAGE.....	15
3.4.1 Logistic regression	16
3.4.2 Multinomial Naïve Bayes	16
4. RÉALISATION ET MISE EN ŒUVRE	18
4.1 COLLECTE DES DONNEES	18
4.2 PRETRAITEMENT DES DONNEES.....	18
4.2.1 La mise en minuscule	18
4.2.2 La suppression des mention @	19
4.2.3 La suppression des hashtag #	19
4.2.4 La suppression des chiffres	19
4.2.5 La conversion des emojis et émoticônes	19
4.2.6 La suppression des URLs	19
4.2.7 La suppression de la ponctuation.....	20
4.2.8 La suppression des stop words	20
4.2.9 Stemming	20
4.3 PREPARATION DE DONNEES.....	21
4.3.1 Vectorisation de mots	21
4.3.2 Création de données d'apprentissage/test	23
4.4 CONSTRUCTION DES MODELE	24
4.4.1 Cross-validation	24
4.4.2 Pipelines	25

4.4.3	Hyperparameters	26
4.5	ÉVALUATION ET RESULTATS	29
4.5.1	Mesures d'évaluation	29
4.5.2	Discussion	31
4.6	VISUALISATION DES RESULTATS	33
4.6.1	L'interface	33
4.6.2	Flux de processus	33
4.6.3	Visualisation	34
5.	CONCLUSION	37
5.1	APPORTS ET GAINS	37
5.2	DIFFICULTES	39
5.3	PERSPECTIVES	39
RÉFÉRENCES		41
ANNEXES		44

Table des figures

FIGURE 1.1-1 TRANSPORT AÉRIEN, PASSAGERS TRANSPORTÉS - MAROC (2000-2018)	1
FIGURE 1.2-1 TECHNIQUES DE CLASSIFICATION DES SENTIMENTS	7
FIGURE 2.1-1 MÉTHODOLOGIE KDD MODIFIÉE	8
FIGURE 2.3-1 RÉPARTITION DES SENTIMENTS DANS LE DATASET	10
FIGURE 2.3-2 RÉPARTITION DES RAISONS DU SENTIMENT NÉGATIF DANS LE DATASET	10
FIGURE 2.3-3 LE NOMBRE DE MOTS SELON CHAQUE SENTIMENT DANS LE DATASET	11
FIGURE 2.3-4 LES PHASES DE SÉLECTION DES TWEETS DU NOUVEAU DATASET	12
FIGURE 2.3-5 RÉPARTITION DES RAISON DU SENTIMENT NÉGATIF DANS LE NOUVEAU DATASET	13
FIGURE 4.3-1 DÉMONSTRATION DE COUNTVECTORIZER	22
FIGURE 4.3-2 DÉMONSTRATION TF-IDFVECTORIZER	23
FIGURE 4.3-3 REPRÉSENTATION DE TRAIN/TEST SPLIT	24
FIGURE 4.4-1 REPRÉSENTATION DE TRAIN/TEST SPLIT ET CROSS VALIDATION	25
FIGURE 4.4-2 LE PROCESSUS DE CROSS-VALIDATION DE 5-FOLD	26
FIGURE 4.6-1 MODÈLE MVT(MODEL-VIEW-TEMPLATE)	33
FIGURE 4.6-2 ORGANIGRAMME DE L'APPLICATION	34
FIGURE 4.6-3 DISTRIBUTION DES SENTIMENTS POUR CHAQUE COMPAGNIE AÉRIENNE	35
FIGURE 4.6-4 LES ASPECTS RECONNUS POUR CHAQUE COMPAGNIE AÉRIENNE ET LEURS INTENSITÉ	35
FIGURE 4.6-5 VISUALISATION DE LA CARTE DU MONDE	36
FIGURE 5.3-1 PAGE D'ACCUEIL AIRCOMPARE	44
FIGURE 5.3-2 FORMULAIRE DE RECHERCHE EN TEMPS RÉEL	44
FIGURE 5.3-3 PAGE FEATURES	45
FIGURE 5.3-4 RENSEIGNEMENT SUR AIRCOMPARE	45
FIGURE 5.3-5 RÉCAPITULATIF DE LA RECHERCHE	45
FIGURE 5.3-6 SÉLECTION DE LA MEILLEURE COMPAGNIE AÉRIENNE	46
FIGURE 5.3-7 VISUALISATION DU SENTIMENT ET DES RAISONS DE LA NÉGATIVITÉ	46
FIGURE 5.3-8 LOCALISATION DES TWEETS	46

Liste des tableaux

TABLEAU 2.3-1 EXEMPLES DE TWEETS DE CROWDFLOWER-AIRLINE-TWITTER-SENTIMENT	10
TABLEAU 2.3-3 EXEMPLES DE TWEETS DU NOUVEAU DATASET	13
TABLEAU 4.5-1 ANALYSE DE LA PERFORMANCE DU MODÈLE DE SENTIMENT	30
TABLEAU 4.5-2 ANALYSE DE LA PERFORMANCE DU MODÈLE D'ASPECT	31

Résumé

Partout dans le monde, Internet joue un rôle important dans la prise des décisions. De nos jours, une grande population de personnes partage leurs opinions et points de vue sur Internet via les médias sociaux, les blogs et d'autres plateformes en ligne. Cela conduit Internet à être rempli d'informations à la fois pertinentes et non pertinentes. Par conséquent, pour obtenir les informations souhaitées, il n'est pas possible de parcourir chaque document présent sur Internet.

L'année dernière, Michael Phillips, stagiaire en **science des données** à Cambridge Analytica, a publié des scripts dans un ensemble d'échantillons de travail sur son compte GitHub personnel. En tant qu'**outil d'exploration** de médias sociaux en temps réel qui utilise des outils courants comme Tweepy, Matplotlib et TextBlob, cela ne semble pas être de la science-fiction ou extrêmement complexe. Cependant, ce n'est pas ce qui rend le code intéressant en tant que recherche clé, preuve politique et objet culturel. La partie la plus fascinante du **sentiment-miner** sur Twitter que Phillips a publiée est la façon dont il semble extraire les identifiants des utilisateurs et trouver leurs «tweets récents» et leurs favoris pour élargir le corpus de mots clés de l'entreprise autour des objets spécifiques de sentiment d'indignation électorale (immigration, contrôle des frontières...).

Partout dans le monde, de nombreuses personnes utilisent les blogs, les médias sociaux et d'autres plateformes en ligne en vue de partager leurs pensées ainsi que leurs opinions via Internet. Cela engendre un grand défi de récupérer les informations souhaitées sur Internet en analysant chaque document.

Ici, Notre projet de fin d'étude qui a été effectué au sein du département d'informatique de la FSSM adopte l'analyse de sentiment comme une solution. Cette recherche vise à fournir une aide à la décision pour les clients afin de sélectionner la compagnie aérienne la mieux adaptée, en fournissant une analyse des sentiments au niveau Aspect à partir des opinions des autres clients présents dans Twitter en collectant, analysant et visualisant les données. Nous avons aussi décidé de mettre en œuvre une plate-forme capable de répondre aux besoins exprimés.

Keywords: sentiment-miner, science des données, outil d'exploration

Abstract

Around the world, internet plays a significant role when it comes to decisions making. Nowadays, a large population of people shares their opinions and views on the internet through social media, blogs and other online platforms. This leads internet to be full of information both relevant and irrelevant. Therefore, in order to get the desired information, it is not possible to go through each document present on the internet.

Last year, Michael Phillips, a **data science** intern at Cambridge Analytica, posted the following scripts to a set of work samples” on his personal GitHub account. As a real-time social media **mining tool** which uses common tools like Tweepy, Matplotlib and TextBlob, this doesn’t appear to be science fiction or extremely complex. However, this is not what makes the code interesting as a key research, political evidence, and cultural object. The most fascinating part of the Twitter **sentiment-miner** that Phillips’ posted is how it appears to pull users’ IDs and find their «recent tweets» and favorites to expand the company’s corpus of keywords around specific objects of election outrage sentiment (ie, immigration, border control, etc.).

Many people around the world use blogs, social media and other online platforms to share their thoughts and opinions over the Internet. This creates a great challenge to retrieve the desired information from the Internet by analyzing each document.

Here, our graduation project that was carried out within the IT department of the FSSM adopts sentiment analysis as a solution. This research aims to provide decision support for customers in order to select the most suitable airline, by providing an analysis of feelings at Aspect level based on the opinions of other customers present in Twitter by collecting, analyzing and visualizing data. We also decided to implement a platform capable of meeting the needs expressed.

Keywords: sentiment-miner, data science, mining tool

Introduction Générale

Un service américain en ligne d'actualité et de réseautage social appelé Twitter, permet à ses utilisateurs de bloguer et interagir avec d'autres personnes grâce à des courts messages, des tweets. Twitter a été créé en mars 2006 et lancé en Juillet de la même année. Dans ce projet, le site de microblogging Twitter est utilisé, puisque Twitter est l'un des sites de médias sociaux les plus populaires à travers le monde avec une disponibilité en plus de 40 langues. Les tweets sont sous forme des publications textuelles limitées à 280 caractères UTF-8 sur les mises à jour de petites choses qui se produisent dans la vie quotidienne des utilisateurs. La nature courte des mises à jour permet aux utilisateurs de publier rapidement en temps réel, atteignant immédiatement leur public. Par défaut, les tweets sont visibles publiquement, mais le propriétaire peut définir la confidentialité pour ne s'afficher que pour ses amis.

L'industrie du transport aérien est l'une des plus grandes et des plus importantes industries au monde qui offre des services à des milliers de clients en une seule journée. Environ 2 152 362 passagers adoptent des vols au Maroc par mois selon les rapports fournis par l'Office National des Aéroports [ONDA]. La figure ci-dessous donne un aperçu sur les passagers aériens transportés comprennent les passagers des avions nationaux et internationaux des transporteurs aériens enregistrés dans le Maroc pendant 2000-2018.

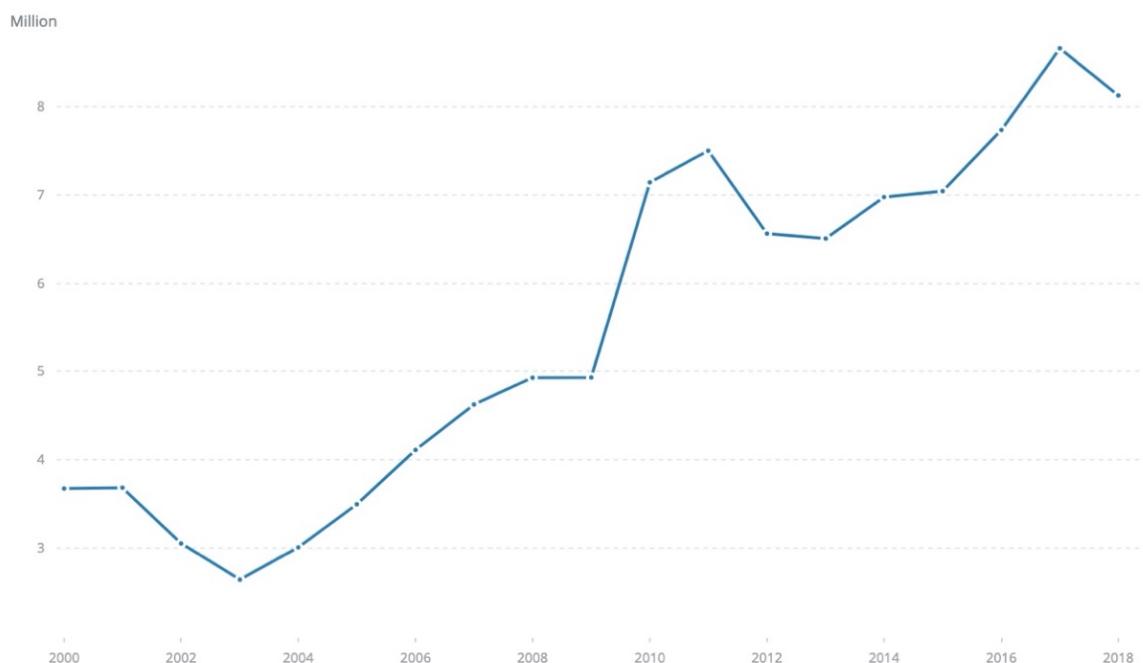


Figure 1.1-1 Transport aérien, passagers transportés - Maroc (2000-2018)

Royal Air Maroc, Air France, Ryanair, EasyJet, Etihad Airways, EgyptAir, Qatar Airways, Air arabia, Tuifly.be et Saudia, sont parmi les principaux transporteurs aériens au Maroc.

Au Maroc, la même zone géographique est couverte par ces compagnies aériennes pendant le vol. Ainsi, une grande concurrence se déroule entre eux, ce qui les oblige à créer un bon avantage concurrentiel.

Afin d'obtenir des avis ou des opinions des clients, les gens avaient l'habitude de préparer des questionnaires et des enquêtes sur la base des réponses, ils obtiennent par la suite des informations sur leurs services du point de vue du client. Ce processus prenait beaucoup de temps et parfois les clients ne répondaient pas sérieusement aux questions de l'interrogateur ou des formulaires et les laissaient parfois vides, ce qui conduisait parfois à des informations trompeuses. De surcroît, les questions des intervenants ont été conçues spécialement pour leurs besoins et n'étaient pas accessibles au public. Dans ce monde innovant et en perpétuelle évolution, chacun a la liberté de s'exprimer, d'exprimer ses opinions sur n'importe quelle plateforme. Par conséquent, les gens utilisent l'opinion des autres et prennent des suggestions avant de prendre des décisions afin d'obtenir une expérience meilleure et sans tracas. Cela permet non seulement d'économiser leur temps, mais aussi leur argent, leur énergie et leurs efforts. Étant donné que n'importe qui peut publier son point de vue sur Internet, il devient parfois très difficile pour les gens de parcourir chaque article ou tweet ou feedback afin de décrocher les informations qu'ils souhaitent. Dans ces scénarios, l'analyse des sentiments sert à extraire les réactions, les opinions, les critiques et les commentaires des gens à l'égard d'un produit ou service spécifique d'une entreprise. Elle a fait l'objet d'un examen considérable ces dernières années.

Ce document englobe le travail effectué dans le cadre d'un projet interne au sein du département d'informatique de la faculté des Sciences Semlalia de Marrakech. Le sujet étant de *concevoir et réaliser une solution d'analyse des sentiments au niveau Aspect à partir des opinions des autres clients présents dans Twitter*, on avait comme objectifs principaux de :

- Collecter les Tweets.
- Analyser et traiter les données en utilisant des techniques spéciales.
- Visualiser les résultats obtenus.

1. Présentation générale du projet

1.1 Motivations et Objectifs

Étant passionné par l'intelligence artificielle et ses sous-ensemble (Machine Learning, Deep Learning...), nous sommes très intéressés par l'offre de ce stage.

le défi technique, qui consistait à l'utilisation de plusieurs technologies dans le but de répondre à un besoin précis nous a attirés. C'était pour nous une occasion pour tester nos capacités d'analyse, de conception et de réalisation.

1.2 Introduction

1.2.1 Comprendre l'énoncé du problème

Notre objectif final est de reconnaître les 6 aspects différents que les gens sont susceptibles de mentionner lorsqu'ils parlent de leur expérience avec les compagnies aériennes, et d'évaluer leur opinion par rapport à ces aspects. Ceci nous permettra de voir comment chaque compagnie aérienne a été évaluée par ses clients en se basant sur six aspects intéressants:

- **Value for money** : Un utilitaire dérivé de chaque achat ou de chaque somme d'argent dépensée. Il est basé non seulement sur le prix d'achat minimum (économie) mais également sur l'efficience et l'efficacité maximales de l'achat.
- **Ground service** : Fait référence à la large gamme de services fournis pour faciliter le vol d'un aéronef ou le repositionnement au sol d'un aéronef, la préparation et la conclusion d'un vol, qui comprendra à la fois des fonctions de service client et de service au sol.
- **Cabin Staff service** : Fournir un niveau exceptionnel de service client aux passagers pendant leur voyage, tout en assurant leur confort et leur sécurité, avec une attitude joyeuse et aidante.
- **Legroom** : Les compagnies aériennes sont souvent comparées en termes de marge pour les jambes qu'elles fournissent dans chaque classe de service. Surtout classe économique, l'espace des sièges est souvent la différence la plus critique entre les différentes compagnies aériennes.

- **Food & Beverages** : Le repas servi aux passagers à bord d'un avion. Ces repas sont préparés par des spécialistes de restauration aérienne et normalement servis aux passagers à l'aide d'un chariot de service aérien.
- **Inflight entertainment** : fait référence aux divertissements offerts aux passagers d'un avion pendant un vol.

Le bruit est considéré comme un obstacle lorsqu'on analyse le contenu des médias sociaux. À titre d'exemple, nous avons essayé d'analyser les Tweets pour la première fois; nous avons constaté que près des deux tiers n'avaient aucune mention aux aspects que nous avons énumérés ci-dessus. Ces tweets se concentraient principalement sur des sujets comme le marketing d'entreprise ou même des blagues sur les compagnies aériennes ou des bots de tracking.

D'une part, cela pourrait être surprenant de voir des aspects tels que Food & Beverages et Legroom mentionnés si rarement – quand on pense des personnes critiquant les compagnies aériennes, on les imagine se plaindre de la nourriture ou le manque d'espace pour leurs jambes. Mais d'une autre part, cela n'est pas inattendu ou étonnant d'entendre beaucoup parler des autres aspects tels que Ground service et Cabin Staff service.

On peut également spéculer que le confort et la nourriture sont assez standard chez les compagnies aériennes, mais la ponctualité peut varier, de sorte que les gens peuvent être déçus par ces comportements.

Ground service est le plus mentionné lors des tweets concernant une compagnie aérienne, les gens sont très négatifs sur les services aériens au sol en général.

La deuxième plus grande préoccupation des consommateurs était la ponctualité, ces derniers se plaignent de la ponctualité. Personne ne peut nier que la nourriture des compagnies aériennes n'est pas la meilleure, mais lorsque nous avons examiné le sentiment à propos de Food & Beverages dans les Tweets, nous avons constaté qu'ils ne se soucient généralement pas de cet aspect.

1.2.2 Analyse des sentiments

L'analyse des sentiments, ou soi-disant exploration d'opinion, a été étudiée par de nombreux chercheurs ces dernières années. L'analyse des sentiments est un type d'étude informatique du texte en langage naturel qui vise à identifier la polarité, l'intensité et les sujets des sentiments auxquels ces

sentiments s'appliquent. L'analyse des sentiments révèle la nécessité d'un système automatisé de divulgation et de synthèse des opinions traitant de grandes quantités de données pour permettre à la machine de comprendre le contenu généré par l'être humaine. Techniquelement, l'analyse des sentiments fait partie de l'étude Natural Language Processing (NLP). La classification de la subjectivité et la classification des sentiments sont peut-être les sujets les plus étudiés dans ce domaine. La classification de la subjectivité est un processus pour séparer les phrases subjectives des phrases objectives ou distinguer les opinions des faits, tandis que la classification des sentiments est un processus pour déterminer l'orientation des sentiments, que cette phrase exprime des sentiments positifs ou négatifs. De plus, certaines recherches s'intéressent à mesurer l'intensité de la polarité des sentiments pour mesurer l'intensité sémantique. L'analyse des sentiments feature-based est une étude approfondie qui se réfère à la détermination des sentiments exprimés sur différentes caractéristiques des entités.

Diverses méthodes d'analyse des sentiments ont été étudiées aux différents niveaux de granularité du texte. Le premier niveau commence à partir d'une tâche de classification au niveau du document vers un niveau de grain plus fin d'une phrase au niveau de la phrase. Fondamentalement, les approches utilisées pour l'analyse des sentiments peuvent être divisées en deux catégories, l'approche de Machine Learning et l'approche basée sur le lexique.

- Les approches de Machine Learning sont des approches d'apprentissage supervisé. Il s'agit d'un processus de formation qui apprend à un agent à classer les entrées en sorties. On s'arrête une fois que les données d'entraînement étiquetées avec des valeurs de sentiment sont apprises par l'algorithme. Nous citons brièvement trois algorithmes de classification populaires : Naïve Bayes(NB), Max Entropy(MaxEnt) ou bien Multinomial Logit model, Support Vector Machines(SVM).
- Les approches basées sur le lexique sont généralement des approches non supervisées. Il s'agit d'une règle basée sur des caractéristiques fournies par un score de lexique sentimental prédéterminé pour estimer la polarité, qu'elle soit positive ou négative. Ces approches peuvent fonctionner sans corpus de référence et sans apprentissage préalable. Les lexiques de sentiment estiment correctement la polarité

générique du terme d'une manière qui ne prend pas en compte les informations de domaine.

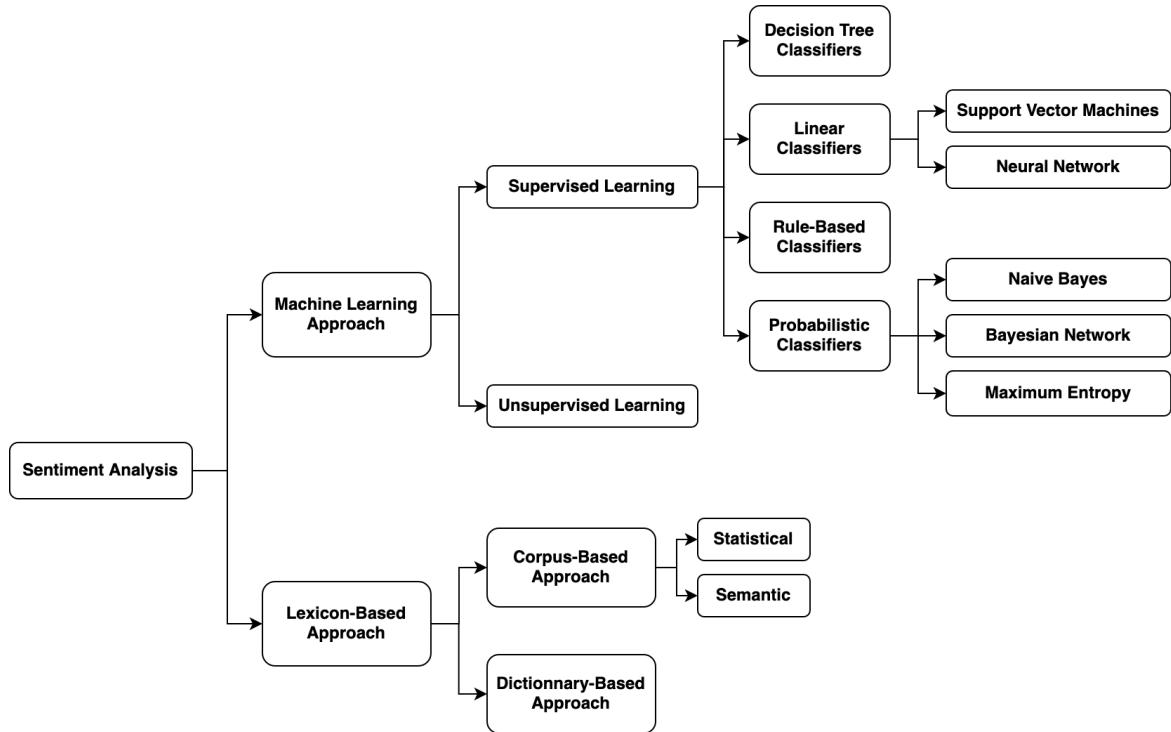


Figure 1.2-1 Techniques de classification des sentiments

1.3 Problématique

L'analyse du sentiment des données des tweets est perçue comme un problème beaucoup plus difficile que celui des textes conventionnels tels que les documents. Cela est dû en partie à la courte longueur des tweets, à l'utilisation fréquente de mots informels et irréguliers et à l'évolution rapide du langage sur Twitter.

1.4 Solution proposée

Dans ce projet, nous proposons une approche basée sur l'analyse des sentiments pour déterminer si un tweet possède un sentiment positif, neutre ou négatif et pour en déterminer les aspects nous utilisons Twitter pour représenter le service de micro-blog, et pour inférer le sentiment des tweets et la raison de négativité, nous utilisons deux classificateurs: Logistic Regression et MultinomialNB. Nous allons régler les hyperparamètres des deux classificateurs avec Gridsearch. Nous comparerons la performance avec trois mesures: precision, recall et F-score.

2. Analyse et Conception

2.1 Méthodologie KDD modifiée

Le processus d'extraction des informations essentielles des données pour fournir un meilleur support aux décisions est connu sous le nom de Data Mining. Des méthodologies telles que Knowledge Discovery in Databases (KDD) et Cross Industry Standard Process for Data Mining (CRISP-DM) sont largement utilisées pour Data Mining. La méthodologie CRISP-DM a été introduite en 1996 et suit un flux de processus en six étapes. Étant donné que dans ce projet, des étapes comme *business understanding* ne peut pas être utilisée, la méthodologie CRISP-DM ne correspond donc pas à ce projet. Ce projet suivra une méthodologie KDD modifiée où quelques étapes sont modifiées comme illustrée dans la figure 2.1-1, par ex. *étape de mise en œuvre* modifiée en étape de *détection de sentiment et de raison de négativité* et des attributs supplémentaires sont ajoutés.

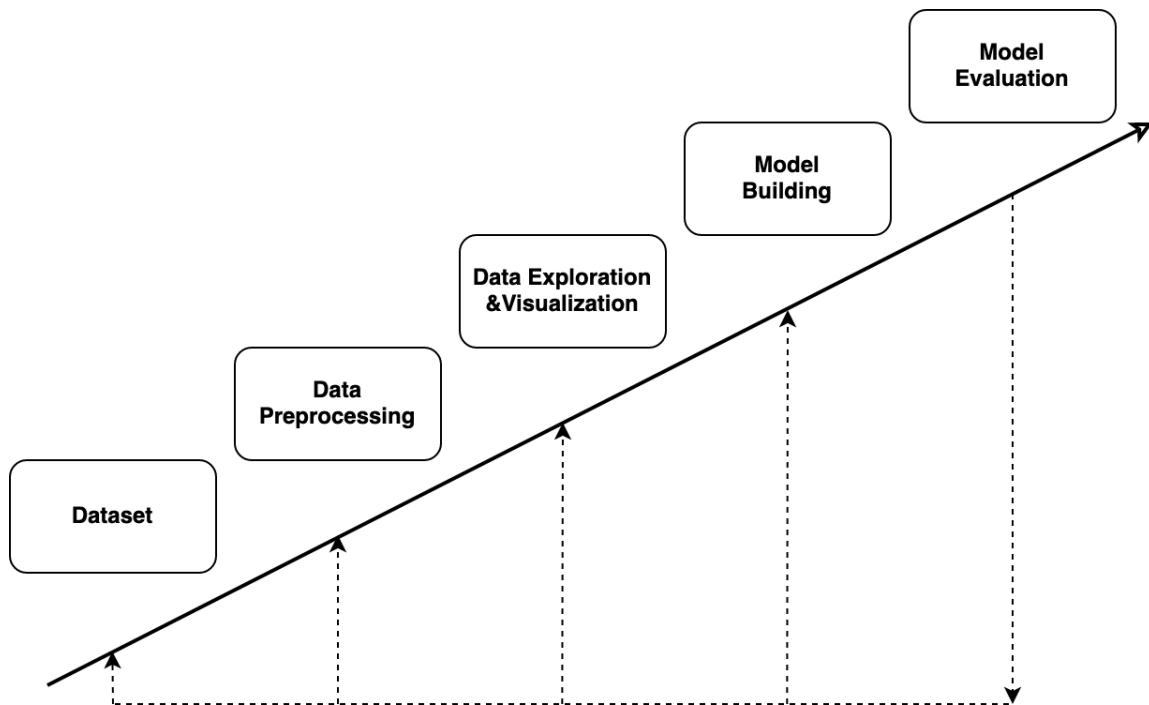


Figure 2.1-1 Méthodologie KDD modifiée

2.2 Vue d'ensemble

La partie principale de notre travail est résumée ci-dessous:

- L'approche utilise les avis des passagers recueillis sur le service de micro-blog Twitter.
- Élaboration d'une stratégie de prétraitement pour gérer la suppression des informations sans importance.
- La représentation des tokens en tant que sparse matrix et le pipelining sont appliqués pour améliorer les performances des classifieurs.
- Représentation graphique des données sentimentalement analysées via une application Web qui permet de comparer les compagnies aériennes.

2.3 Analyse exploratoire des données

Pour le travail et les expériences décrits dans ce projet, nous avons utilisé le dataset *crowdflower-airline-twitter-sentiment*. La table 2.3-1 donne un aperçu du dataset. Cet ensemble de données comprend 14 640 tweets en anglais. Les classifieurs ont été générés en utilisant 2363 avis positifs, 9178 avis négatifs et 3099 avis neutres.

La taille de la base d'apprentissage est de : 13176 tweets dont 2125 positifs, 8247 négatifs et 2804 neutres. La base de test contenait 1464 tweets dont 238 positifs, 931 négatifs et 295 neutres.

tweet_id	airline_sentiment	negativereason	text	tweet_created
570306133677760513	neutral	--	@VirginAmerica What @dhepburn said.	2015-02-24 11:35:52 -0800
570301130888122368	positive	--	@VirginAmerica plus you've added commercials to the experience... tacky.	2015-02-24 11:15:59 -0800
570301031407624196	negative	Bad Flight	@VirginAmerica it's really aggressive to blast obnoxious "entertainment" in your guests' faces & they have little recourse	2015-02-24 11:15:36 -0800
...

569363173817643009	negative	Late Flight	@united not a good day to fly u, two delayed flights still on plane at Ewr waiting over 1,5 hours for gate get to gate no gate agent.	2015-02-21 21:08:53 -0800
---------------------------	----------	-------------	---	------------------------------

Tableau 2.3-1 Exemples de tweets de crowdflower-airline-twitter-sentiment

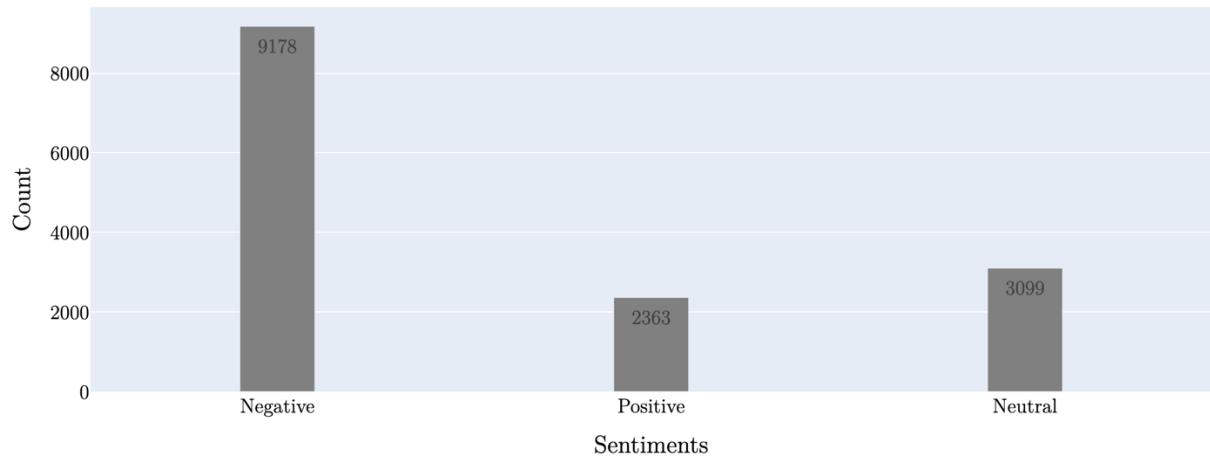


Figure 2.3-1 Répartition des sentiments dans le dataset

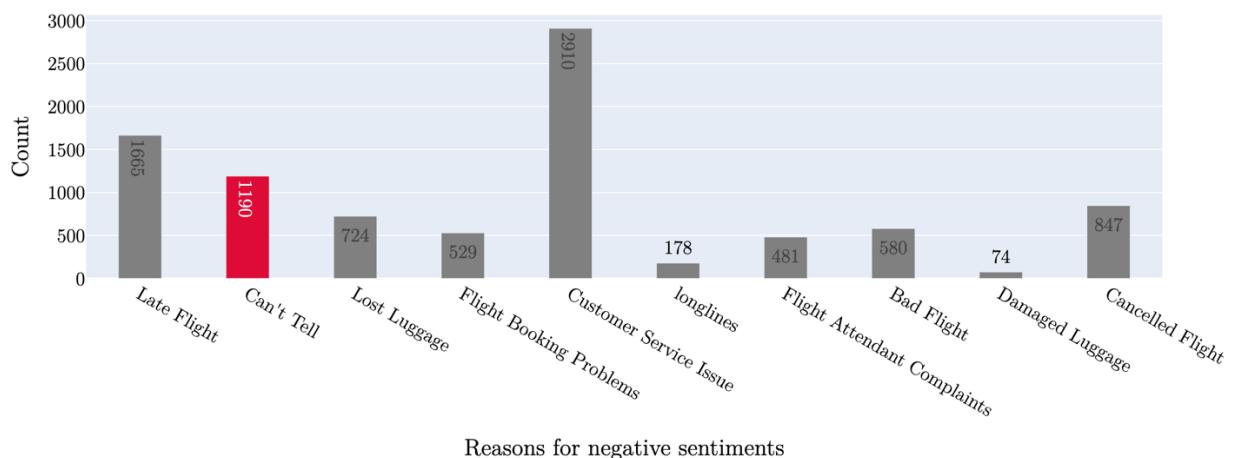


Figure 2.3-2 Répartition des raisons du sentiment négatif dans le dataset

Les étiquettes de classe sont déséquilibrées comme nous pouvons le constater ci-dessus dans la figure 2.3-1 et la figure 2.3-2. C'est quelque chose que nous devons les tenir en compte pendant la phase d'entraînement des classifieurs.

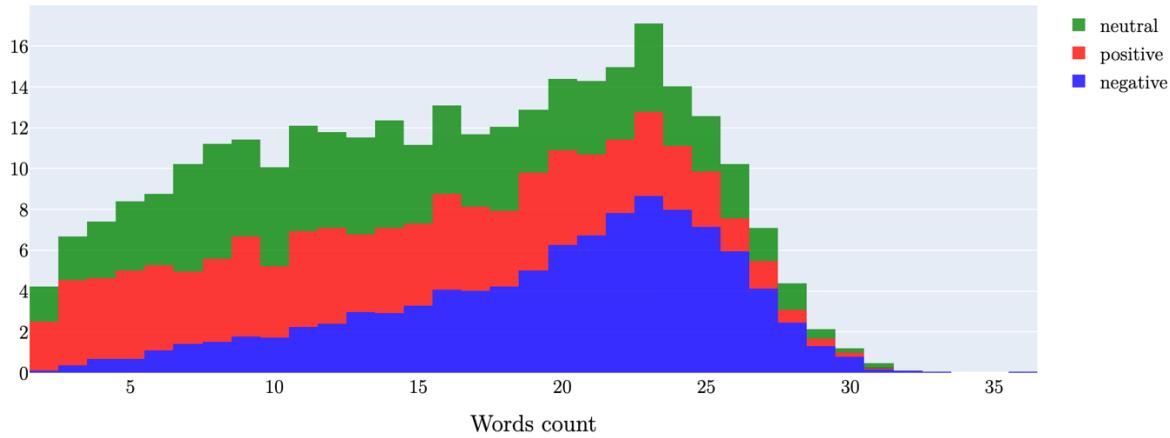


Figure 2.3-3 Le nombre de mots selon chaque sentiment dans le dataset

- D'après la figure 2.3-3 le nombre de mots utilisés dans les tweets est plutôt faible. Le plus grand nombre de mots est 36 et il y a même des tweets avec seulement 2 mots. Nous devrons donc faire attention lors du prétraitement des données pour ne pas supprimer trop de mots. Cependant le traitement de texte sera plus rapide. Les tweets négatifs contiennent plus de mots que les tweets neutres ou positifs.
- Tous les tweets ont au moins une mention. C'est le résultat de l'extraction des tweets sur la base des mentions dans les données Twitter. Il ne semble pas y avoir de différence dans le nombre de mentions concernant le sentiment.
- La plupart des tweets ne contiennent pas de hashtags. Cette variable ne sera donc pas conservée lors de l'entraînement du modèle. Encore une fois, aucune différence dans le nombre de hashtags en ce qui concerne le sentiment.
- La plupart des tweets ne contiennent pas de mots en majuscule et nous n'observons aucune différence de distribution entre les sentiments.
- Les tweets positifs semblent utiliser un peu plus d'exclamation ou de points d'interrogation.
- La plupart des tweets ne contiennent pas d'URL.
- La plupart des tweets n'utilisent pas d'emojis.

2.3.1 Reconstitution de dataset

Dans cette partie, nous avons reconstruit un nouvel ensemble de données qui contient tous les 6 aspects introduits dans l'introduction.

Tout d'abord, nous avons utilisé tous les 9178 tweets avec un sentiment négatif, puis nous avons regroupé ces tweets par la colonne *negativereson*. Nous constatons que de nombreux tweets de la catégorie *Can't tell* contient des mots clés liés aux autres aspects. À ce moment, nous avons commencé à filtrer les tweets manuellement en fonction de mots clés liés à chaque aspect. Par exemple pour Food & Beverages on recherche les tweets qui contiennent des mots-clés comme : food, drink, vegan, coffee, meal, gluten, coke, caffeine, cookie, snack, water...

puis nous vérifions le contexte des tweets s'ils sont en relation ou non. La figure 2.3-4 résume la méthode.

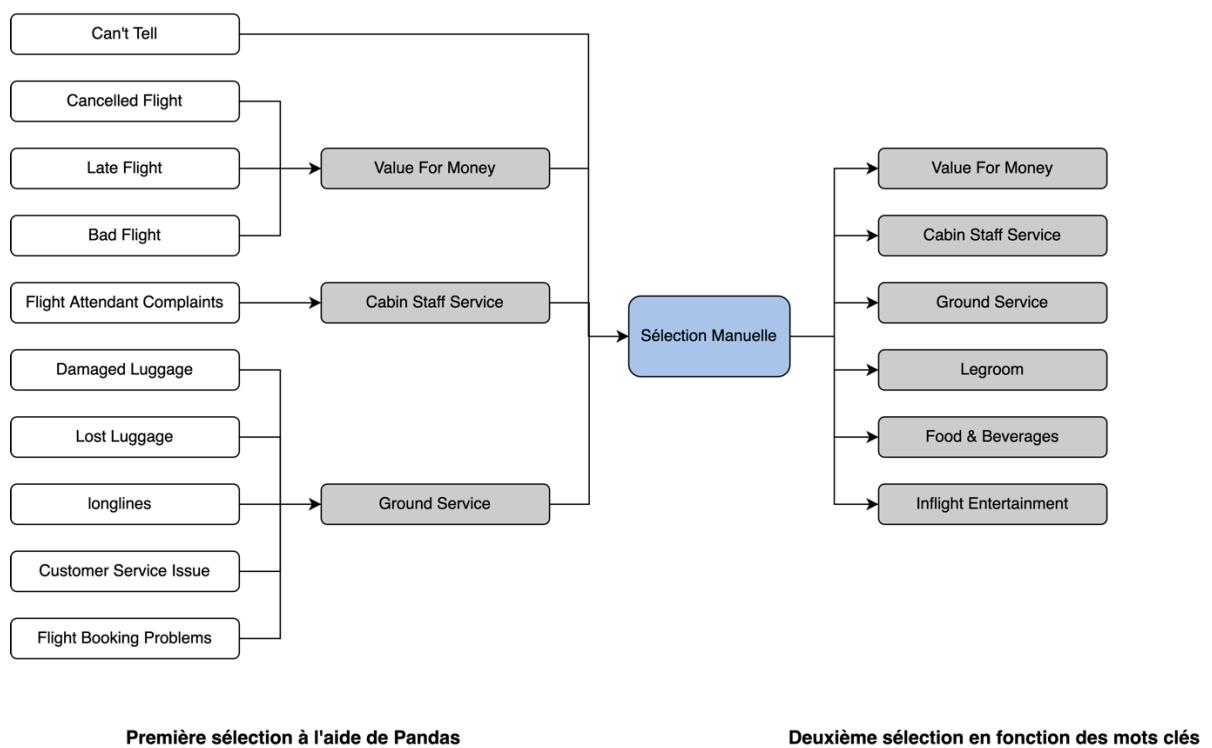


Figure 2.3-4 Les phases de sélection des tweets du nouveau dataset

La table 2.3-3 donne un aperçu du dataset. Cet ensemble de données comprend 5543 tweets en anglais.

tweet_id	text	aspect
570426133672340513	@USAirways ...these changes as well to Late Flightr find out that the flight I was scheduled for isn't ready	Value For Money
570301130888124135	@SouthwestAir and now no wifi??? Come on.	Inflight Entertainment

570301031407629324

@AmericanAir your customer service
is deplorable. I am disgusted in your
company and the ignorant people on
the phones for lost baggage.

Ground Service

569363173817649085

If you're flying Ryanair anytime soon,
at least you'll know that Stansted Labs
is the company who have probably
tested the drinking water for
#legionella. One less thing to worry
about.

Food & Beverages

Tableau 2.3-2 Exemples de tweets du nouveau dataset

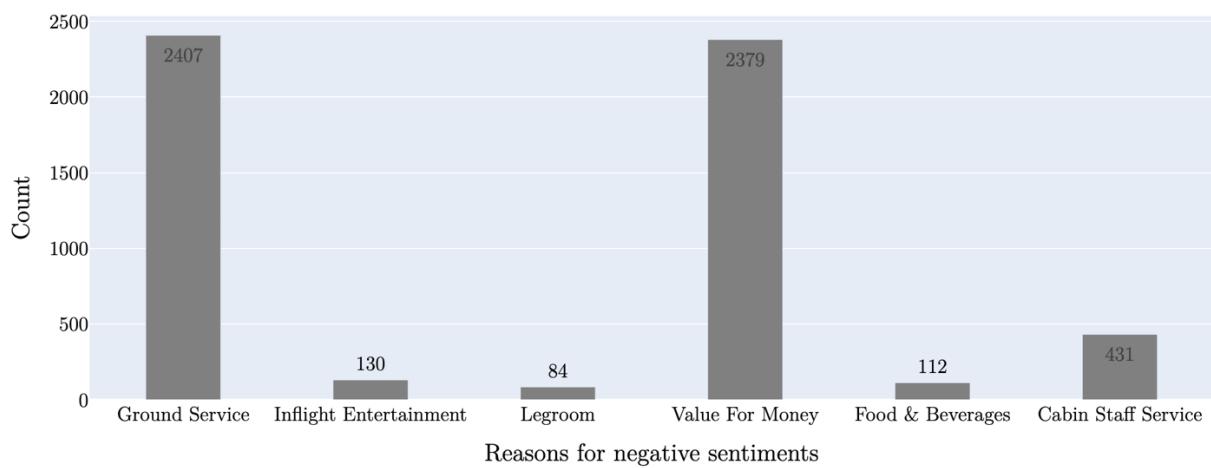


Figure 2.3-5 Répartition des raison du sentiment négatif dans le nouveau dataset

3. Outils et Technologies utilisés

3.1 Python & Flask

3.1.1 Python

Python est facile à utiliser en matière de calcul quantitatif et analytique. C'est un leader depuis un certain temps, maintenant il est largement utilisé dans divers domaines comme le traitement du signal, la finance et d'autres. De plus, Python a été utilisé pour renforcer l'infrastructure interne de Google et pour créer des applications comme YouTube.

Python est largement utilisé en plus d'être un langage flexible et open source. Ses bibliothèques massives sont destinées à la manipulation de données et sont très faciles à apprendre même pour un analyste de données débutant.

3.1.2 Django vs Flask

Principale différence entre Django et Flask, Django fournit un cadre complet de modèle - vue - contrôleur. Son objectif est de simplifier le processus de développement de sites Web. Il repose sur moins de code, des composants réutilisables et un développement rapide. Flask, d'autre part, est un microframework basé sur le concept de bien faire une chose. Il ne fournit pas d'ORM et n'est fourni qu'avec un ensemble d'outils de base pour le développement Web.

Les applications Flask sont principalement des applications à page unique (SPA). C'est un bon choix pour les applications Web de petite et moyenne taille.

Afin de mener à bien notre projet, nous avons opté pour l'utilisation de ce dernier pour plusieurs raisons plus techniques, à savoir :

- Besoin d'une structure permettant de gérer plusieurs fichiers statiques.
- Besoin de transmission de données de façon directe vers du code JavaScript.
- Flask est minimaliste et n'a aucune restriction, ce qui signifie que nous pouvons implémenter exactement ce que nous voulons en utilisant des bibliothèques externes.

3.2 Natural Language Processing

Le traitement du langage naturel (NLP) est une branche de l'intelligence artificielle qui aide les ordinateurs à comprendre, interpréter et manipuler le langage humain. La NLP fait appel à de nombreuses disciplines, y compris l'informatique et la linguistique informatique, dans sa quête pour combler le fossé entre la communication humaine et la compréhension informatique.

3.2.1 La bibliothèque NLTK

NLTK (Natural Language Toolkit) est une plate-forme leader pour la construction de programmes Python pour travailler avec des données de langage humain. Il fournit des interfaces faciles à utiliser à plus de 50 corpus et ressources lexicales telles que WordNet, ainsi qu'une suite de bibliothèques de traitement de texte pour classification, tokenization, stemming, tagging, parsing, et semantic reasoning.

3.3 Visualisation

Plotly Express est une nouvelle bibliothèque de visualisation Python de haut niveau: c'est un wrapper pour Plotly.py qui expose une syntaxe simple pour les graphes complexes. Inspiré par Seaborn et ggplot2, il a été spécialement conçu pour avoir une API concise, cohérente et facile à apprendre. Cela donnera la possibilité de créer des tracés richement interactifs en un seul appel de fonction, y compris les facettes, les cartes, les animations...

3.4 Algorithmes d'apprentissage

Scikit-learn est l'un des packages Python les plus utilisés pour la Big Data et Machine Learning. Il permet d'effectuer de nombreuses opérations et fournit une variété d'algorithmes d'apprentissage non supervisés et supervisés. Il est construit sur NumPy et SciPy. Il s'agit d'un package open-source comme la plupart des éléments de l'écosystème Python.

3.4.1 Logistic regression

Logistic regression est un classifieur linéaire, il utilise donc une fonction linéaire également appelée logit :

$$f(x) = b_0 + b_1x_1 + \dots + b_rx_r$$

Les variables b_0, b_1, \dots, b_r sont les estimateurs des coefficients de régression, également appelés poids prédits ou juste coefficients.

La fonction de régression logistique $P(x)$ est la fonction sigmoïde de

$$f(x): P(x) = \frac{1}{1 + e^{-f(x)}}$$

En tant que telle, elle est souvent proche de 0 ou 1. La fonction $P(x)$ est souvent interprétée comme la probabilité prédite que la sortie pour un x donné soit égale à 1. Par conséquent, $1 - P(x)$ est la probabilité que la sortie soit 0.

Logistic regression détermine les meilleurs poids prédits b_0, b_1, \dots, b_r de sorte que la fonction $P(x)$ soit aussi proche que possible de toutes les réponses réelles $y_i, i = 1, \dots, n$, où n est le nombre d'observations. Le processus de calcul des meilleurs poids à l'aide des observations disponibles est appelé apprentissage du modèle ou ajustement.

Pour obtenir les meilleurs poids, on maximise la fonction log-likelihood (LLF) pour toutes les observations $i = 1, \dots, n$. Cette méthode est appelée estimation du maximum de likelihood et il est représentée par l'équation :

$$LLF = \sum_i (y_i \log(P(x_i)) + (1 - y_i) \log(1 - P(x_i)))$$

Logistic regression est rapide, simple et pratique pour interpréter les résultats. Bien qu'il s'agisse essentiellement d'une méthode de classification binaire, elle peut également être appliquée à des problèmes multi classes.

3.4.2 Multinomial Naïve Bayes

Multinomial Naïve Bayes est le classifieur le plus employé pour les problèmes de classification des documents. La conception basique des classificateurs Naïve Bayes les rend attrayants, de plus, ils se sont révélés rapides, fiables et précis dans les applications de la NLP. Il s'appuient sur le théorème de Bayes, qui

est une équation décrivant la relation des probabilités conditionnelles des quantités statistiques.

Dans MultinomialNB, un document d peut être supposé être une collection d'entités de mots f_i . Il est représenté par $d = \{f_1, f_2, \dots, f_n\}$. f_i peut-être supposé être la fréquence du mot w_i dans un document. Le vocabulaire est le dictionnaire du répertoire distinctif de mots dans la collection totale de documents. Les documents sont représentés comme un modèle de bag of words car la fréquence des mots dans un document est répertoriée en tant que fonctionnalité. La probabilité de classe c étant donné le document de test d est représentée par :

$$P(c|d) = \frac{P(c) \prod_{i=1}^n P(w_i|c)^{f_i}}{P(d)}$$

où $P(c)$ est la probabilité antérieure d'une classe c , $P(c)$ est la probabilité qu'un mot w_i apparaisse dans un document d correspondant à la classe c , $P(d)$ est la probabilité d'un document et f_i est le nombre d'occurrences d'un mot dans un document. La probabilité antérieure d'une classe est donnée par $P(c) = \frac{N_c}{N}$ où N_c est le nombre de documents correspondant à la classe c et N le nombre total de documents. La probabilité conditionnelle $P(c)$ qu'un mot w_i apparaisse dans un document d d'une classe est donnée par $P(c) = \frac{N_{ci} + \alpha}{N_c + \alpha \cdot |V|}$, où N_{ci} est le nombre de fois w_i apparaissant en c et est N_c le nombre total de mots dans la classe c et $|V|$ est le nombre de mots uniques dans le vocabulaire.

Afin d'éviter des probabilités nulles, le paramètre α est défini. Si $\alpha = 1$, le lissage appliqué est appelé lissage Laplacien.

4. Réalisation et mise en œuvre

4.1 Collecte des données

L'API Twitter expose des dizaines de points de terminaison HTTP qui peuvent être utilisés pour récupérer, créer et supprimer des tweets, des retweets et des likes. Il fournit un accès direct à des données de tweet en temps réel, mais nécessite d'avoir un traitement de nombreux détails de bas niveau. Tweepy est un package open source qui permet de contourner un grand nombre de ces détails de bas niveau.

L'API Twitter utilise OAuth, qui est un protocole d'autorisation ouvert pour authentifier les demandes, donc la première chose à faire est d'avoir des clés d'accès (*CONSUMER_KEY*, *CONSUMER_SECRET*, *ACCESS_TOKEN*, *ACCESS_TOKEN_SECRET*) en contactant Twitter Developer.

L'API permet d'accéder aux tweets en temps réel. Les résultats sont filtrés par l'utilisateur en fonction de noms des compagnies aériennes qu'on a rassemblé dans une base de données.

Les tweets fournies par l'API seront regroupés dans un Dataframe pour faciliter la manipulation de ces données.

4.2 Prétraitement des données

Le prétraitement des données texte est une étape essentielle parce qu'il rend le texte brut prêt pour l'exploration, c'est-à-dire qu'il devient plus facile d'extraire des informations du texte et de lui appliquer des algorithmes de Machine Learning. Si nous sautons cette étape, il y a plus de chances que nous travaillions avec des données bruyantes et incohérentes. L'objectif de cette étape est de nettoyer le bruit qui est moins pertinent pour trouver le sentiment des tweets. Ce processus détient une possession de grande importance car il aide à la construction d'un modèle efficace, stable et robuste.

Il existe différentes stratégies de prétraitement. Dans ce projet, nous nous concentrerons sur les points suivants:

4.2.1 La mise en minuscule

La mise en minuscule de toutes les données, est l'une des formes de prétraitement de texte les plus simples et les plus efficaces. Il pourra aider

dans les cas où l'ensemble de données n'est pas assez volumineux et contribue de manière significative à la cohérence de la sortie attendue.

4.2.2 La suppression des mention @

Pour répondre ou mentionner, les utilisateurs peuvent référer d'autres utilisateurs dans leurs tweets en utilisant le symbole @ devant le nom d'utilisateur. Par exemple, "*@denis c'est tellement génial, félicitations!*" Cette action établit un lien entre les utilisateurs et permet des conversations entre utilisateurs. On procèdera dans un premier temps par la suppression des mentions, car nous voulons également généraliser aux tweets d'autres compagnies aériennes.

4.2.3 La suppression des hashtag #

Pour classer librement les tweets ensemble, les utilisateurs peuvent étiqueter en utilisant des hashtags ou le symbole # devant les mots. Par exemple, "*Le nouveau #iphoneX est incroyable*". Nous allons supprimer le signe (#) mais pas le tag réel puisqu'il peut contenir des informations.

4.2.4 La suppression des chiffres

les chiffres ne donnent pas beaucoup d'importance pour obtenir des informations utiles. On supprime tous les chiffres.

4.2.5 La conversion des emojis et émoticônes

Les emojis et les émoticônes deviennent la langue principale pour communiquer avec n'importe qui dans le monde. Les emojis et les émoticônes jouent un rôle majeur dans l'analyse de texte, on va les convertir en des mots expressifs. Par exemple :

😎 → smiling_face_with_sunglasses
:-) → happy_face_smiley

4.2.6 La suppression des URLs

Pour partager un lien dans un espace restreint, les utilisateurs peuvent utiliser le service de raccourcissement d'URL pour générer une URL abrégée

unique qui redirige vers le site Web souhaité. Depuis mars 2010, Twitter propose *t.co*, un service de raccourcissement des liens, pour les liens publiés sur Twitter afin de protéger les utilisateurs contre les sites malveillants et de suivre les clics sur les liens dans les tweets. Nous supprimerons les URL car ils ne contiennent pas d'informations utiles. Nous n'avons pas remarqué de différence dans le nombre d'URL utilisé entre les classes de sentiment.

4.2.7 La suppression de la ponctuation

L'apostrophe est un caractère de ponctuation important, qui doit être traité avec soin, car beaucoup de texte peut être basé sur des apostrophes. Par exemple, « ‘aren’t’, ‘shouldn’t’, ‘didn’t’, ‘would’ve’... ».

On supprime toutes la ponctuation, y compris les points d'interrogation et d'exclamation.

4.2.8 La suppression des stopwords

Les mots vides sont principalement un ensemble de mots couramment utilisés dans n'importe quelle langue. La raison pour laquelle les mots vides sont essentiels pour de nombreuses applications est que, si nous supprimons les mots qui sont très couramment utilisés dans une langue donnée, par exemple « ‘but’, ‘again’, ‘there’, ‘about’, ‘once’, ‘during’... » nous pouvons plutôt nous mettre en évidence sur les mots importants. Nous ne voudrions pas que ces mots prennent un temps de traitement précieux. Pour cela, nous pouvons les supprimer facilement, en stockant une liste de mots que nous considérons comme des mots vides. NLTK (Natural Language Toolkit) en python regroupe une liste de mots vides stockés en 16 langues différentes.

4.2.9 Stemming

Stemming est le processus de réduction de l'infexion des mots, par exemple *troublé*, *troubles* à leur forme racine *trouble*. La racine dans ce cas peut ne pas être un vrai mot racine, mais juste une forme canonique du mot d'origine. Le Stemming utilise un processus heuristique simple qui coupe les extrémités des mots en espérant transformer correctement les mots en leur forme racine. Les mots *trouble*, *troublé* et *troubles* pourraient en fait être convertis en *troubl* au lieu de *trouble* parce que les extrémités étaient juste coupées. Pour cela

nous avons utilisé l'algorithme Porter de NLTK, qui est connu pour être empiriquement efficace pour l'anglais.

4.3 Préparation de données

4.3.1 Vectorisation de mots

Les algorithmes de Machine Learning fonctionnent sur un espace d'entités numériques, en attendant l'entrée en tant que tableau à deux dimensions où les lignes sont des instances et les colonnes sont des entités. Afin d'appliquer un algorithme de Machine Learning sur un texte, nous devons transformer nos données en représentations vectorielles. Ce processus est appelé extraction d'entités ou plus simplement, vectorisation, il constitue une étape essentielle dans une analyse. La représentation numérique des documents nous donne la possibilité d'effectuer des analyses significatives et crée également les instances sur lesquelles les algorithmes de Machine Learning fonctionnent. Dans l'analyse de texte, les instances sont des documents ou des énoncés entiers, dont la longueur peut varier de citations ou tweets à des livres entiers. Dans ce projet, nous nous concentrerons sur les vectoriseurs suivants :

- CountVectorizer :

CountVectorizer du modèle *sklearn.feature_extraction*, a ses propres méthodes internes de tokenisation et de normalisation. La méthode d'ajustement du vectoriseur attend une itérable ou une liste de chaînes ou d'objets de document, et crée un dictionnaire du vocabulaire sur le corpus. Lorsque *transform* est appelé, CountVectorizer prend tous les mots dans tous les tweets, attribue un ID et compte la fréquence du mot par tweet. Nous utilisons ensuite ce sac de mots comme entrée pour le classifieur. Ce sac de mots est un ensemble de données clairsemées. Cela signifie que chaque enregistrement aura de nombreux zéros pour les mots ne figurant pas dans le tweet. Figure 4.3-1 illustre une démonstration de CountVectorizer.

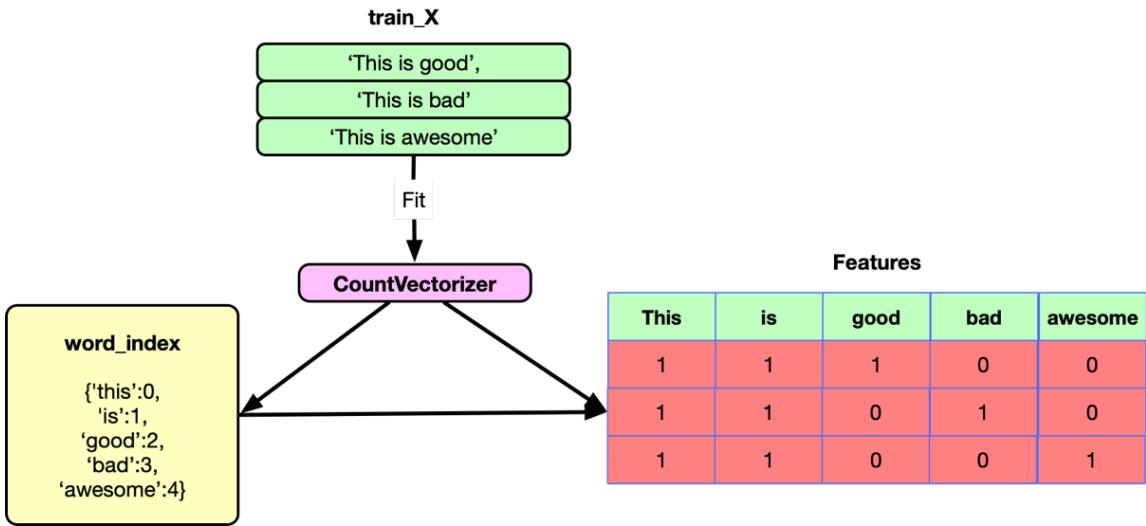


Figure 4.3-1 Démonstration de CountVectorizer

- TF-IDF Vectorizer :

Scikit-Learn fournit `TfidfVectorizer` dans le module appelé `feature_extraction.text` pour vectoriser des documents avec des scores TF-IDF. Sous le capot, le `TfidfVectorizer` utilise l'estimateur `CountVectorizer` que nous avons utilisé pour produire le codage en bag-of-words pour compter les occurrences de tokens, suivi d'un `TfidfTransformer`, qui normalise ces nombres d'occurrences par la fréquence inverse du document. L'entrée d'un `TfidfVectorizer` est censée être une séquence de noms de fichiers, d'objets de type fichier ou de chaînes qui contiennent une collection de documents bruts, similaire à celle du `CountVectorizer`. Par conséquent, une méthode de tokenisation et de prétraitement par défaut est appliquée, sauf si d'autres fonctions sont spécifiées. Le vectoriseur renvoie une représentation matricielle clairsemée sous la forme de ((doc, terme), tfidf) où chaque clé est un document, une paire de termes et le score TF – IDF. Figure 4.3-2 illustre une démonstration de `TfidfVectorizer`.

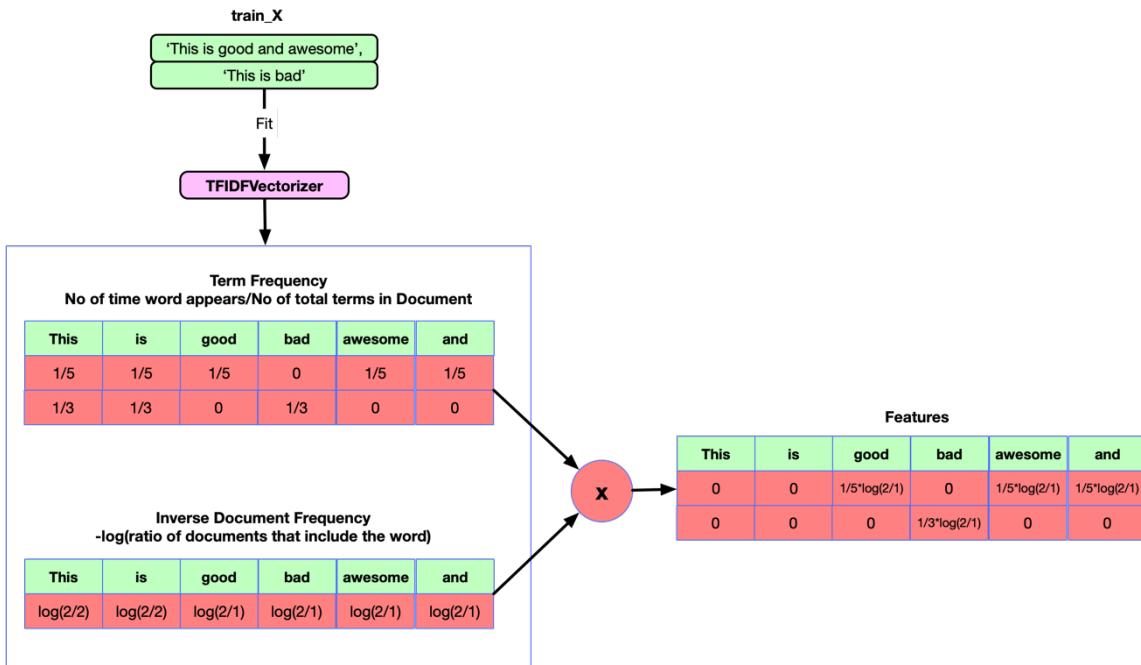


Figure 4.3-2 Démonstration TF-IDFVectorizer

Dans notre cas, on fait appel à `TfidfVectorizer` pour alléger les mots qui apparaissent fréquemment dans `CountVectorizer` et qui peuvent ne pas contenir d'informations discriminatoires.

4.3.2 Crédation de données d'apprentissage/test

Dans les statistiques et Machine Learning, nous divisons généralement nos données en deux sous-ensembles: les données d'apprentissage et les données de test (et parfois à trois: apprentissage, validation et test), et adapter notre modèle aux données d'apprentissage, afin de faire des prédictions sur les données de test. L'ensemble d'apprentissage contient une sortie connue et le modèle apprend sur ces données afin d'être généralisé à d'autres données ultérieurement. L'ensemble de test fournit l'étalement-or utilisé pour évaluer le modèle. Il n'est utilisé qu'une fois qu'un modèle est complètement formé.

Nous le ferons en utilisant la bibliothèque Scikit-Learn et en particulier la méthode `train_test_split`. Cela dépend principalement de 2 choses. Premièrement, le nombre total d'échantillons dans nos données et deuxièmement, sur les modèles que nous formons.



Figure 4.3-3 Représentation de Train/Test Split

4.4 Construction des modèles

Dans ce projet nous allons se concentrer sur la construction de 2 modèles. le premier pour l'analyse des sentiments au niveau des tweets sur Twitter pour arbitrer si le message est de sentiment positif, négatif ou neutre. Dans ce modèle nous allons utiliser le dataset *crowdflower-airline-twitter-sentiment*. Le deuxième pour la détection des raisons de négativité des tweets où nous allons utiliser le dataset des raisons que nous avons reconstruit.

4.4.1 Cross-validation

Pour choisir les meilleurs paramètres, nous devons tester sur un ensemble de validation distinct. Ce dernier n'a pas été utilisé durant l'apprentissage. Pourtant, l'utilisation d'un seul ensemble de validation risque de ne pas produire des résultats de validation fiables.

Cross-validation ou k-fold cross-validation signifie que le dataset est divisé au hasard en k groupes. L'un des groupes est utilisé comme ensemble de test et les autres sont utilisés comme ensemble d'apprentissage. Le modèle est formé sur l'ensemble d'entraînement et noté sur l'ensemble de test. Ensuite, le processus est répété jusqu'à ce que chaque groupe unique soit utilisé comme ensemble de test. Dans notre cas, nous avons d'abord divisé dataset en 2 - train et test. Après cela, nous avons mis de côté l'ensemble de test et choisi au hasard X% de l'ensemble de données train pour être l'ensemble de train réel et le% (100-X) restant pour l'ensemble de validation, où X est un nombre fixe, le modèle est ensuite formé itérativement et validés sur ces différents ensembles.

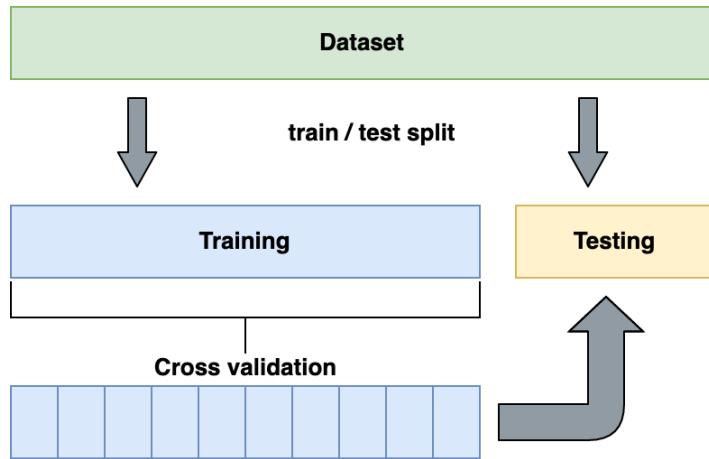


Figure 4.4-1 Représentation de Train/Test Split et Cross Validation

4.4.2 Pipelines

Scikit-learn fournit un utilitaire pipeline pour aider à automatiser les workflows de Machine Learning.

Les pipelines fonctionnent en permettant l'enchaînement d'une séquence linéaire de transformations de données aboutissant à un processus de modélisation qui peut être évalué. L'objectif est de garantir que toutes les étapes sont limitées aux données disponibles pour l'évaluation, telles que l'ensemble de données d'apprentissage ou chaque fold de la procédure cross validation. Les pipelines fournissent également une interface unique pour Gridsearch de plusieurs estimateurs à la fois. Plus important encore, les pipelines fournissent une opérationnalisation des modèles de texte en couplant les méthodologies de vectorisation avec un modèle prédictif. Les pipelines sont construits en décrivant une liste de paires (clé, valeur) où la clé est une chaîne qui nomme l'étape et la valeur est l'objet estimateur.

Les pipelines peuvent être arbitrairement complexes par la mise en œuvre de FeatureUnions. L'objet FeatureUnion combine plusieurs objets transformateur en un nouveau transformateur unique similaire à l'objet Pipeline. Cependant, au lieu d'ajuster et de transformer les données en séquence à travers chaque transformateur, elles sont plutôt évaluées indépendamment et les résultats sont concaténés dans un vecteur composite. Les objets FeatureUnion sont instanciés de la même manière que les objets Pipeline avec une liste de paires (clé, valeur) où la clé est le nom du transformateur et la valeur est l'objet transformateur.

4.4.3 Hyperparameters

Hyperparameters sont les propriétés qui régissent l'ensemble du processus de l'apprentissage. Cette configuration est externe au modèle et dont les valeurs ne peuvent être estimées à partir des données. Nous ne pouvons pas connaître les meilleures valeurs sur un problème donné. Nous pouvons utiliser des règles empiriques ou bien rechercher par essais et erreurs.

Hyperparameters sont importants puisqu'ils contrôlent directement le comportement de l'algorithme de surplus ils ont un impact significatif sur les performances du modèle en cours d'apprentissage.

Le processus de recherche des Hyperparameters les plus optimaux dans Machine Learning est appelé optimisation des Hyperparameters. Dans notre projet nous allons utiliser GridSearchCV du *sklearn.model_selection* comme moyen pour optimiser ces Hyperparameters.

Gridsearch est une méthode traditionnelle pour effectuer une optimisation des Hyperparameters. Elle nécessite de créer deux ensembles, Learning Rate et Number of Layers. Gridsearch forme l'algorithme pour toutes les combinaisons en utilisant les deux ensembles et mesure les performances en utilisant la technique de Cross Validation. La figure 4.4-2 illustre le processus de cross-validation de 5-fold.

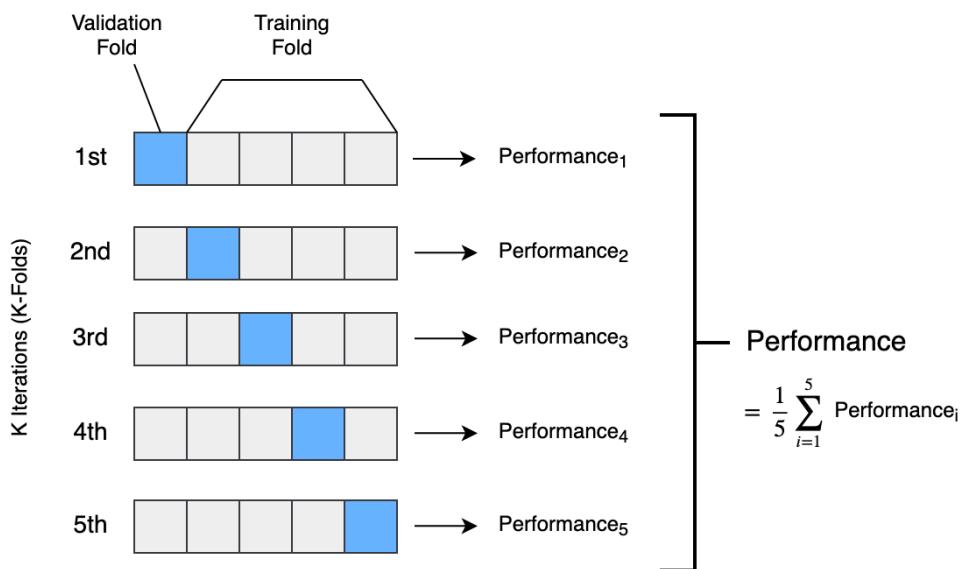


Figure 4.4-2 Le processus de cross-validation de 5-fold

Les valeurs de précision de l'apprentissage et de la validation donnent un aperçu sur la performance du modèle. Ce processus est répété «k» fois et chaque fois qu'un fold est choisi pour être utilisé comme ensemble de

validation. Ainsi, «k» modèles sont entraînés et la précision est la moyenne des «k» modèles.

Le but de Gridsearch est de trouver le meilleur tuple dans l'espace hyperparamétrique.

- La première étape consiste à créer tous les tuples possibles.
- Ensuite, un tuple est choisi et utilisé pour définir les valeurs d'hyper-paramètre du modèle. Maintenant, pour cet ensemble de valeurs, une k-fold cross est effectuée.
- Une fois que tous les k-folds ont été utilisés, la précision du modèle est enregistrée.
- Ensuite, un autre tuple est choisi pour définir les valeurs d'hyper-paramètre dans le modèle et la même procédure est effectuée jusqu'à ce que tous les tuples aient été utilisés.
- À la fin, le tuple qui donne la meilleure précision est choisi.

L'avantage de Gridsearch est qu'elle est hautement parallélisable puisque nous pouvons potentiellement évaluer les performances de chaque combinaison d'hyper-paramètre sur différentes machines en parallèle. Ceci est possible car les différentes combinaisons d'hyper-paramètres sont indépendantes les unes des autres.

Dans GridSearchCV, nous avons utilisé l'ensemble des paramètres utilisés par *Bert Carremans* pour tester les performances.

- Paramètres de Gridsearch pour les vectoriseurs (Count et TFIDF) :

- max_df : (0.25, 0.5, 0.75)

Lors de la construction du vocabulaire, ce paramètre ignore les termes qui ont une fréquence de document strictement supérieure au seuil donné (corpus-specific stop words). Le paramètre représente une proportion de documents.

- min_df : (1, 2, 3, 4)

Lors de la construction du vocabulaire, ce paramètre ignore les termes qui ont une fréquence de document strictement inférieure au seuil donné. Cette valeur est également appelée cut-off dans

la littérature. Le paramètre représente une proportion de documents.

- ngram_range : ((1, 1), (1, 2), (1, 3))

Les ngram sont simplement toutes les combinaisons de mots ou de lettres adjacentes de longueur n que vous pouvez trouver dans votre texte source. le paramètre ngram_range = (a, b) où a est le minimum et b est la taille maximale des ngram, Par exemple, un ngram_range de (1, 1) signifie uniquement des unigrammes, (1, 2) signifie des unigrammes et des bigrammes et (2, 2) signifie uniquement des bigrammes.

- Paramètres de Gridsearch pour MultinomialNB :

- alpha : (0.2, 0.25, 0.3, 0.4, 0.5, 0.75)

Paramètre de lissage additif (Laplace / Lidstone) qui est utile pour éviter une probabilité nulle pour les features rares (une probabilité nulle exclurait ces features de l'analyse).

- Paramètres de Gridsearch pour LogisticRegression :

- C : (0.25, 0.5, 1.0)

Paramètre de régularisation inverse - La régularisation λ est un moyen de trouver un bon compromis biais-variance en ajustant la complexité du modèle. Il s'agit d'une méthode très utile pour gérer la colinéarité (forte corrélation entre les entités), filtrer le bruit des données et éventuellement empêcher le sur-ajustement. Le paramètre $C = 1/\lambda$ fonctionnera dans l'autre sens. Pour les petites valeurs de C, nous augmentons la force de régularisation qui créera des modèles simples qui sous-tendent les données. Pour les grandes valeurs de C, nous diminuons la puissance de régularisation, ce qui implique que le modèle est autorisé à augmenter sa complexité et, par conséquent, à surcharger les données.

- penalty : ('l1', 'l2', 'elasticnet')

Utilisé pour spécifier la norme utilisée dans la pénalisation. La pénalisation évite le overfitting en ne générant pas de coefficients élevés pour les prédicteurs rares et stabilise les estimations, en particulier lorsqu'il y a colinéarité dans les données. Un modèle de régression qui utilise la technique de régularisation L1 est appelé régression Lasso et le modèle qui utilise L2 est appelé régression Ridge. La principale différence entre ces techniques est que Lasso réduit ainsi le coefficient de la caractéristique la moins importante à zéro, supprimant ainsi une certaine caractéristique. Elasticnet est une combinaison de L1 et L2.

4.5 Évaluation et résultats

4.5.1 Mesures d'évaluation

Selon [McLaughlin et Herlocker](#), les algorithmes peuvent être mesurés en utilisant plusieurs méthodes d'évaluation. Les plus connus et simples à utiliser sont Precision, Recall et F-score.

On pose :

- **condition positive (P)** : le nombre de cas positifs réels dans les données.
 - vrai positif (VP)
 - faux positif (FP)
- **condition négative (N)** : le nombre de cas négatifs réels dans les données.
 - vrai négatif (VN)
 - faux négatif (FN)

la précision (ou valeur prédictive positive) est la proportion des items pertinents parmi l'ensemble des items proposés, elle permet de répondre à la question suivante : Quelle proportion d'identifications positives était effectivement correcte ?

$$Précision = \frac{VP}{VP + FP}$$

le rappel (ou sensibilité) est la proportion des items pertinents proposés parmi l'ensemble des items pertinents, il permet de répondre à la question suivante : Quelle proportion de résultats positifs réels a été identifiée correctement ?

$$Recall = \frac{VP}{VP + FN}$$

Accuracy est une mesure d'évaluation qui nous permet de mesurer le nombre total de prédictions que le modèle obtient correctement.

$$Accuracy = \frac{VP + VN}{VP + VN + FP + FN}$$

F-score est la moyenne harmonique de la précision et du rappel, elle donne une meilleure mesure des cas mal classés que la métrique de précision.

$$Fscore = 2 * \frac{Precision * Recall}{Precision + Recall}$$

	CountVectorizer				TfidfVectorizer			
	precision	recall	f-score	accuracy	precision	recall	f-score	accuracy
MultinomialNB	0.79	0.80	0.79	0.80	0.77	0.77	0.74	0.77
LogisticRegression	0.80	0.81	0.80	0.81	0.79	0.78	0.79	0.79

Tableau 4.5-1 Analyse de la performance du modèle de sentiment

D'après le tableau 4.5-1 ci-dessus, les deux classificateurs obtiennent les meilleurs résultats lors de l'utilisation des fonctionnalités de CountVectorizer. Les meilleures performances sur l'ensemble de test proviennent de LogisticRegression avec des fonctionnalités de CountVectorizer.

Meilleurs paramètres:

```

clf_C: 0.5
clf_max_iter: 200
clf_penalty: 'l2'
features_pipe_vect_max_df: 0.25
features_pipe_vect_min_df: 1
features_pipe_vect_ngram_range: (1, 2)

Train score with best_estimator_: 0.953
Test score with best_estimator_: 0.807

```

	CountVectorizer				TfidfVectorizer			
	precision	recall	f-score	accuracy	precision	recall	f-score	accuracy
MultinomialNB	0.80	0.80	0.80	0.80	0.76	0.77	0.75	0.77
LogisticRegression	0.78	0.78	0.78	0.78	0.78	0.77	0.76	0.77

Tableau 4.5-2 Analyse de la performance du modèle d'aspect

D'après le tableau 4.5-2 ci-dessus, les deux classificateurs obtiennent les meilleurs résultats lors de l'utilisation des fonctionnalités de CountVectorizer. Les meilleures performances sur l'ensemble de test proviennent de MultinomialNB avec des fonctionnalités de CountVectorizer.

Meilleurs paramètres:

```

clf_alpha: 0.3
features_pipe_vect_max_df: 0.5
features_pipe_vect_min_df: 4
features_pipe_vect_ngram_range: (1, 2)

Test score with best_estimator_: 0.796
Train score with best_estimator_: 0.872

```

4.5.2 Discussion

MultinomialNB et LogisticRegression sont des modèles log-linéaires; c'est-à-dire que, dans les deux cas, la probabilité qu'un document appartienne à une classe est proportionnelle à $e^{(w \cdot x)}$, où w est un paramètre classifieur et x est un vecteur caractéristique du document. La principale différence est

que, dans MultinomialNB, le modèle est spécifié de sorte que les données et les étiquettes dépendent de w , tandis que dans LogisticRegression, seules les étiquettes dépendent de w .

En bref, MultinomialNB a un biais plus élevé mais une variance inférieure par rapport à LogisticRegression. LogisticRegression fait une prédiction de la probabilité en utilisant une forme fonctionnelle directe où MultinomialNB détermine comment les données ont été générées compte tenu des résultats.

avant de choisir LogisticRegression et MultinomialNB comme modèles principaux, l'analyse des sentiments a été effectuée au moyen d'une comparaison avec divers algorithmes d'ensemble dans lesquels accuracy a été obtenue.

Accuracy of KNeighborsClassifier	0.589139344262
Accuracy of SVC	0.645150273224
Accuracy of DecisionTreeClassifier	0.758879781421
Accuracy of AdaBoostClassifier	0.785519125683
Accuracy of GaussianNB	0.572404371585

La méthode de prétraitement choisie est discutée dans de nombreux articles; les effets de la méthode de prétraitement du texte sur les performances de classification des sentiments dans deux types de tâches de classification, a résumé les performances de classification de six méthodes de prétraitement à l'aide de deux modèles d'entités et de quatre classifieurs sur cinq jeux de données Twitter. Les expériences montrent que la précision et la mesure F-score du classifieur de classification des sentiments de Twitter sont améliorées lors de l'utilisation des méthodes de prétraitement de suppression de stopwords, stemming..., mais changent à peine lors de la suppression des URL, de la suppression des nombres ou de la ponctuation. Le prétraitement des données réduit le bruit dans le texte et devrait contribuer à améliorer les performances du classifieur et à accélérer le processus de classification. Les résultats expérimentaux montrent que la suppression des URL affecte à peine les performances des classifieurs dans les deux modèles d'entités sur tous les jeux de données. Cela indique que les URL ne contiennent pas d'informations utiles pour la classification des sentiments. il y a également peu d'effets sur les performances des classifieurs dans le modèle N-grammes avant et après la suppression des stopwords. L'une des raisons pourrait être

que les mots vides apparaissent fréquemment dans les tweets. la suppression des mots vides entraîne une fluctuation des performances du classifieur, car les mots vides contiennent une polarité de sentiment différente, ce qui signifie qu'il est nécessaire de supprimer les mots vides pour la classification des sentiments. la suppression des nombres n'a aucun effet sur la précision de la classification des sentiments car les nombres sont neutres.

4.6 Visualisation des résultats

4.6.1 L'interface

Comme évoqué précédemment, nous avons utilisé le Framework Flask afin de réaliser l'interface. Flask est construit sur le pattern architecturale MVT (Model-View-Template), issue principalement du célèbre MVC(Model-View-Controller), il aide à créer des applications Web complexes basés sur des bases de données. Celui-ci nous a permis de facilement naviguer entre la partie back-end et la partie front-end afin de faciliter la visualisation par l'utilisateur final.

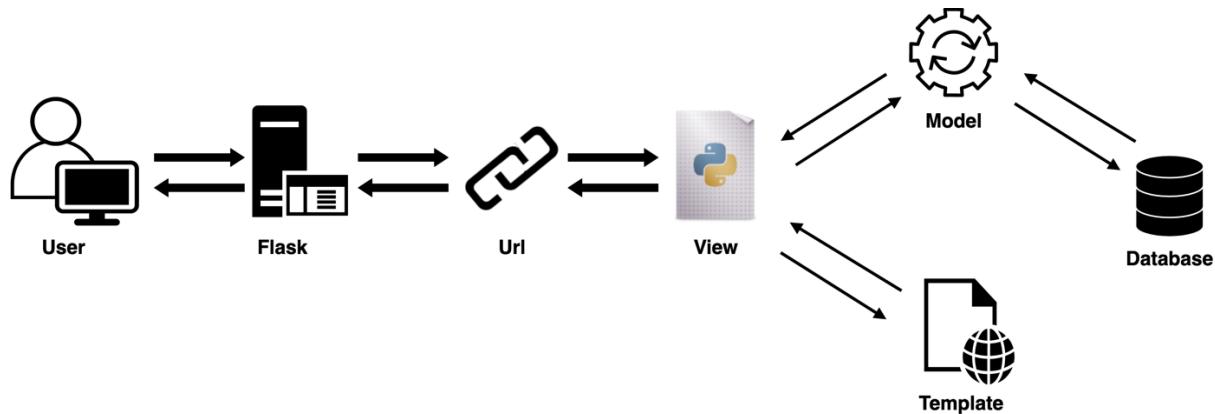


Figure 4.6-1 Modèle MVT(Model-View-Template)

4.6.2 Flux de processus

Il s'agit d'une application Web qui peut être utilisée pour analyser les sentiments des utilisateurs à travers les tweets en se basant sur les aspects. Elle est créée à l'aide de Flask et utilise un modèle LogisticRegression formé sur l'ensemble de données Kaggle *crowdflower-airline-twitter-sentiment* et servi comme REST API.

Le serveur extrait les tweets à l'aide de Tweepy et effectue l'analyse des sentiments à l'aide du modèle. Il extrait également des données de l'API Wikipédia en fonction de la compagnie aérienne sélectionnée pour afficher une brève description, il utilise également l'API OpenStreetMap pour obtenir l'emplacement exact de chaque tweet et son origine. Pour faciliter la recherche nous avons collecté tous les noms des compagnie aériennes afin d'éviter les erreurs d'orthographe et pour faciliter la recherche à l'utilisateur. Après que le modèle effectue ces tâches, nous trions et organisons toutes les données dans une base de données MySQL. La figure 4.6-2 résume le flux de travail de l'application.

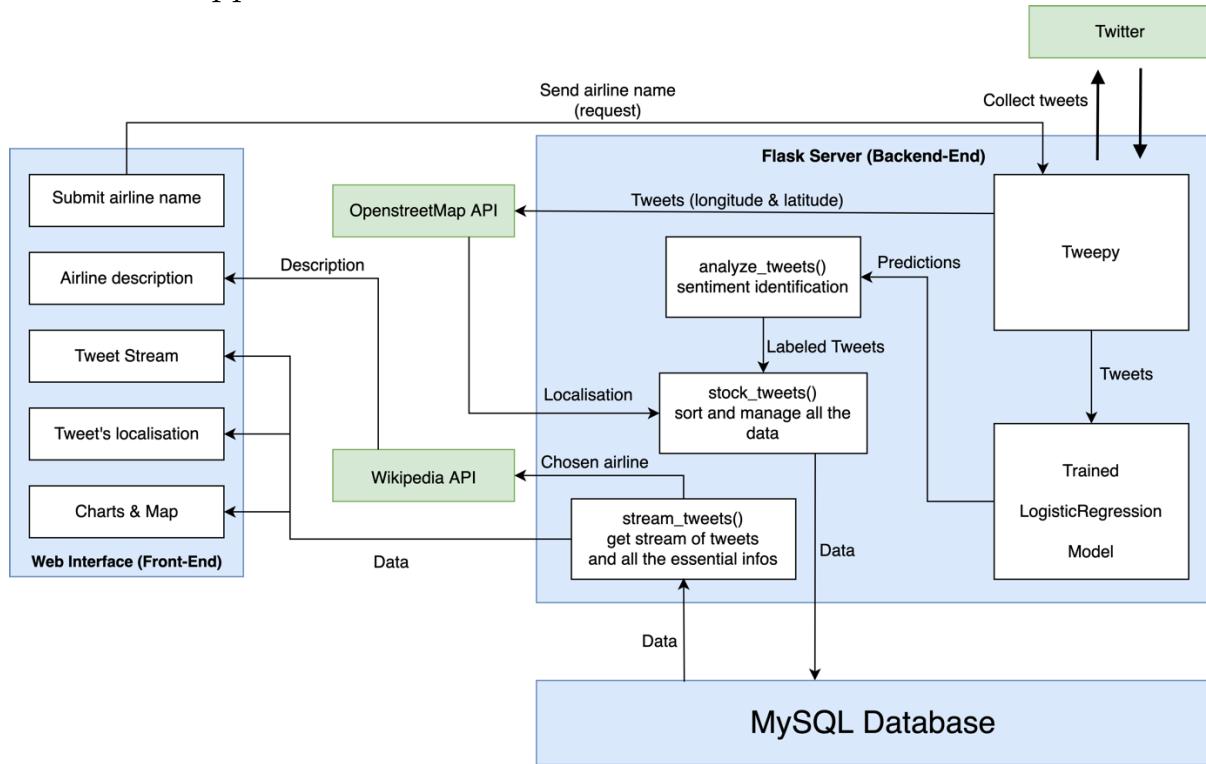


Figure 4.6-2 Organigramme de l'application

4.6.3 Visualisation

Ayant comme objectif de visualiser les résultats générés par le modèle et pour faciliter l'analyse de ces résultats , nous avons opté pour l'utilisation d'un outil développé en langage Python nommé Plotly. Celui-ci permet de visualiser les données issue de la base de données dans une page web HTML. L'utilisation consiste en l'établissement d'un lien entre le code Python et le code JavaScript à l'aide de Jinja.

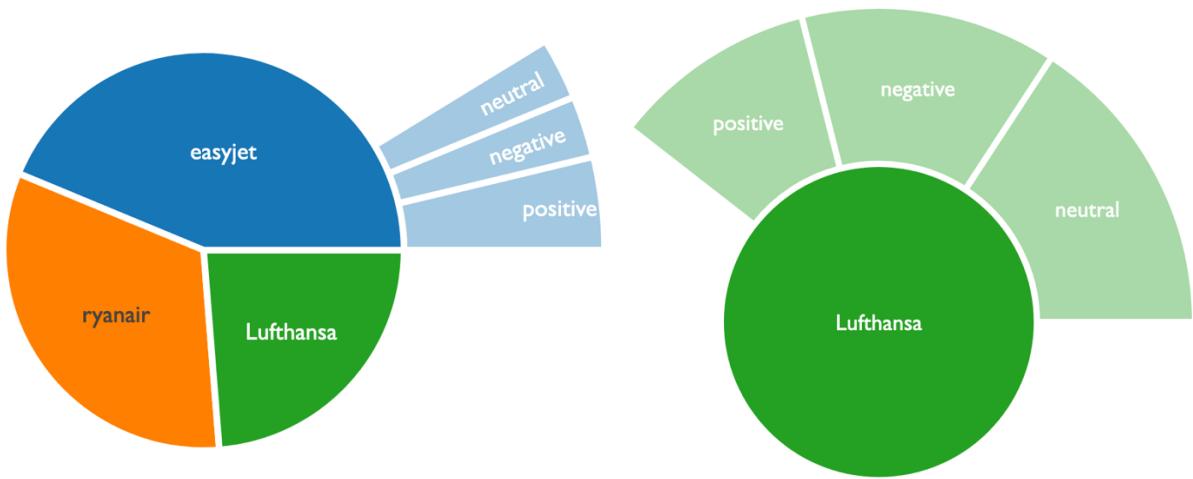


Figure 4.6-3 Distribution des sentiments pour chaque compagnie aérienne

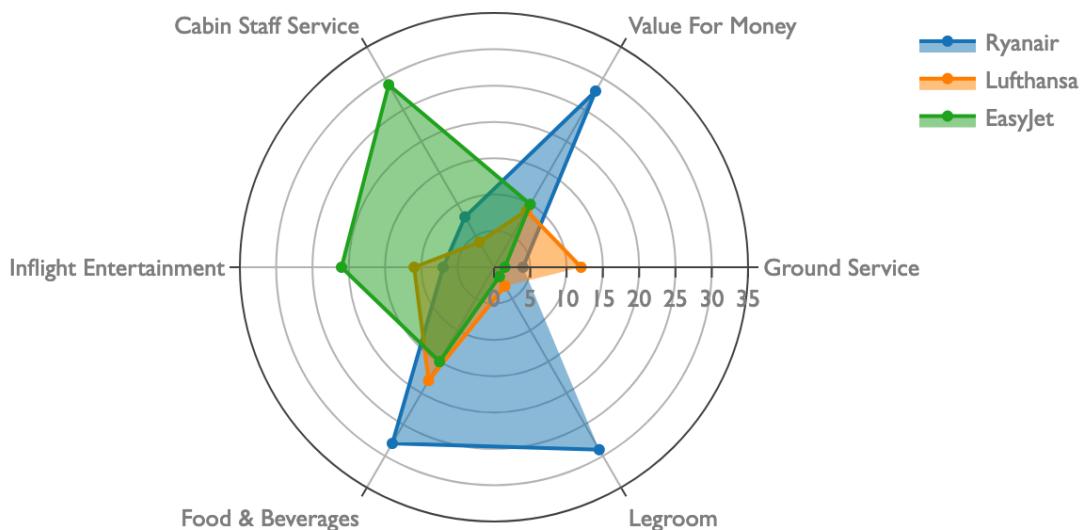


Figure 4.6-4 Les aspects reconnus pour chaque compagnie aérienne et leurs intensité

Les tweets collectés offrent une multitude de données pouvant être traitées, dont la localisation de l'endroit de la publication du tweet. A l'aide de l'API OpenStreetMap et MapBox, nous avons pu situer les pays dans une carte du monde en créant une liste pour les pays évoqué dans les tweets. La construction de cette carte consiste en la transmission des données relatifs à la localisation vers le script. L'un des problèmes rencontrés dans la réalisation de cette tâche était l'absence de la localisation pour certains tweets, et l'incohérence de celle-ci dans d'autres.



Figure 4.6-5 Visualisation de la Carte du monde

5. Conclusion

Ce projet est axé autour de sur l'industrie du transport aérien, notamment en utilisant une analyse des sentiments basée sur les aspect. Une contribution significative va apporter cette recherche aux clients pour décider des compagnies aériennes et contribuer également à combler l'écart entre les transporteurs et les clients, la précision des résultats incite les clients à comprendre et à choisir les meilleurs transporteurs aériens en fonction du modèle. Cette recherche aide grandement les compagnies aériennes à rechercher les aspects à améliorer et pourrait facilement comparer leurs performances avec leurs concurrents pour obtenir un meilleur avantage concurrentiel sur le marché. Dans ce projet, il y a aussi des lacunes. Le modèle utilisé pour cette recherche peut être appliqué uniquement pour la langue anglaise. Puisqu'il y aura une structure grammaticale distincte pour différentes langues. Ainsi, pour une autre langue, ce modèle ne fonctionnera pas pour d'autres langues.

Enfin, nous concluons que l'analyse des sentiments sur Twitter est un outil très utile pour la recherche auprès des consommateurs, en particulier dans les secteurs où les clients passent leur temps sur les réseaux sociaux. Moins nous collectons de données sur les réseaux sociaux, plus nous subirons d'erreurs du résultat analysé.

5.1 Apports et gains

Notre projet de fin d'étude a été pour nous non seulement une étape à franchir pour achever notre parcours académique, mais aussi une opportunité en or de prendre part, à un projet réel qui s'inscrit dans le cadre de l'intelligence artificiel.

La réalisation de ce travail a été pour nous une occasion pour mettre en pratique les connaissances théoriques acquises durant notre parcours universitaire. De ce fait, plusieurs modules étudiés, tout au long de notre formation de licence en Sciences Mathématiques et Informatique ont été impliqués dans ce projet, à compter : l'algorithme, la gestion des bases de données, La conception de systèmes d'information, la programmation orientée objet, la programmation et technologie du Web et finalement Probabilités et Statistique. Par conséquent, ce stage que nous avons effectué

nous a donné l'occasion de tisser un lien entre nos connaissances académique et le monde professionnel. D'une part, ils nous ont permis de développer nos compétences techniques, d'approfondir nos connaissances théoriques et pratiques et de stimuler notre créativité.

Concernant l'aspect des connaissances, le tableau ci-dessous ressemble les différents outils, langages et technologies utilisées lors de l'élaboration de chaque tâche du projet.

Tâche	Outil - Technologie	Description
Collecte des données	Tweepy	Une bibliothèque Python facile à utiliser pour accéder à l'API Twitter.
	Twitter Developer API	Fournit un large accès aux données Twitter publiques que les utilisateurs ont choisi de partager avec le monde.
	OpenStreetMap API	Un service web pour récupérer et enregistrer des géodonnées brutes de / vers la base de données OpenStreetMap.
	Wikipédia API	Un service web qui permet d'accéder à certaines fonctionnalités wiki.
Analyse et Traitement des sentiments	Python	Un langage de programmation interprété, multi-paradigme et multiplateformes.
	NLTK	Une bibliothèque Python permettant un traitement automatique des langues
	scikit-learn	Une bibliothèque libre Python destinée à l'apprentissage automatique.
Apprentissage du modèle	Google Colab	un service cloud, offert par Google, basé sur Jupyter Notebook et destiné à la formation et à la recherche dans l'apprentissage automatique.
Interface	Flask	Un Framework open-source de développement web en Python.
Visualisation	Plotly	Une bibliothèque graphique crée des graphiques interactifs de qualité publication.

5.2 Difficultés

Parmi les difficultés majeures qu'on a eu pendant la réalisation de cette solution :

- La difficulté d'avoir accès à l'API Twitter, en comparaison avec les autres plateformes (YouTube par exemple) : communication en mails avec le service développeur afin d'avoir des clés d'accès.
- Les tweets collectées ne sont pas toujours valides, parfois on trouve des tweets d'actualité des compagnies aériennes ou des évènements organisé où ces compagnies aériennes sont des sponsors.
- La difficulté de choisir un modèle approprié et de le configurer par les paramètres convenables.
- Lors de la phase de prétraitement des données, nous avons eu des difficultés pour convertir les emojis et les émoticônes puisqu'il sont nécessaires pour la prédiction du sentiment.
- Lors de la phase de prédiction, nous avons été confrontés à un problème de formats : Les fichiers de sorties JSON collectés grâce à l'API ne sont pas adaptés au passage direct au modèle.

5.3 Perspectives

Dans le but de rendre ce projet plus puissant/fiable, nous avons pensé à plusieurs ajouts pouvant contribuer par la suite à la solution d'être beaucoup plus précise :

- La détection de sujet de sentiment peut être employée pour déterminer le sujet le plus discuté.
- Améliorer le modèle afin de gérer les tweets contenant plusieurs polarités dans le même post, et de gérer plusieurs négations dans le même post.
- Ajouter une détection d'opinion fausse pour garantir la crédibilité des tweets.
- Classer les tweets par sentiment positif et négatif, mais plutôt par une échelle plus granulaire avec des émotions allant de «triste» à «en colère» et plus encore.
- La mise en place d'un système multilingue capable de classer les sentiments exprimés dans différentes langues.

- Concevoir un algorithme d'apprentissage automatique qui sépare les déclarations ironiques, sarcastique et d'humour.
- L'intégration d'autres plateformes/réseaux sociaux (Facebook, Instagram...) peut mener vers des résultats plus complets et pertinents en réalisant des intersections entre les données obtenues.
- Création d'un dictionnaire des sentiments d'argot pour faciliter l'analyse des sentiments du contenu des médias sociaux.

Références

- Anuja Nagpal (2017). L1 and L2 Regularization Methods. [online] Medium. Available at: <https://towardsdatascience.com/l1-and-l2-regularization-methods-ce25e7fc831c>
- McLaughlin, M.R. and Herlocker, J.L. (2004). A collaborative filtering algorithm and evaluation metric that accurately model the user experience. Proceedings of the 27th annual international conference on Research and development in information retrieval - SIGIR '04.
- AYLIEN. (2017). Understanding Customer Frustrations in the Airline Industry with Aspect-based Sentiment Analysis. [online] Available at: <https://blog.aylien.com/understanding-customer-frustrations-in-the-airline-industry-with-aspect-based-sentiment-analysis/>
- Barbieri, F. and Saggion, H. (n.d.). Modelling Irony in Twitter: Feature Analysis and Evaluation. [online] Available at: http://www.lrec-conf.org/proceedings/lrec2014/pdf/231_Paper.pdf
- Bharti, S.K., Vachha, B., Pradhan, R.K., Babu, K.S. and Jena, S.K. (2016). Sarcastic sentiment detection in tweets streamed in real time: a big data approach. Digital Communications and Networks, [online] 2(3), pp.108–121. Available at: <https://www.sciencedirect.com/science/article/pii/S235286481630027X>
- Carremans, B. (2018). Sentiment Analysis with Text Mining. [online] Towards Data Science. Available at: <https://towardsdatascience.com/sentiment-analysis-with-text-mining-13dd2b33de27>
- Chamlertwat, W., Bhattarakosol, P., Rungkasiri, T. and Haruechaiyasak, C. (n.d.). Discovering Consumer Insight from Twitter via Sentiment Analysis. [online] Available at: http://jucs.org/jucs_18_8/discovering_consumer_insight_from/jucs_18_08_0973_0992_chamertwat.pdf
- data.worldbank.org. (n.d.). Air transport, passengers carried - Morocco | Data. [online] Available at: <https://data.worldbank.org/indicator/IS.AIR.PSGR?end=2018&locations=MA&start=2000&type=points&view=chart&year=2018>
- DBD, R. (2019). KDD Process/Overview. [online] Uregina.ca. Available at: http://www2.cs.uregina.ca/~dbd/cs831/notes/kdd/1_kdd.html
- Deb, S. (2016). Naive Bayes vs Logistic Regression. [online] Medium. Available at: https://medium.com/@sangha_deb/naive-bayes-vs-logistic-regression-a319b07a5d4c
- DEI, M. (2019). Hyperparameter Tuning Explained — Tuning Phases, Tuning Methods, Bayesian Optim, and Sample Code! [online] Medium. Available at: <https://towardsdatascience.com/hyperparameter-tuning-explained-d0ebb2ba1d35>

Federici, M. and Dragoni, M. (n.d.). A Branching Strategy For Unsupervised Aspect-based Sentiment Analysis. [online] Available at: http://ceur-ws.org/Vol-1874/paper_6.pdf

Hale, J. (2020). Don't Sweat the Solver Stuff. [online] Medium. Available at: <https://towardsdatascience.com/dont-sweat-the-solver-stuff-aea7cddc3451>

Joshi, S. (2017). Aspect based sentiment analysis for United States of America Airlines. [online] trap.ncirl.ie. Available at: <http://trap.ncirl.ie/3077/>

Lee, E. (2019). An Intro to Hyper-parameter Optimization using Grid Search and Random Search. [online] Medium. Available at: <https://medium.com/@cj12fv/an-intro-to-hyper-parameter-optimization-using-grid-search-and-random-search-d73b9834ca0a>

Lim, F. (2019). Twitter U.S. Airline Sentiment Analysis using Keras and RNNs. [online] Medium. Available at: https://medium.com/@francesca_lim/twitter-u-s-airline-sentiment-analysis-using-keras-and-rnns-1956f42294ef

Prabhakar, E., Santhosh, M., Krishnan, A., Kumar, T. and Sudhakar B B Student, R. (n.d.). Sentiment Analysis of US Airline Twitter Data using New Adaboost Approach. [online] Available at: <https://www.ijert.org/research/sentiment-analysis-of-us-airline-twitter-data-using-new-adaboost-approach-IJERTCONV7IS01003.pdf>

Reddy, K. mohan (2018). Model Selection Using Cross Validation and GridSearchCV. [online] Medium. Available at: <https://medium.com/@kesarimohan87/model-selection-using-cross-validation-and-gridsearchcv-8756aac1e9d7>

Saif, H., He, Y. and Alani, H. (n.d.). Semantic Sentiment Analysis of Twitter. [online] Available at: https://link.springer.com/content/pdf/10.1007%2F978-3-642-35176-1_32.pdf

Sanjay.M (2018). Why and how to Cross Validate a Model? [online] Medium. Available at: <https://towardsdatascience.com/why-and-how-to-cross-validate-a-model-d6424b45261f>

Sarang Narkhede (2018). Understanding Confusion Matrix. [online] Medium. Available at: <https://towardsdatascience.com/understanding-confusion-matrix-a9ad42dcfd62>

Stack Overflow. (n.d.). python - Multinomial Naive Bayes parameter alpha setting? scikit-learn. [online] Available at: <https://stackoverflow.com/questions/33830959/multinomial-naive-bayes-parameter-alpha-setting-scikit-learn>

Stack Overflow. (n.d.). python - Understanding min_df and max_df in scikit CountVectorizer. [online] Available at: <https://stackoverflow.com/questions/27697766/understanding-min-df-and-max-df-in-scikit-countvectorizer>

oreilly. (n.d.). 4. Text Vectorization and Transformation Pipelines - Applied Text Analysis with Python [Book]. [online] Available at: <https://www.oreilly.com/library/view/applied-text-analysis/9781491963036/ch04.html>

Xavier, A., Mohan, V., Sree, H. and Venu (n.d.). FULL PAPER PROCEEDING Multidisciplinary Studies-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>) Peer-review under responsibility of the Scientific & Review committee of Sentiment Analysis Applied to Airline Feedback to Boost Customer's Endearment-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>) Peer-review under responsibility of the Scientific & Review committee of MISG-2015. [online] 2, pp.219–232. Available at: <https://www.globalilluminators.org/wp-content/uploads/2015/12/MISG-15-198.pdf>

Agarwal, A., Xie, B., Vovsha, I., Rambow, O. and Passonneau, R. (2011). Sentiment Analysis of Twitter Data. [online] Association for Computational Linguistics, pp.30–38. Available at: <https://www.aclweb.org/anthology/W11-0705.pdf>

Goel, A., Gautam, J. and Kumar, S. (2016). Real time sentiment analysis of tweets using Naive Bayes. [online] IEEE Xplore. Available at: <https://ieeexplore.ieee.org/abstract/document/7877424>

Kumar, A. and Sebastian, T. (n.d.). Sentiment Analysis on Twitter. [online] Available at: https://acl-arc.comp.nus.edu.sg/~wing.nus/sig/papers_nlp/20120831_1.pdf

Jianqiang, Z. and Xiaolin, G. (2017). Comparison Research on Text Pre-processing Methods on Twitter Sentiment Analysis. IEEE Access, 5, pp.2870–2879

Chavan, A. (2018). Logistic Regression. [online] Medium. Available at: <https://medium.com/@akshayc123/logistic-regression-87f7fbb4aaf6>

T, S. (2019). Custom Transformers and ML Data Pipelines with Python. [online] Medium. Available at: <https://towardsdatascience.com/custom-transformers-and-ml-data-pipelines-with-python-20ea2a7adb65>

Wu, L., Morstatter, F. and Liu, H. (2018). SlangSD: building, expanding and using a sentiment dictionary of slang words for short-text sentiment classification. Language Resources and Evaluation, [online] 52(3), pp.839–852. Available at: <https://arxiv.org/pdf/1608.05129.pdf>

Annexes

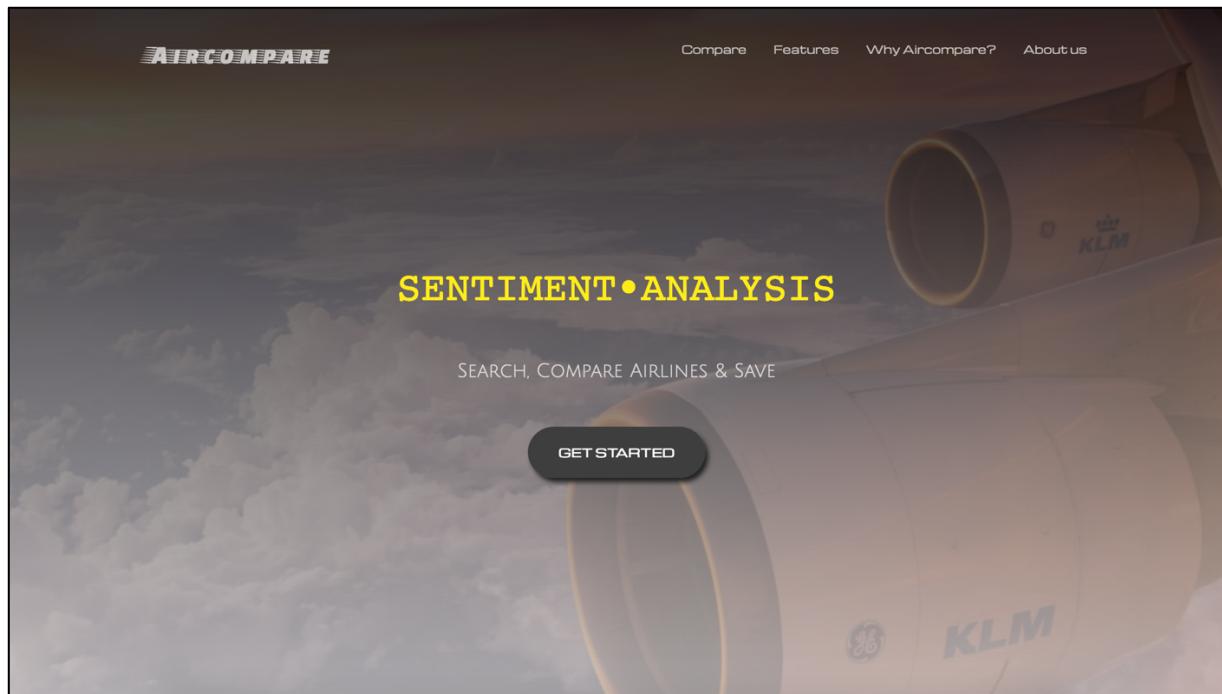


Figure 5.3-1 Page d'accueil Aircompare

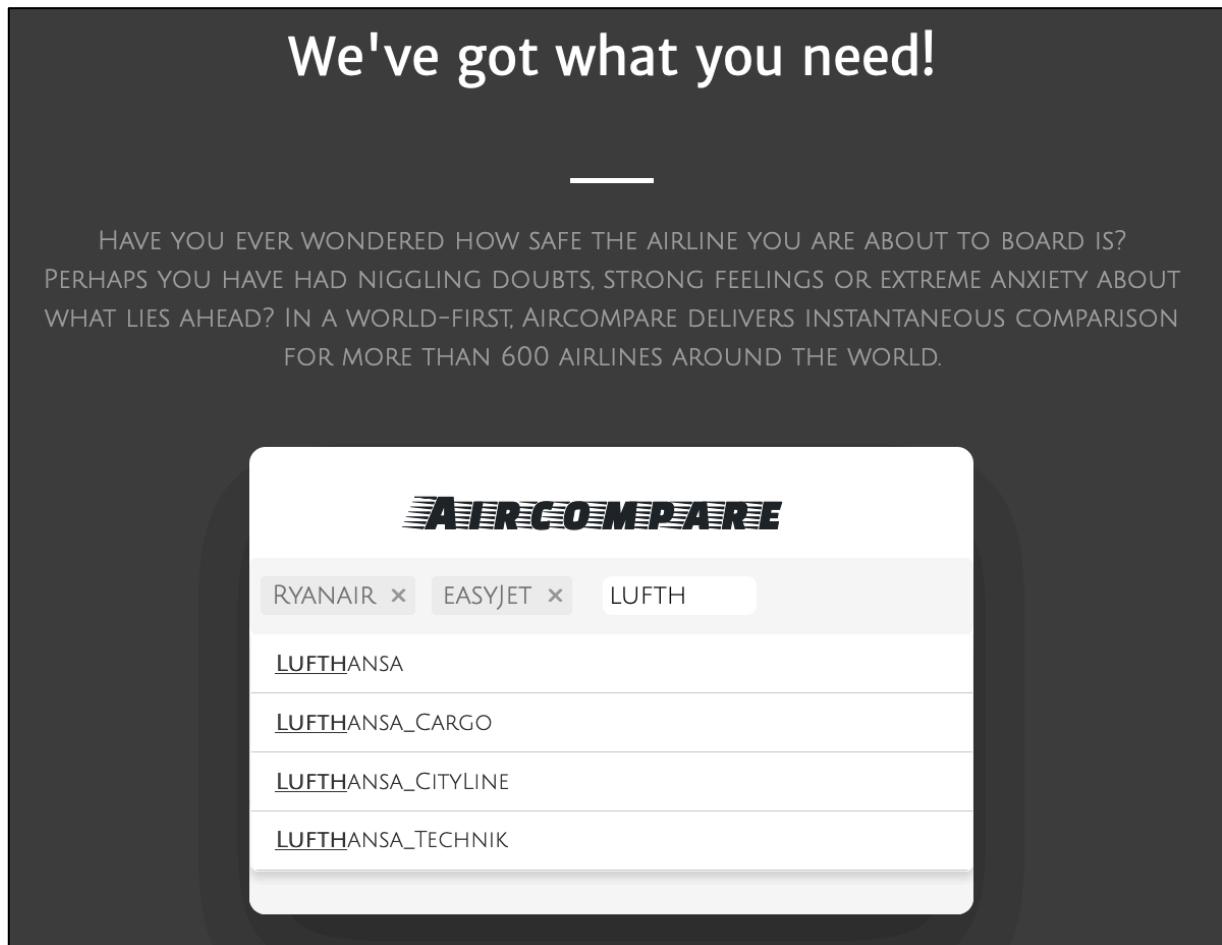


Figure 5.3-2 Formulaire de recherche en temps réel

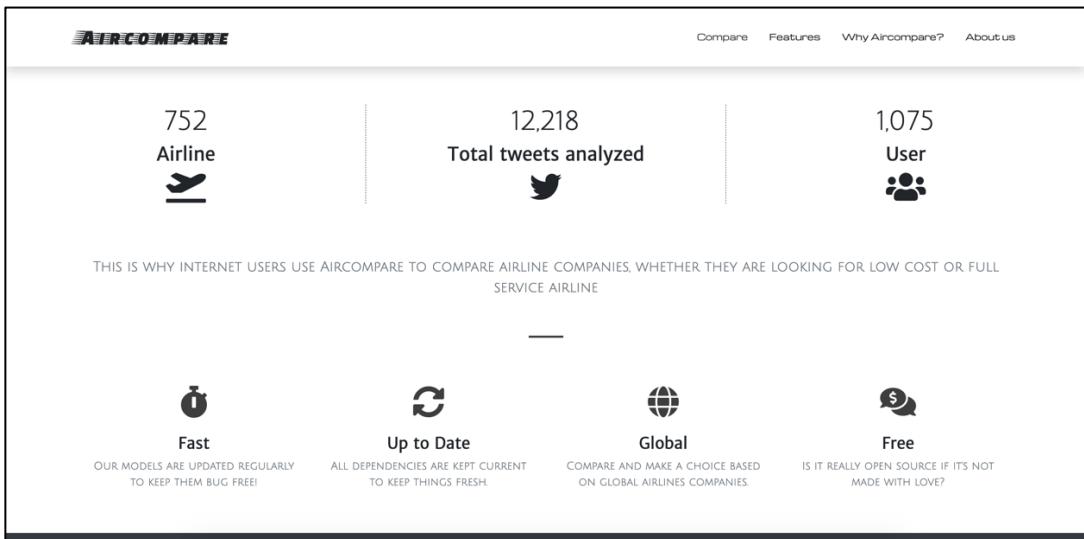


Figure 5.3-3 Page features

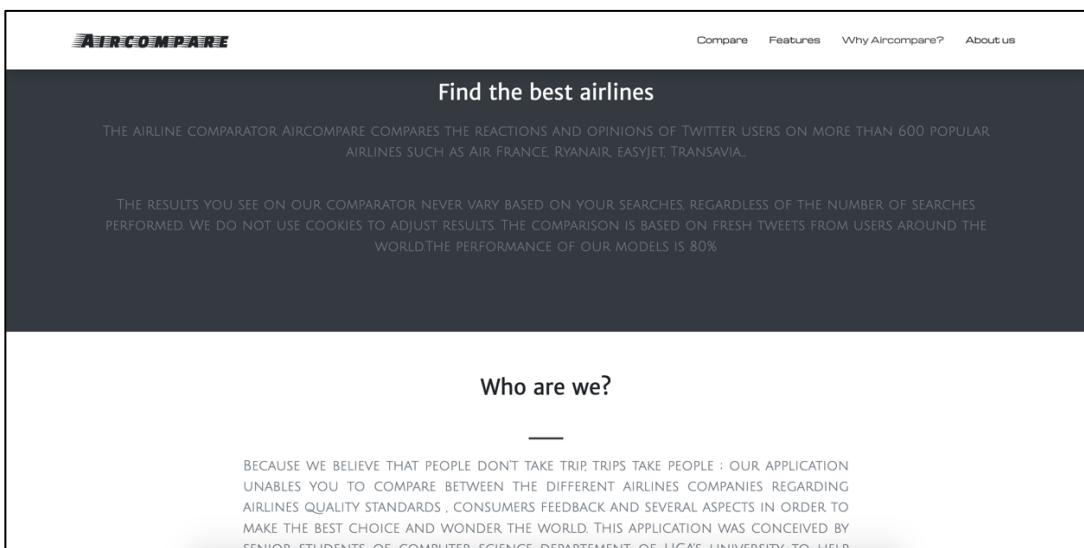


Figure 5.3-4 Renseignement sur Aircompare

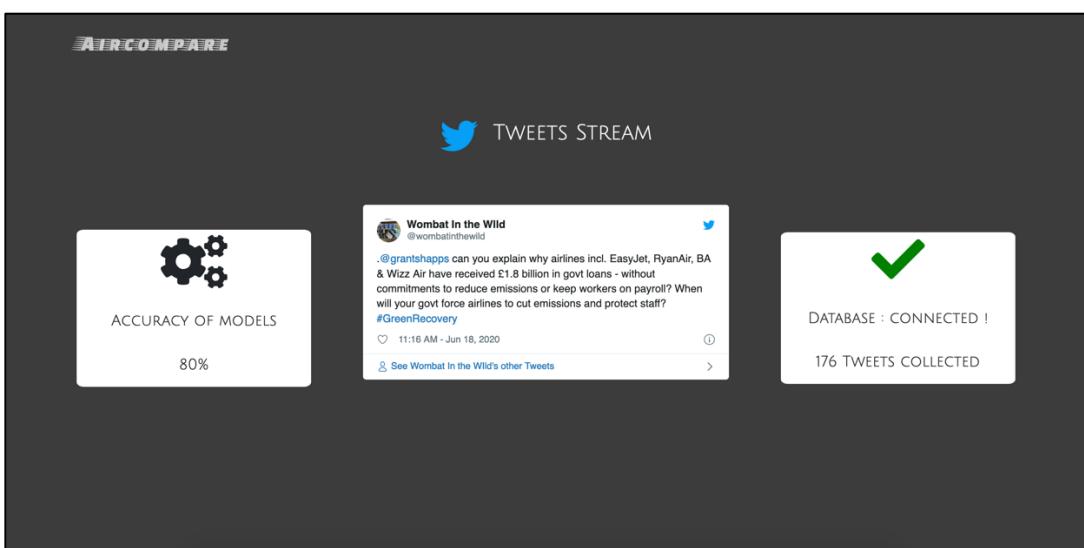


Figure 5.3-5 Récapitulatif de la recherche

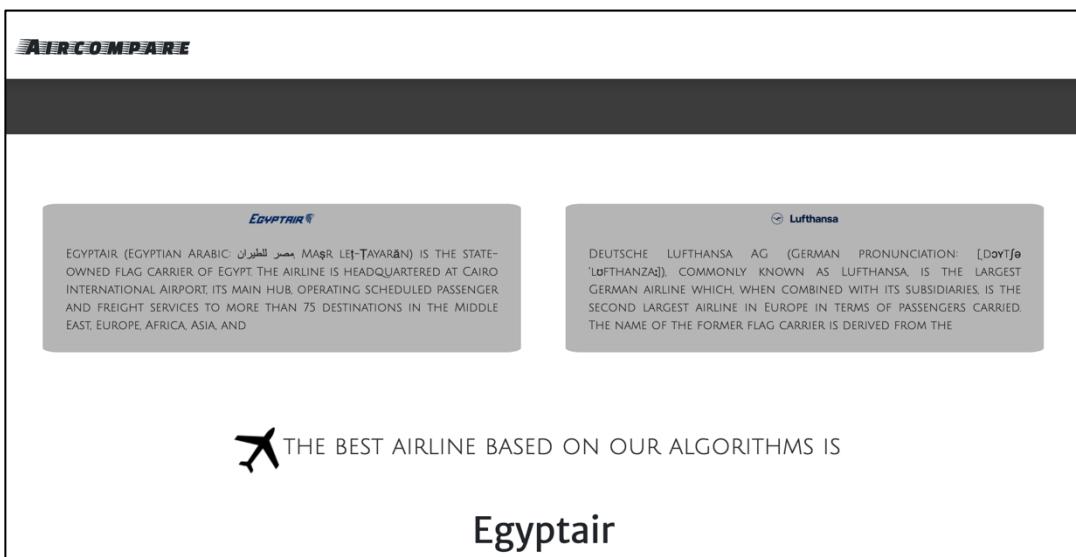


Figure 5.3-6 Sélection de la meilleure compagnie aérienne

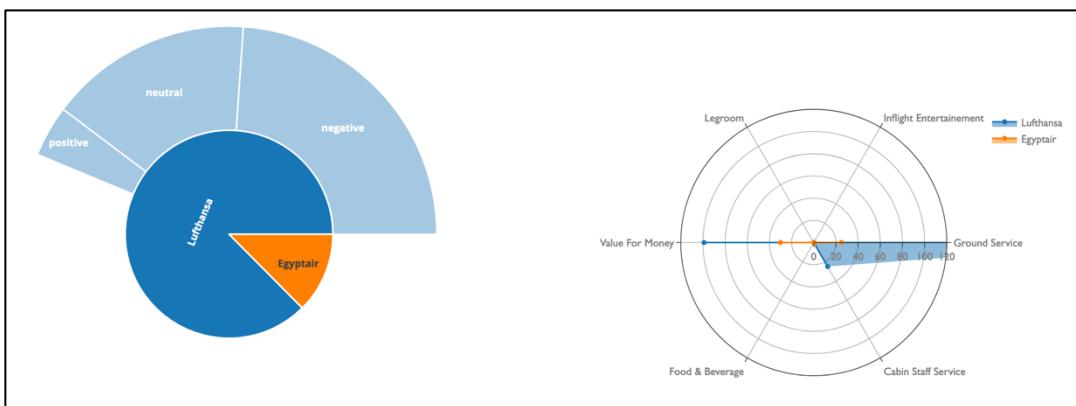


Figure 5.3-7 Visualisation du sentiment et des raisons de la négativité

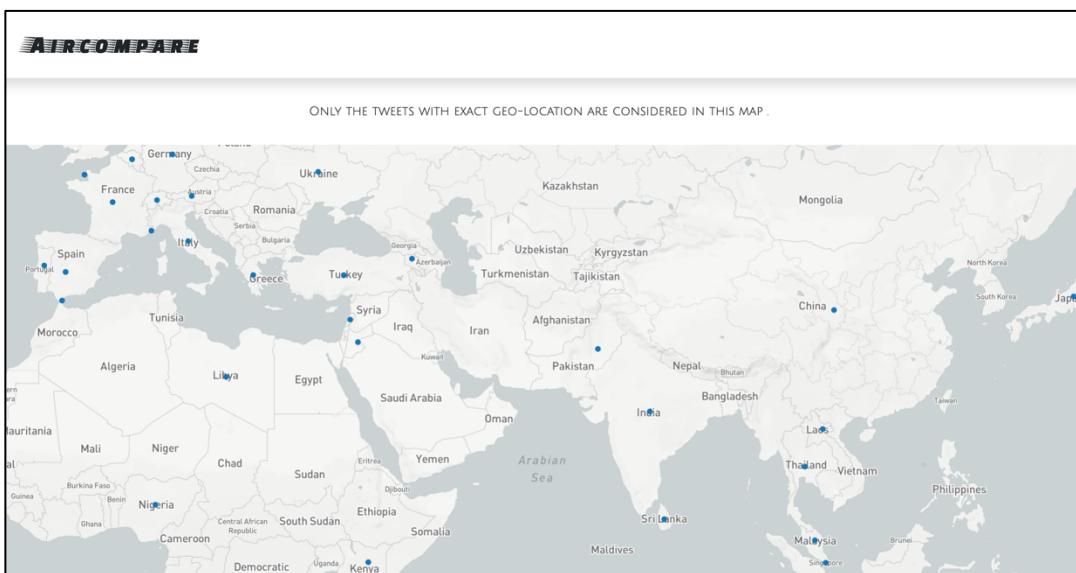


Figure 5.3-8 Localisation des tweets