Application de techniques de Data Mining en utilisant le logiciel RapidMiner

1 Classification de spam

Pour la création des fichiers CSV utilisé pour la génération de règles d'associations, un nettoyage des données a été fait.
Les lignes qui ont des données vides sont filtrés, comme par exemple le nom du produit. Quand la quantité(Quantity) est plus petite que zéro, les lignes sont aussi filtrées.

Un algorithme a aussi été fait pour détecter des noms de produit incohérent et pour filtrer les lignes. Voici la liste des noms incohérents trouvés : *wet pallet,sold in set ?,wet damaged,faulty,posy candy bag,michel oops,wet/mouldy, lost,dagamed,daisy notebook,fba,taig adjust,water damage,ribbons purse, ?display ?,found in w/hse,damaged stock, samples, mixed up, ???missing,breakages,chilli lights,dotcom set,re-adjustment,packing charge,amazon sales, ?, mix up with c,owl doorstop,stock check,mia,wrap folk art,carriage,damages, ?lost,john lewis,found box,20713,show samples,damages wax,website fixed,jumbo bag owls,wet boxes,frog candle,found,given away,mouldy, ???,broken, wrong code,damages ?,display,wet,bunny egg box,polkadot pen, ?missing,dotcom,amazon adjust,damaged,returned, toybox wrap,smashed, ?sold as sets ?,showroom,led tea lights,wet/rusty,missing ?,wrong barcode,lost in space, crushed ctn,popcorn holder,dotcom sales,lost ??,sold as 1,bingo set,thrown away.,crushed boxes,test,check,crushed, wet ?,mailout,amazon,rain poncho,adjust,ebay,adjustment,spotty bunting,wet rusty,cracked,sale error, ???lost, water damaged,missing,garage key fob,dotcomstock,dotcom adjust,sold as 22467, ??,dotcom postage,jumbo bag toys, shoe shine box, ?? missing,on cargo order,wrong code ?,check ?,label mix up,postage,wrap carousel,can't find, wrongly marked,retrospot lamp,thrown away,counted, ????missing,cordial jug*

De plus, chaque donnée de colonne est systématiquement trimée pour éviter tout problème et pour faciliter la suite du traitement.

Voici des données qui sont calculées lors du filtrage :
- Nb ligne dans le fichier : 541909
- Nb ligne après nettoyage : 397924
- Nb produit : 3639

Nous avons donc filtré 143985 lignes.

Un autre algorithme a été fait pour standardiser la description des produits en liant la description avec le numéro du produit et l'on garde la description qui est la plus utilisée.
Pour voir une partie de l'algorithme, il faut regarder la fonction Main.resolveProductDescription() dans le code Java.

Voici d'autres informations que nous calculons lors de la création des nouveaux fichiers CSV
Création du fichier par pays(permets de tester rapidement le code) :
- Nb ligne in new file(byCountry.csv) : 37
- Processing Time : 00 :01.625
- Total time for create the file : 00 :01.656

Création du fichier par client :
- Nb ligne in new file(byCustomer.csv) : 4339
- Processing Time : 01 :44.361
- Total time for create the file : 01 :44.674

Création du fichier par facture :
- Nb ligne in new file(byInvoice.csv) : 18536

— Processing Time : 07 :26.901
— Total time for create the file : 07 :27.901

Afin de pouvoir générer ces données nous avons utilisé le multi threading ce qui nous permet de diviser par deux le temps de création des fichiers.

Pour lire le fichier, nous avons dû en créer un nouveau et copiant et collant son contenu. Nous pensons qui doit avoir un problème d'encodage.

Pour tester le programme il suffit de lancer la commande mvn clean install, de modifier le fichier application.properties, pour définir ou se trouve le fichier de base, et où l'on veut créer les nouveaux fichiers.

La génération de ces fichiers n'est pas simple et nous a pris beaucoup de temps. Nous pourrions encore améliorer la génération, en tenant compte des produits retourner, par exemple. Il serait aussi bien d'avoir un meilleur descriptif de ces données.

1.1 Questions sur la classification

Dans le bloc "Process Documents from Data" nous n'avons pas mis d'etape de stemming. Est-ce que l'ajout de ce préprocessing a un impact sur les resultats obtenus ?

Dans l'exemple ci-dessus, nous avons utilise un classificateur bayésien, veuillez essayer d'autres familles de classificateurs, quel est l'impact sur le resultat obtenu ?

Finalement vous utiliserez la seconde source de donnees (emails.zip) sur laquelle vous appliquerez le meme process. Que constatez-vous en comparant les 2 résultats ?

2 Market basket analysis

2.1 Questions about association rules

Constatez-vous des différences dans les règles d'associations obtenues entre les 2 regroupements différents (par facture/par client) ? Veuillez commenter vos résultats.

Est-il possible de générer une/des autre/s colonne/s à partir des donnees initiales qui produisent des règles interessantes ?

3 Utilisation d'un WordNet sur des commentaires d'utilisateurs

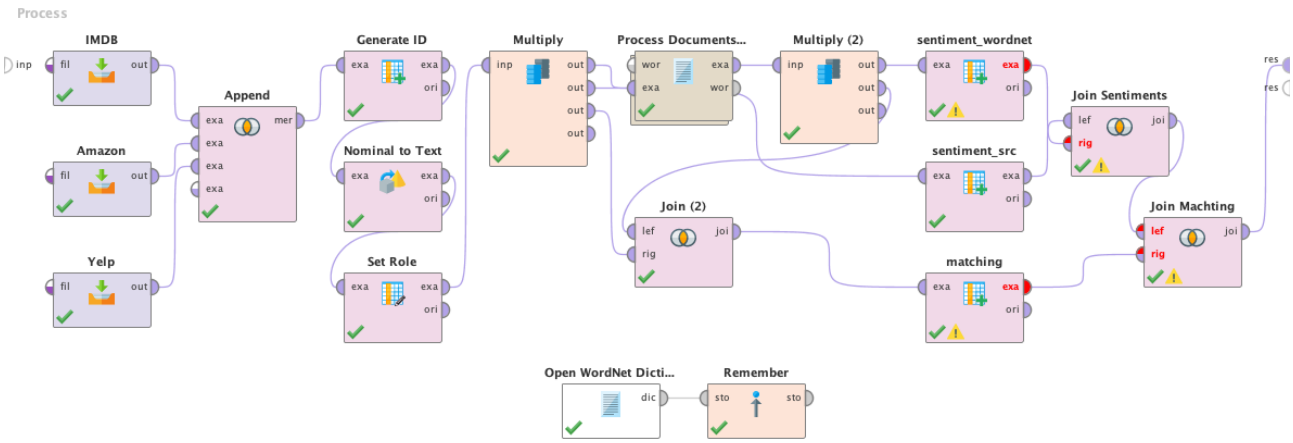

FIGURE 1 – Processus complet : WordNet sur 3000 commentaires

Dans le cadre de cette 3e partie, nous avons mis en place un processus capable de comparer des commentaires sous format texte fourni avec un étiquetage du sentiment (Positif, Négatif, et Neutre), avec un algorithme de prédiction capable d'extraire le sentiment d'un texte basé sur le dictionnaire WordNet. Pour simplifier la comparaison, nous avons ajouté un attribue qui vérifie que le sentiment source est le même que le sentiment prédit (matching).

3.1 Questions sur les règles d'association

Commentez les resultats obtenus par rapport aux etiquettes existantes sur le dataset. Quelle est l'influence des différentes étapes de "text processing" sur le résultat que vous obtenez ?

3.1.1 Process Documents from Data sur touts les éléments communs

Dans le cas où on utilise tout les éléments courant pour prétraiter le text fourni à la fonction "Extract Sentiment" (Fig. 13), nous obtenons un résultat à 46% correct. C'est à dire :

- tokenisation
- transformation en caractères minuscules
- suppression des stop words
- filtre sur la longueur des tokens
- stemming (particularité nous utilisons le stemming basé sur le WordNet)

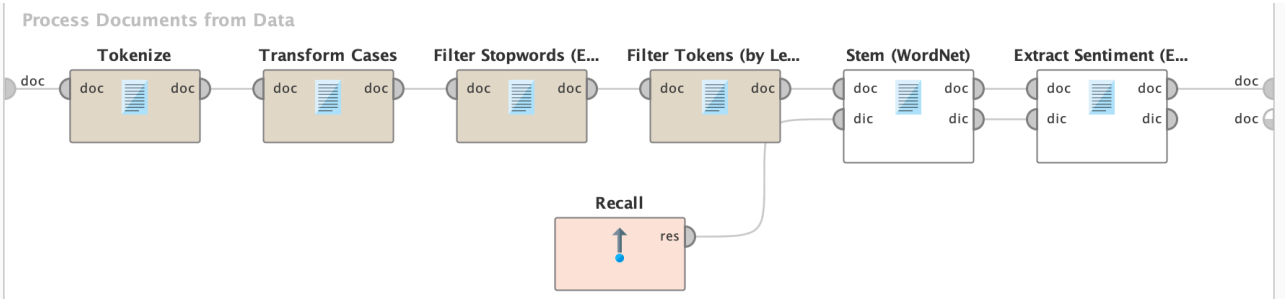

FIGURE 2 – Process Documents from Data with all elements

...	id	sentiment_src	sentiment_wordnet	matching
1	id_1	Neutral	Negative	false
2	id_2	Neutral	Negative	false
3	id_3	Neutral	Negative	false

matching
Nominal
Missing: 0
Least: true (1396)
Most: false (1604)

FIGURE 3 – Results of the Process Documents from Data with all elements

3.1.2 Process Documents from Data uniquement avec le module d'extraction du sentiment WordNet

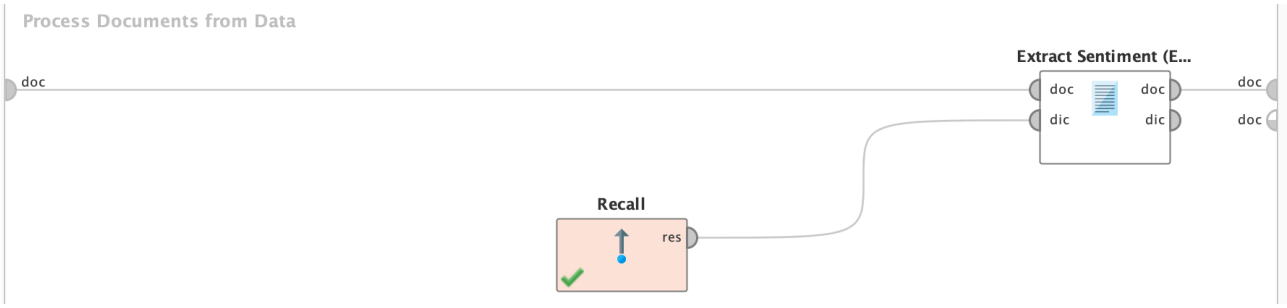

FIGURE 4 – Process Documents from Data with all elements

Row No.	id	sentiment_...	sentiment_...	matching
1	id_1	Neutral	Neutral	true
2	id_2	Neutral	Neutral	true
3	id_3	Neutral	Neutral	true

matching
Nominal
Missing: 0
Least: true (1500)
Most: false (1500)

FIGURE 5 – Process Documents from Data with all elements

3.1.3 Process Documents from Data composé de la tokenisation

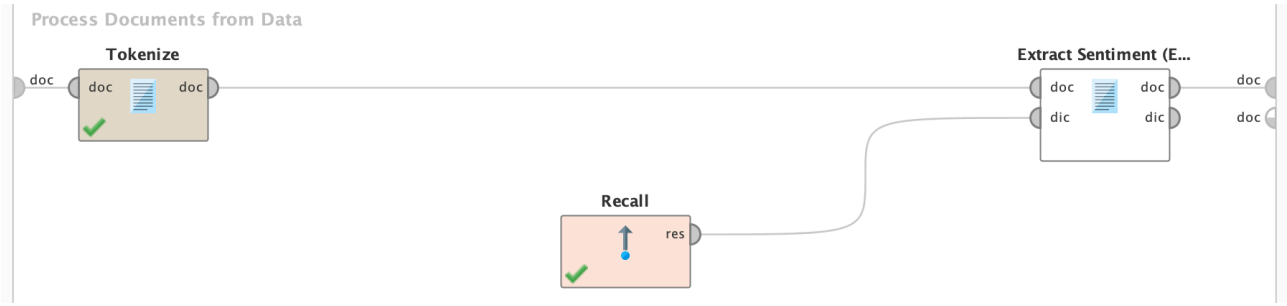

FIGURE 6 – Process Documents from Data with all elements

Row No.	id	sentiment_...	sentiment_...	matching
1	id_1	Neutral	Positive	false
2	id_2	Neutral	Negative	false
3	id_3	Neutral	Negative	false
4	id_4	Neutral	Negative	false

matching
Nominal
Missing: 0
Least: true (1282)
Most: false (1718)

FIGURE 7 – Process Documents from Data with all elements

3.1.4 Process Documents from Data composé de la tokenisation ainsi que de la transformation en caractères minuscules

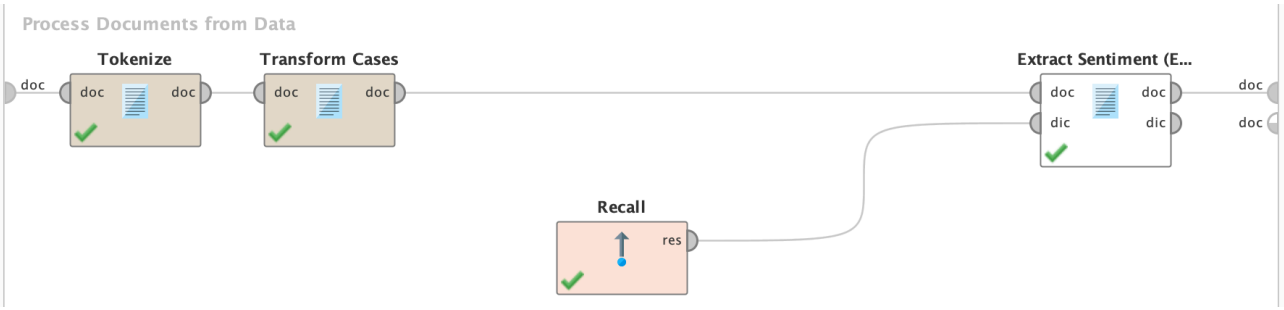

FIGURE 8 – Process Documents from Data with all elements

Row No.	id	sentiment_...	sentiment_...	matching
1	id_1	Neutral	Positive	false
2	id_2	Neutral	Negative	false
3	id_3	Neutral	Negative	false
4	id_4	Neutral	Negative	false

matching
Nominal
Missing: 0
Least: true (1283)
Most: false (1717)

FIGURE 9 – Process Documents from Data with all elements

3.1.5 Process Documents from Data composé de la tokenisation, la transformation en caractères minuscules, et la suppression des stop words

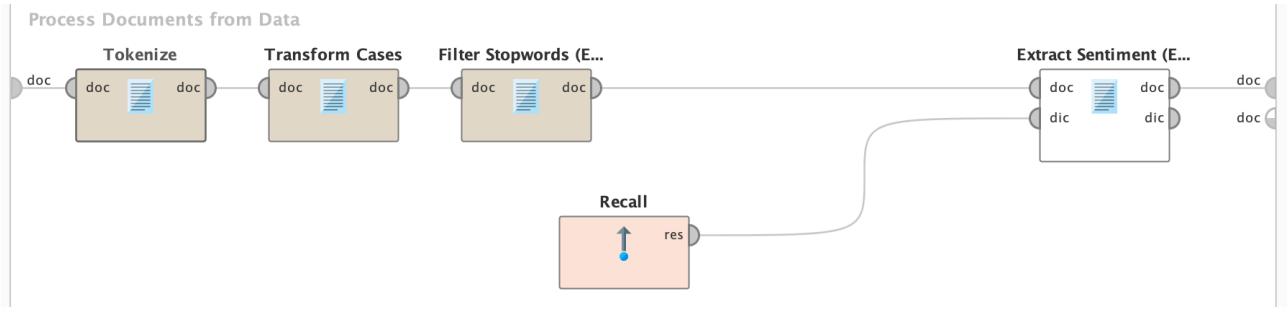

FIGURE 10 – Process Documents from Data with all elements

Row No.	id	sentiment_...	sentiment_...	matching
1	id_1	Neutral	Negative	false
2	id_2	Neutral	Negative	false
3	id_3	Neutral	Negative	false

matching
Nominal
Missing: 0
Least: true (1263)
Most: false (1737)

FIGURE 11 – Process Documents from Data with all elements

3.1.6 Process Documents from Data composé de la tokenisation, la transformation en characters minuscule, la suppression des stop words, et un filtre sur la longueur des tokens

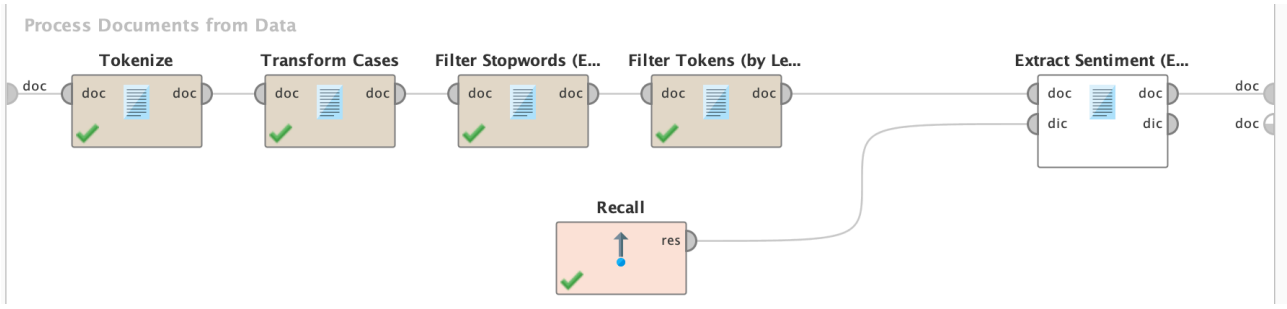

FIGURE 12 – Process Documents from Data with all elements

Row No.	id	sentiment_...	sentiment_...	matching
1	id_1	Neutral	Negative	false
2	id_2	Neutral	Negative	false
3	id_3	Neutral	Negative	false
4	id_4	Neutral	Positive	false

matching
Nominal
Missing: 0
Least: true (1398)
Most: false (1602)

FIGURE 13 – Process Documents from Data with all elements

3.1.7 Seuil par default du threshold du module d'extraction de sentiment de WordNet à 0.05

Les résultats sont basé sur la topologie complète (Fig.13)

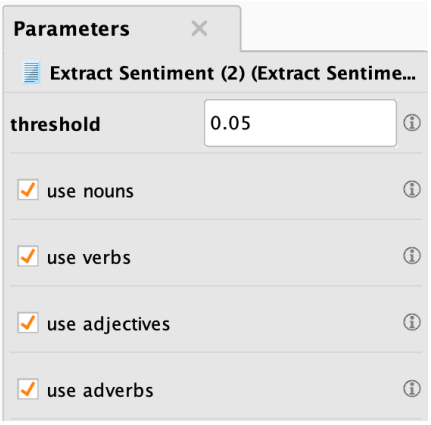

FIGURE 14 – Extract Sentiment Threshold at 0.05

Row No.	id	sentiment_...	sentiment_...	matching
1	id_1	Neutral	Neutral	true
2	id_2	Neutral	Neutral	true
3	id_3	Neutral	Neutral	true
4	id_4	Neutral	Neutral	true

matching
Nominal
Missing: 0
Least: false (1450)
Most: true (1550)

FIGURE 15 – Result of the extracted sentiment with the threshold at 0.05

3.1.8 Modification du threshold du module d'extraction de sentiment de WordNet à 0.5

FIGURE 16 – Extract Sentiment Threshold at 0.5

Row No.	id	sentiment_...	sentiment_...	matching
1	id_1	Neutral	Negative	false
2	id_2	Neutral	Negative	false
3	id_3	Neutral	Positive	false

matching
Nominal
Missing: 0
Least: false (1465)
Most: true (1535)

FIGURE 17 – Result of the extracted sentiment with the threshold at 0.5

3.1.9 Modification du threshold du module d'extraction de sentiment de WordNet à 0.01

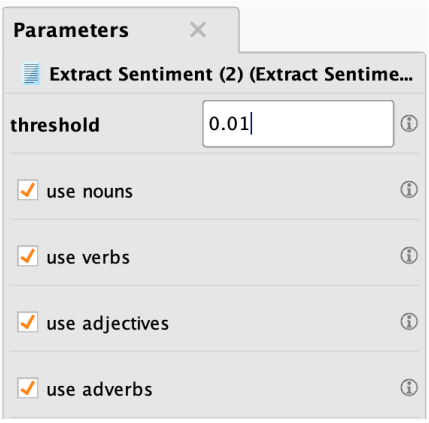

FIGURE 18 – Extract Sentiment Threshold at 0.05

Row No.	id	sentiment_...	sentiment_...	matching
1	id_1	Neutral	Negative	false
2	id_2	Neutral	Negative	false
3	id_3	Neutral	Negative	false
4	id_4	Neutral	Positive	false

matching
Nominal
Missing: 0
Least: true (1310)
Most: false (1690)

FIGURE 19 – Result of the extracted sentiment with the threshold at 0.01

4 Conclusion