Application de techniques de Data Mining en utilisant le logiciel RapidMiner

Ce laboratoire a pour objectif d'expérimenter un logiciel de Data Mining, à savoir RapidMiner. Celui-ci va nous permettre de mettre en place rapidement et simplement diverses tâches de prétraitement, d'appliquer des algorithmes de classification, de clustering, etc. et d'évaluer les résultats obtenus.

1 Classification de spams

Ci-dessous, la matrice de confusion résultant de la classification des SMS, avec le classificateur *Naive Bayes*.

		Prédiction	
		Spam	**Non-spam**
Vérité	**Spam**	92	3
	Non-spam	38	263

Table 1 – Classification des SMS avec un filtrage Bayesien.

Ce qui donne une accuracy de **89.65%**.

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

Une étape de Stemming Porter (algorithme adapté à la langue anglaise) a été ajoutée après le filtrage des stopwords. Ceci ne change quasiment pas les résultats obtenus, comme le montre la matrice du confusion suivante :

		Prédiction	
		Spam	**Non-spam**
Vérité	**Spam**	91	4
	Non-spam	38	263

Table 2 – Classification des SMS avec Naive Bayes et Stemming.

Cette nouvelle étape amène juste le système à se tromper sur un SMS en ne le classant pas comme spam, ajoutant donc un Faux Négatif.

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

Tout d'abord, un *Arbre de décision* va avoir une excellente précision lorsqu'il va prédire un SMS comme étant un spam, il ne fait aucune erreur dans ce cas là. En revanche, il laisse passer beaucoup de spams en ne les détectant pas.

		Prédiction	
		Spam	**Non-spam**
Vérité	**Spam**	71	24
	Non-spam	0	301

TABLE 3 – Classification des SMS avec un arbre de décision.

Malgrès tout, l'arbre de décision permet d'arriver à une accuracy de **93.94%**, ce qui est sensiblement meilleur qu'un filtrage bayésien.

Nous avons ensuite changé le classificateur pour un *Réseau de neurones*.

		Prédiction	
		Spam	**Non-spam**
Vérité	**Spam**	88	7
	Non-spam	4	297

TABLE 4 – Classification des SMS avec un réseau de neurones.

En ne commettant que 11 erreurs, sur 396 SMS, le réseau de neuronnes arrive à la meilleure accuracy parmis les classificateurs testés, à savoir **97.22%**.

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

Nous avons au préalable eu à préprocesser les données des emails. Dans RapidMiner, nous avons ajouté un bloc permettant de transformer la colonne « spam » en type binomiale et de lui définir le rôle de label (blocs *Numerical to Binomial* et *Set Role*).
De plus, le contenu de la colonne avec le contenu des emails a été tronquée de « Subject : ». Ceci n'affecte évidemment pas le résultats, car il est présent dans chaque élément et n'est donc pas relevant et est ignoré par les algorithmes. Mais ce traitement permet de rendre le modèle réutilisable.

Avec un classificateur *Naive Bayes*, nous arrivons à la matrice de confusion suivante :

		Prédiction	
		Spam	**Non-spam**
Vérité	**Spam**	340	11
	Non-spam	38	1249

TABLE 5 – Classification des emails avec un filtrage Bayesien.

Avec une accuracy de **97.01%**, il s'agit d'un très bon résultat. Ajouter une étape de stemming n'améliore pas non plus le résultat sur ce type de données.

Nous avons également essayé d'utiliser un arbre de décision avec les emails. Celui-ci nous a permis d'arriver à une accuracy de **86.81%**, avec beaucoup de Faux Négatifs, ce qui est bien moins efficace. Un arbre de décision est de moins en moins efficace plus la taille et la complexité des données augmente.

		Prédiction	
		Spam	**Non-spam**
Vérité	**Spam**	166	212
	Non-spam	4	1256

TABLE 6 – Classification des emails avec un arbre de décision.

2 Market basket analysis

Pour la création des fichiers CSV utilisé pour la génération de règles d'associations, un nettoyage des données a été fait.
Les lignes avec des données vides, comme par exemple le nom du produit sont filtrées. Il en est de même si la qantité est plus petite que 0.

Un programme a été implémenté afin de détecter et filtrer les lignes avec des éventuels noms de produit incohérents. Voici une 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 quelques statistiques calculées lors du filtrage :
- Nombre de lignes dans le fichier : 541'909
- Nombre de lignes après nettoyage : 397'924
- Nombre de produits : 3'639

Le programme a donc retiré 143'985 lignes.

Un autre programme a été implémenté pour standardiser la description des produits en la liant avec le numéro du produit et celle qui est la plus utilisée, dans le cas un produit aurait plusieurs description.
Pour voir une partie de l'algorithme, il faut regarder la fonction *Main.resolveProductDescription()* dans le code Java fourni en annexe.

Ci-dessous, 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) :
- Nombre de lignes dans le nouveau fichier byCountry.csv : 37
- Durée du traitement : 00 :01.625

Création du fichier par client :
- Nombre de lignes dans le nouveau fichier byCustomer.csv : 4'339
- Durée du traitement : 01 :44.361

Création du fichier par facture :
- Nombre de lignes dans le nouveau fichier byInvoice.csv : 18'536
- Durée du traitement : 07 :26.901

Afin de pouvoir générer ces données nous avons utilisé du multi-threading ce qui nous permis d'améliorer considérablement le temps nécessaire à la création des fichiers.

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

Pour tester le programme, il suffit de lancer la commande *mvn clean install*, de modifier le fichier *application.properties* avec l'emplacement des fichiers de base et de destination.

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

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 données initiales qui produisent des règles intéressantes ?

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

Commentez les résultats obtenus par rapport aux étiquettes existantes sur le dataset.

Quelle est l'influence des différentes étapes de "text processing" sur le résultat que vous obtenez ?

4 Conclusion