1. Conclusion tirées

Le projet a permis de confirmer qu'il est possible de prédire de manière fiable le ressenti d'un client à partir du texte de son avis grâce à des techniques classiques de Machine Learning appliquées au traitement du langage naturel. Après avoir collecté et nettoyé plus de 5 600 avis issus de Trustpilot, nous avons construit progressivement un modèle capable de distinguer un avis positif d'un avis négatif. La transformation de la tâche de classification de multiclass (notes de 1 à 5) en classification binaire s'est révélée pertinente, car elle correspond davantage à un besoin métier (savoir si un client est satisfait ou non). Le modèle final, basé sur TF-IDF et une régression logistique, atteint un F1-score de 0,96 et une AUC de 0,98, ce qui atteste d'une excellente capacité de généralisation. Au-delà de la performance scientifique, ce projet démontre qu'un pipeline simple, reproductible et interprétable peut avoir un véritable intérêt opérationnel.

2. Difficultés rencontrées lors du projet

- Quel a été le principal verrou scientifique rencontré lors de ce projet ?
- Pour chacun des points suivants, si vous avez rencontré des difficultés, détaillez en quoi elles vous ont ralenti dans la mise en place de votre projet.
- Prévisionnel : tâches qui ont pris plus de temps que prévu, etc.
- Jeux de données : acquisition, volumétrie, traitement, agrégation, etc.
- Compétences techniques / théoriques : timing d'acquisition des compétences, compétence non proposée en formation, etc..
- Pertinence : de l'approche, du modèle, des données, etc..
- IT : puissance de stockage, puissance computationnelle, etc.
- Autres

Quel a été le principal verrou scientifique rencontré ?

Le principal verrou scientifique rencontré réside dans la subjectivité du langage humain. Certains avis étaient ambigus, ironiques ou neutres, ce qui rendait difficile leur classification automatique. Par exemple, des avis notés 3 étoiles contenaient des formulations ni clairement positives ni négatives, ou exprimaient un ressenti mitigé (« produit correct mais service lent »). Le modèle pouvait alors hésiter entre les deux classes. Ce phénomène, inhérent à toutes les tâches d'analyse de sentiments, a limité la précision maximale du modèle.

Prévisionnel

Certaines tâches ont pris plus de temps que prévu, notamment la collecte des données et le nettoyage du texte. L'extraction via web scraping a nécessité plusieurs tests avant d'obtenir un fichier exploitable. De plus, le preprocessing (suppression du HTML, normalisation, lemmatisation…) a été plus long que prévu, car il a fallu tester plusieurs approches pour obtenir un texte homogène. Cela a retardé le début de la phase de modélisation.

Jeux de données

Le premier problème rencontré à été le webscraping pour obtenir un jeu de données brut extraites de trust pilot, effectivement le choix des bibliothèques fut assez complexe car nous n'avions pas encore vu en formation la façon de webscraper et il a été très compliqué car la bibliothèque **BeautifulSoup** ne permettait pas de scraper Trustpilot car ce site charge la majorité de ses contenus (avis, notes, auteurs…) **dynamiquement via JavaScript**. Lorsque l'on récupère la page avec une simple requête HTTP, le code HTML obtenu est incomplet et ne contient pas les avis. De plus, Trustpilot utilise des **protections anti-scraping**(Cloudflare, cookies, vérifications JavaScript) qui empêchent l'accès direct aux données.

C'est pour cette raison que nous avons essayé d'utiliser **Selenium**, qui simule un véritable navigateur et permet de charger le contenu après exécution des scripts JavaScript.

La volumétrie n'était pas un problème majeur (≈ 5 600 avis), mais plusieurs limites ont été rencontrées : présence de doublons (près de 900), déséquilibre entre les classes (plus de 50 % d'avis à 5 étoiles), et 17 % de valeurs manquantes dans certaines colonnes (Title, Country…). Il a fallu nettoyer les doublons, éliminer les variables inutiles au modèle (Country, Author) pour éviter les biais, et gérer l'équilibre des classes à l'aide d'un RandomOverSampler.

Compétences techniques et théoriques

Nous débutions dans le domaine du NLP, ce qui a nécessité un temps d'apprentissage important pour comprendre TF-IDF, la lemmatisation, les métriques comme le F1-score, ou encore la validation croisée. Certaines compétences, comme l'interprétation des coefficients d'un modèle logistique ou l'utilisation d'outils comme SHAP, n'avaient pas été abordées en cours et ont dû être apprises en autonomie.

Pertinence de l'approche

Le modèle choisi (TF-IDF + Logistic Regression) s'est révélé performant mais pose la question de la pertinence à long terme face à des modèles plus modernes (BERT, embeddings contextuels). Toutefois, l'approche retenue était justifiée dans un cadre pédagogique : simple, explicable et efficace. Le choix de transformer le problème en binaire s'est également imposé pour respecter la logique métier.

IT / Ressources techniques

Les ressources informatiques nécessaires sont restées modestes : un ordinateur classique suffisait pour entraîner les modèles TF-IDF + LogReg ou TF-IDF + XGBoost. Aucune contrainte forte de mémoire ou GPU n'a été rencontrée. En revanche, la gestion des librairies et des versions (scikit-learn, nltk, xgboost) a parfois nécessité des ajustements.

Autres difficultés

La coordination à deux a demandé une organisation rigoureuse : répartition des tâches, partage du code, suivi des versions et homogénéisation des rapports écrits. Il a parfois été difficile d'avancer au même rythme en raison de disponibilités et des niveaux différentes.

3. Bilan

- Détaillez quelle a été votre contribution principale dans l'atteinte des objectifs du projet.
- Avez-vous modifié le modèle depuis la dernière itération ? Si oui, détaillez.
- Présentez les résultats obtenus et comparez-les au benchmark
- Pour chacun des objectifs du projet, détaillez en quoi ils ont été atteints ou non.
- S'ils ont été atteints, dans quel(s) process(es) métier(s) votre modèle peut-il s'inscrire ? Détaillez.

Globalement nous avons travaillé ensemble, chacun proposait son code et son travail puis nous nous concertions et ensuite nous débâtions et départagions quelle approche aborder.

o Nous avons initialement abordé le problème comme une classification multiclass, en conservant les notes de 1 à 5. L'analyse de la matrice de confusion a toutefois mis en évidence des confusions fréquentes entre les classes voisines, notamment entre les

notes 2 et 3 ainsi qu'entre 3 et 4. Ce constat nous a conduits à reformuler le problème en classification binaire, une approche plus adaptée à notre problématique métier, qui vise avant tout à distinguer les avis satisfaisants des avis insatisfaisants.

Le modèle final de classification binaire (avis "positif" vs "négatif"), construit à partir de la combinaison TF-IDF + Régression Logistique, a atteint un **F1-score de 0,96**, une **AUC-ROC de 0,98** et une **accuracy de 0,95** sur le jeu de test. Ces résultats indiquent un excellent compromis entre précision et rappel, et une très bonne capacité du modèle à reconnaître aussi bien les avis positifs que négatifs.

À titre de comparaison, les benchmarks académiques disponibles sur l'analyse de sentiments en langue anglaise (comme IMDB ou Amazon Reviews) montrent généralement des performances entre **0,90 et 0,95** avec TF-IDF + LogReg. Nos résultats se situent donc **au-dessus de la moyenne**, ce qui peut s'expliquer par le fait que les avis Trustpilot sont souvent très expressifs et moins ambigus que des textes neutres.

Bien que les performances obtenues par notre modèle soient élevées (F1-score ≈ 0,96), il est important de nuancer la comparaison avec les benchmarks issus de bases de données plus volumineuses comme Amazon Reviews, Yelp ou IMDB. En effet, notre dataset initial, constitué d'environ 5 600 avis Trustpilot, est plus restreint, moins hétérogène et centré sur un secteur spécifique (bagues connectées). Cette taille réduite, combinée à une structure d'avis relativement homogène, peut faciliter la tâche de classification pour le modèle et expliquer en partie les excellents résultats obtenus.Ainsi, plutôt que d'affirmer que notre modèle surpasse les benchmarks existants, il est plus juste de dire qu'il atteint un niveau de performance très satisfaisant **dans son propre contexte d'application**, mais que la généralisation à des données plus diverses ou à grande échelle nécessiterait de nouveaux tests et ajustements.

L'ensemble des objectifs fixés au début du projet a été atteint de manière progressive et conforme à la méthodologie proposée. Nous avons d'abord rempli les attentes du premier rendu en collectant, nettoyant et analysant les avis issus de Trustpilot, ce qui nous a permis de mieux comprendre la structure des données et de préparer un jeu exploitable. La deuxième phase, consacrée à la modélisation, a également été menée à bien : plusieurs modèles ont été testés, comparés, puis optimisés, aboutissant à une solution performante et interprétable (TF-IDF + Régression Logistique) affichant un F1-score d'environ 0,96. Ainsi, les objectifs techniques, méthodologiques et analytiques ont été atteints et les résultats obtenus sont pleinement satisfaisants.

Sur le plan métier, ce modèle peut être intégré à plusieurs processus concrets. Il pourrait notamment servir à automatiser la **veille de satisfaction client**, en classant

automatiquement les nouveaux avis en positifs ou négatifs. Il pourrait aussi s'intégrer dans un **tableau de bord marketing** afin d'identifier en temps réel les tendances et signaux faibles liés aux retours clients. Enfin, il pourrait être utilisé par un **service client**, pour déclencher des alertes lorsqu'un avis négatif est publié, ou pour prioriser les réponses aux clients insatisfaits. Cela montre que le projet n'a pas seulement atteint ses objectifs académiques, mais qu'il présente également un véritable potentiel d'application opérationnelle.

L'objectif principal du projet — **prédire automatiquement le ressenti d'un client à partir de son avis** — a été atteint avec succès. Nous avons collecté et nettoyé les données, mis en place une modélisation robuste, analysé les résultats et proposé des pistes d'amélioration.

Certains objectifs secondaires ont également été atteints :

- Compréhension du pipeline complet de traitement de données textuelles.
- Création d'un modèle stable, explicable et reproductible.
- Capacité à identifier les mots les plus discriminants (interprétabilité).

Seuls certains points sont restés limités :

- Pas d'industrialisation du modèle (API ou interface utilisateur).
- Pas de prise en compte du multilingue ni de la sémantique profonde (transformers).

Intégration dans un processus métier :

Dans une entreprise, un tel modèle pourrait être intégré dans :

- Un **outil de veille de satisfaction** permettant de classer automatiquement les nouveaux avis.
- Un **dashboard marketing** qui remonte quotidiennement le nombre d'avis négatifs.
- Un système interne qui **déclenche une alerte au service client** lorsqu'un avis critique est publié.
- Une **analyse concurrentielle** pour comparer différentes marques sur une période donnée.

4. Suite du projet – améliorations possibles

Plusieurs pistes peuvent être envisagées pour améliorer la performance et la valeur ajoutée du modèle :

- **Approches plus avancées du langage naturel** : utiliser des embeddings contextuels (BERT, CamemBERT, DistilBERT…), qui capturent mieux l'ironie, la négation et le contexte.
- **Prise en compte du multilingue** : certains avis Trustpilot peuvent être en anglais, mais aussi en d'autres langues ; un modèle multilingue serait plus robuste.
- **Extraction d'émotions fines** : au lieu d'un simple positif/négatif, détecter des sentiments comme frustration, surprise, mécontentement logistique, etc.
- **Industrialisation** : création d'une API REST, intégration dans un tableau de bord PowerBI ou Streamlit, automatisation de la collecte des avis.
- **Modèle auto-apprenant** : permettre au modèle de se réentraîner régulièrement avec de nouveaux avis.

5. Contribution à la connaissance scientifique

Même si le projet reste pédagogique, il contribue à la connaissance scientifique dans le sens où :

- Il prouve qu'**une approche simple (TF-IDF + LogReg)** peut rivaliser avec des modèles plus lourds lorsqu'elle est bien optimisée.
- Il montre l'intérêt d'une **transformation de tâche (multiclass → binaire)** pour coller au besoin métier.
- Il met en évidence certaines **caractéristiques linguistiques des avis négatifs** (textes plus longs, lexique émotionnel plus fort).
- Il ouvre la voie à une **industrialisation légère** applicable en entreprise sans infrastructure complexe.

6. Annexes

Bibliographie :

Pour mener à bien ce projet, nous nous sommes appuyés sur plusieurs types de ressources :

- **Documentation technique :**
    - Documentation officielle de Scikit-learn (TF-IDF, LogisticRegression, train_test_split, classification_report)
    - Documentation NLTK / SpaCy pour le preprocessing (stopwords, lemmatisation)
    - GitHub, Medium, Towards Data Science pour des exemples de pipelines NLP.
- **Articles et blogs (Machine Learning & NLP) :**
    - "Sentiment Analysis with TF-IDF and Logistic Regression" – Towards Data Science
    - "Why Logistic Regression works so well on text" – Medium
    - Papers sur TF-IDF : Salton & Buckley (1988)
- **Outils utilisés :**
    - Trustpilot.com pour les données
    - Bibliothèques Python : pandas, scikit-learn, nltk, xgboost, imblearn
    - TextBlob pour la polarité et subjectivité.
    - Plateforme de cours datascientest

Diagramme de Gantt :

Rendu 1 – Exploration & Data Viz (avril – juillet) : léger retard/rendu en septembre

- Choix de la problématique
- Recherche / collecte des données
- Compréhension des variables
- Nettoyage & pré-traitement des données
- Visualisations / statistiques descriptives
- Rédaction du rapport 1

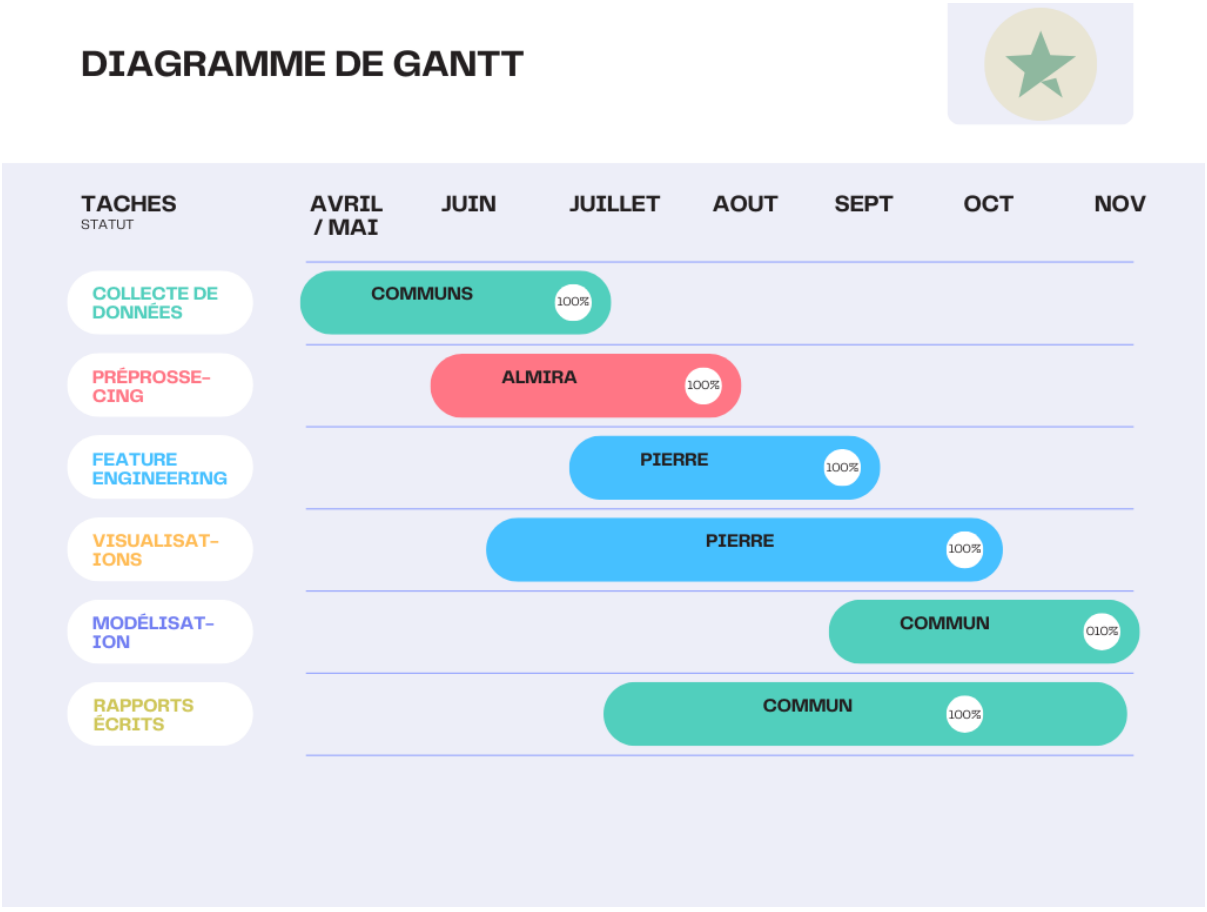
Rendu 2 – Modélisation (septembre – novembre)

- Choix des modèles de ML
- Découpage du dataset / split train-test
- Modèles testés (baseline → avancés)
- Optimisation (grid search, cross-validation)
- Interprétation des résultats
- Rédaction rapport 2

Rapport final (novembre - décembre)

- Intégration des 2 rapports
- Conclusion + limites + améliorations
- Mise en page + bibliographie + annexes
- Préparation soutenance orale

Tâches	Almira	Pierre
Collecte de données	☑	☑
Préprossecing	☑	
Feature engineering		☑
Visualisations		☑
Modélisation	☑	☑
Rapports écrits	☑	☑



Code : https://github.com/DataScientest-Studio/sept24_alt_truspilot_2