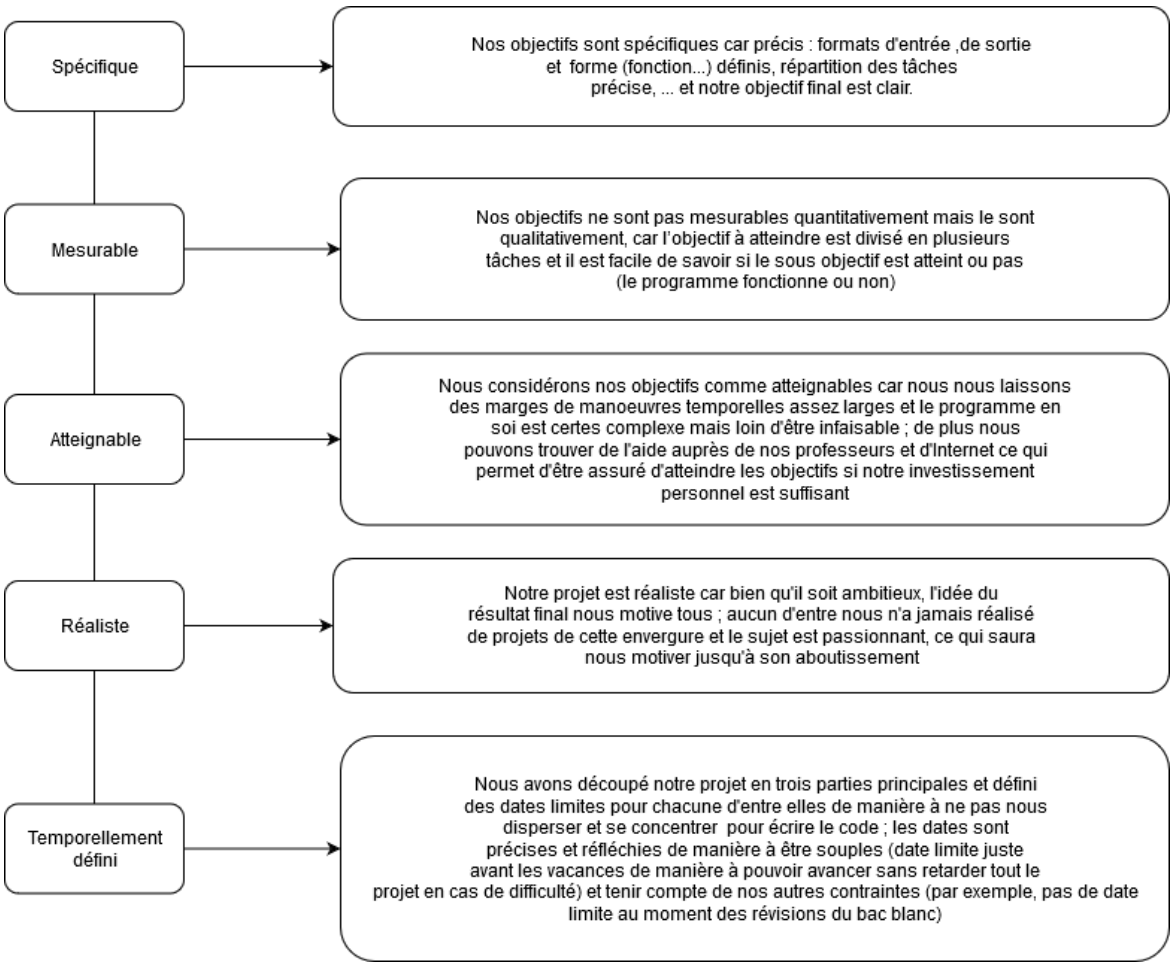
Machine learning : a sentiment analysis project (Sprint 1)

Conception

Résultats de notre brainstorming : nos objectifs

Notre objectif est de déterminer si un commentaire, laissé sur un site de commande en ligne (dans notre cas Amazon), est positif (5 étoiles) ou négatif (1 étoile), en comparant à une base de données de commentaires associés à la note mise par l'utilisateur, par une méthode d'apprentissage. Pour cela nous allons calculer la probabilité qu'il soit positif uniquement. En effet, puisque nous ne gardons que des commentaires de 1 ou 5 étoiles, nous pouvons établir que si la probabilité que le commentaire soit positif vaut P, alors la probabilité que le commentaire soit négatif vaut 1-P. Considérer les commentaires comme soit strictement négatifs soit strictement positifs est une approximation mais tenant compte de l'objectif (déterminer si le commentaire est positif), nous devrions atteindre malgré cela des résultats satisfaisants.

Présentation SMART de nos objectifs



Organisation

Pour arriver à nos objectifs, nous allons utiliser tout au long de notre projet le site GitLab, pour nous

permettre de modifier le code depuis chez nous, qu'il soit accessible pour tous et pour pouvoir profiter d'un système de contrôle de versions, ainsi que HackMD (ce document est d'ailleurs consultable en ligne ici), pour rédiger nos notes.

Nous avons réparti le projet en 3 étapes principales :

- un ensemble de petites tâches de début de projet (tri, pre-processing du texte, tri de nouveau)
- création d'un algorithme associant à chaque mot un score de positivité (par du machine learning)
- application du théorême de Bayes pour associer à chaque commentaire la probabilité qu'il soit positif, selon les mots qu'il contient

Planning

1ère partie : tri divers *(31 janvier)*

Cette partie est en réalité divisée en trois sous-parties que nous avons pu répartir entre nous :

Louise :

- Télécharger et ouvrir un fichier json contenant un dataset de commentaires Amazon en anglais (commentaires et avis associés à une note allant de 1 à 5 étoiles) que l'on peut trouver sur internet et le convertir en un fichier csv.
- Supprimer les données inutiles, c'est-à-dire la date, le nom de la personne, le titre, mais également les commentaires ayant 2, 3 ou 4 étoiles. Ainsi, on ne garde que le commentaire et la note correspondante, qui est de 1 ou de 5 pour simplifier l'analyse (le commentaire est donc soit positif, soit négatif). Commencer avec 1000 lignes (500 commentaires positifs et 500 commentaires négatifs), extraire les données et les transformer en liste.

Mathis :

- Traitement des données pour nettoyer le texte (data pre-processing) : mettre tout en minuscule (lowercasing), supprimer les mots vides, c'est-à-dire les mots de liaisons et les mots très courants qui n'ont pas d'importance dans l'analyse du texte, notamment and, the, a, it, they, this etc… (stop-words removal) et supprimer la ponctuation (simple anomalies noise removal). Cela nous permettra d'avoir un texte plus simple à analyser.

Jeanne :

- Rechercher les mots les plus fréquents dans les commentaires (après le traitement du texte) pour ne garder que ceux qui sont présents dans plus de 30% des commentaires (valeur du pourcentage à ajuster), et calculer leur fréquence d'apparition dans la totalité des commentaires.

Format des entrées/sorties :

Louise :

- entrée : fichier CSV pré-traité à la main
- sortie : liste de type `data_original = [("comm1", True), ("comm2", False), ("comm3", False)]` où True signifie un commentaire positif (5 étoiles) et False un commentaire négatif (1 étoile)

Mathis :

- entrée : liste du même type que `data_original` (voire plus haut)
- sortie identique

- format : fonction

Jeanne :

- entrée : liste du même type que `data_original`
- sortie : une autre liste de type : `freq_words = [("word1", freq_word1), ("word2", freq_word2)]` où `freq_word` est un float compris entre 0 et 1
- format : fonction

2ème partie : score de positivité *(22 février)*

Cette partie est elle aussi divisée en 3 sous-parties. Elle consiste à associer à chaque mot son *score de positivité*, c'est-à-dire la probabilité que le commentaire soit positif lorsqu'il contient ce mot. Nous allons pour cela procéder en plusieurs étapes.

Fréquence d'apparition du mot dans les commentaires positifs (Mathis)

Tout d'abord, après avoir calculé dans la première partie la fréquence d'apparition des mots les plus fréquents dans la totalité des commentaires, nous allons, de la même manière, calculer la fréquence d'apparition de ces mêmes mots dans les commentaires positifs uniquement. Cela nous permettra d'associer à chaque mot sa fréquence d'apparition dans les commentaires positifs, ou `freq_word_in_pos`, ce qui correspond à la probabilité que le mot soit présent dans un commentaire positif. Si cette probabilité est comprise entre 0,45 et 0,55 (valeurs à ajuster), on ne conserve pas ce mot dans la liste car cela signifie qu'il est autant présent dans les commentaires positifs que négatifs, et donc qu'il ne nous permet pas de les différencier.

Format :

- entrée : liste des mots selectionnés lors de la partie 1 et qui sont dans la liste `freq_words`
- sortie : liste de type : `freq_words_in_pos = [("word1", freq_word1_in_pos), ("word2", freq_word2_in_pos)]` où `freq_word_in_pos` est un float compris entre 0 et 1

Probabilité que le commentaire soit positif et qu'il contienne le mot (Louise)

Cela nous permettra de calculer, dans un deuxième temps, la probabilité que le commentaire soit positif (`ComPos`) et que ce soit un commentaire qui contienne le mot (`word1`) :

$$P(ComPos \cap word1) = freq\_word1 + prob\_pos - freq\_word1\_in\_pos$$

Sachant que `freq_word` désigne la fréquence d'apparition du mot dans la totalité des commentaires, donc la probabilité qu'un commentaire contienne ce mot.

Et `prob_pos` désigne la probabilité que le commentaire soit positif, donc dans notre cas, si on conserve une base de données de 1000 commentaires dont 500 positifs, `prob_pos=1/2` .

Probabilité que le commentaire soit positif sachant qu'il contient le mot (Jeanne)

Ensuite, nous allons calculer la probabilité que le mot soit présent dans le commentaire (`word1`) sachant que le commentaire est positif (`ComPos`), afin d'obtenir le *score de positivité* de chaque mot:

$$P(word1|ComPos) = \frac{P(ComPos \cap word1)}{prob\_pos} = pos\_score1$$

Format :

- sortie : dictionnaire de type `pos_score = {"word1":pos_score1, "word2":pos_score2}` où `pos_score1` désigne le score de positivité de `word1` (float compris entre 0 et 1)

3ème partie : Bayes *(3 mai)*

Louise, Mathis et Jeanne :

Cette 3ème partie a pour but d'analyser un nouveau commentaire (toujours de 1 ou 5 étoiles) pour savoir s'il est positif ou non.

- Pour un commentaire donné, faire une liste avec uniquement les mots qui se trouvent dans ce commentaire et qui ont été précédement analysés (Louise ou commun)
- Récupérer dans la liste `pos_score` le score de positivité de ces mots
- Utiliser la classification naïve Bayésienne (issue du théorême de Bayes), pour nous permettre de connaître la probabilité que ce commentaire soit positif, selon le score de positivité des différents mots utilisés. Cela nous donne:

$$P(ComPos|word1, word2, word3) = \frac{pos\_score1 * pos\_score2 * pos\_score3}{freq\_word1 * freq\_word2 * freq\_word3}$$

- Afficher le résultat de l'analyse : si la probabilité que le commentaire soit positif est supérieure à 0,6 alors le commentaire a une note de 5/5, si elle est inférieure à 0,4 alors le commentaire a une note de 1/5 et si elle est comprise entre 0,4 et 0,6 alors on considère que le message est neutre donc qu'on ne peut pas deviner la note (valeurs à ajuster). On affiche ensuite la note réelle qui a été mise par l'utilisateur afin de comparer.

Améliorations possibles :

Si on termine avant la date limite que l'on s'est imposé (le 3 mai), plusieurs améliorations sont possibles et faciles à répartir entre nous :

- Calculer la fréquence d'erreurs (par Jeanne)
- Créer une interface graphique qui afficherait le commentaire, la note devinée par l'algorithme, puis la note réelle, ou éventuellement qui permettrait de rentrer notre propre commentaire (en anglais) et de voir s'il estime que c'est un commentaire positif ou négatif (par Jeanne/Louise)
- Faire un jeu entre l'utilisateur et l'algorithme pour déterminer la note : le commentaire s'affiche et il faut deviner s'il est positif (5 étoiles) ou négatif (1 étoile) (par Louise/Jeanne)
- Améliorer le pre-processing du texte, pour par exemple reconnaître les abréviations qui sont fréquentes et qui pour l'instant ne sont pas reconnues. Cela permettrait de rendre l'analyse du texte plus précise (normalization, voire text enrichment) (par Mathis)

Groupe 7 : DUPOUY Jeanne, FIRMINO Mathis, MERCIER Louise (TS2)
01/2020