UNIVERSITÉ MOHAMMED VI POLYTECHNIQUE

INSTITUTE OF SCIENCE, TECHNOLOGIE & INNOVATION (IST&I)

DÉPARTEMENT ALKHAWARIZMI

MASTER EN MODÉLISATION ET SCIENCES DES DONNÉES

2020-2022

Module : Machine Learning

Thème

Moroccan Darija Sentiment Analysis

(facebook comments dataset)

Préparé Par :

- LMHANI JABRAN

- MELLAL OMAR

Année universitaire : 2020-2021

REMERCIEMENT

Au moment où l'on pense avoir fait œuvre utile, qu'il nous soit permis d'exprimer nos vifs remerciements et notre profonde gratitude à notre chef de département monsieur Ahmad RATNANI, de nous avoir apporté un soutien déterminant et d'avoir témoigné un intérêt constant et particulier à notre projet de fin du module Machine Learning.

Nos remerciements vont aussi et de manière respectueuse et combien reconnaissante à notre encadrant Ali IDRI qui a bien voulu, en dépit de ses multiples engagements tant professionnels que personnels, accepter de diriger et de parrainer ce travail. Qu'il veuille bien nous permettre de rendre hommage à ses qualités exceptionnelles, à son savoir-faire, et à sa contribution bénéfique à la réalisation de ce projet.

Des remerciements spéciaux sont accordés au cadre professoral du Master en modélisation et science des données pour la formation de qualité qu'ils nous ont offert tout au long de cette année.

.

LISTE DES FIGURES

LISTE DES TABLEAUX

RESUME

*L'analyse d'opinions (**Fouille d'opinions**) dans les textes est un domaine de recherche à la croisée du traitement automatique des langues naturelles et de la recherche d'information qui suscite un intérêt croissant. Dans ce contexte, nous nous sommes intéressés à l'**analyse des sentiments** sur des posts Facebook, dans lesquels les opinions exprimées sont extraites puis **classifiées**. Ce présent rapport décrit le système réalisé et les résultats obtenus sur un Data-set récupéré à partir des commentaires Facebook.*

*Dans ce travail, nous allons utiliser trois **algorithmes** qui permettent d'analyser et **classifier** un ensemble de commentaires en dialecte marocain dérivées du réseau social Facebook, en l'occurrence : La **Régression Logistique**, le **Random Forest Classifier, le perceptron multicouche** et le **Naïve Bayes**. Les classes que nous avons définies sont: la classe positive ou négative.*

*Mots-clés : **fouille d'opinions**, **analyse des sentiments**, **classification**, **algorithmes**, **régression logistique**, **Random Forest Classifier**, **perceptron multicouche, Naïve Bayes**.*

ABSTRACT

Opinion analysis (search for opinions) in the texts is a field of research at an intersection between the automatic processing of natural languages and the search for knowledge that gives rise to growing interest. In this sense, we have been interested to analyze the feelings on Facebook postings, in which the views expressed are extracted and classified. This report describes the system achieved and results achieved on a data-set recovered from Facebook comments.

In this work, we will use three algorithms to analyze and classify a number of Moroccan comments from the social network Facebook: The Logistic Regression, The Random Forest Classifier, the multilayer perceptron and the Naives Bayes. The groups we described are: the positive, negative or neutral class.

Keywords: opinion analysis, feeling analysis, sorting, algorithms, logistical regression, Random Forest Classification, multilayer perceptron, Naïve Bayes.

LISTE DES ABREVIATIONS

HPW: Has Positive Word

HNW: Has Negative word

PWC: Positive Word Count

NWC: Negative Word Count

CL: Comment Lenght

SL: Sentiment Level

RF: Random Forest

LR: Logistical Regression

NB: Naive Bayes

MLP: Multilayer Perceptron

« *Le Machine Learning est autant un art qu'une science. C'est comme la cuisine – oui il y a de la chimie, mais pour faire quelque chose de vraiment intéressant, vous devez apprendre comment combiner les ingrédients à votre disposition* »

Greg Corrado, (Google)

TABLE DES MATIERES

CHAPITRE 1 : INTRODUCTION GENERALE

1 Cadre contextuel

Aujourd'hui, nos vies sont basées sur l'information et son analyse. Avec le développement du Web 2.0, ces informations ne sont plus disponibles et sont désormais fournies avec plus de précision sous forme numérique. Les gens communiquent de plus en plus sur divers sujets sur Internet dans des groupes de discussion, des blogs, des forums et d'autres sites liés aux critiques de produits, au partage de contenu et à l'expression d'opinions.

Fin 2013, Facebook a ouvert une page à recommander aux clients. Le produit sélectionné peut être noté par ses fans, allant de 1 à 5, où 1 est une polarité très négative et 5 est une polarité très positive. Les utilisateurs ont un impact significatif: l'enquête montre que la plupart des utilisateurs (80%) ont recherché des opinions sur des produits ou des services et qu'ils paient deux fois plus cher le produit ou le service. Les entreprises sont plus sûres que les autres entreprises, elles tiennent compte de ces paramètres et elles savent que l'analyse d'opinion est un élément important de la prise de décision.

Selon une étude de Pew Research Center, 20% des utilisateurs de médias sociaux ont changé leurs opinions politiques sur un sujet en raison de ce qu'ils voient sur les médias sociaux. Nous pouvons voir l'utilité de la détection d'opinion dans les domaines du marketing, de la politique, de la psychologie, de la santé, de la sécurité routière, du tourisme etc…

Le but de ce travail est d'étudier l'opinion publique, qui concerne principalement les sentiments tirés des commentaires Facebook rédigés en arabe (notamment le dialecte marocain ou le DARIJA).

Dans ce contexte, plusieurs travaux sont réalisés en tous les domaines d'application connus et avec différents sous-objectifs (construction de corpus, détection d'opinion, comparaison de fonctionnalités, application des

méthodes…etc.). Dans son ouvrage, Banfield (1982) a mis en évidence les éléments du langage qui conduisent à la subjectivité d'un texte ou tout du moins d'une partie d'un texte. Hatzivassiloglou et MckKeown (1997) sont les premiers à se pencher sur la classification des opinions. Pang (2002) a fait une étude qui permettait de classifier les sentiments de commentaires de films, il est le premier qui a expérimenté l'apprentissage automatique.

Notre rapport est divisé principalement en trois chapitres excepté celui-ci de l'introduction générale.

Dans le deuxième chapitre, nous nous focalisons sur l'état conceptuel, y compris la nomenclature des « Classifiers » et « Metrcis » de performances utilisées. Le troisième chapitre est consacré au processus d'expérimentation et aux détails de la modélisation de notre système. Le quatrième chapitre présente les résultats obtenus et la discussion de ces derniers. Enfin, nous mettons une conclusion et quelques perspectives.

2 Etat de fait, Objectif et problèmatique

L'analyse de sentiments, aussi parfois désignée sous le nom fouille d'opinions, est un sous-domaine de l'informatique, il est considéré comme une partie du traitement automatique du langage naturel et a pour but de classifier les sentiments exprimés dans des textes ; Comment on peut utiliser les ordinateurs pour mieux comprendre le langage naturel ? Comment peut-on réaliser une classification binaire sur des commentaires en Darija trop ambigus ? Quelles en sont les difficultés ?

2.1 Introduction

Les langues sémitiques (en référence au nom de "Sem" fils de "Noé") forment un ensemble de langues parlées depuis le plus ancien temps au Moyen-Orient, au Proche-Orient ainsi qu'en Afrique du Nord, La langue sémitique est une des branches de la famille des langues afro-asiatiques, où on ne sait pas de manière certaine l'origine et l'expansion géographique de

ces langues, soit de l'Asie vers l'Afrique ou le contraire. L'origine du mot "Arabe" reste inconnu, malgré il y a beaucoup de recherches, le radical "arab" désigne le désert et c'est un mot araméen "arâbâh" ; Le mot arabe peut dériver aussi de la racine sémitique Abhar "se déplacer", mais l'étymologie arabe considère que le mot "arabe" est dérivé du verbe "exprimer".

2.2 Voyellation

Dans la langue arabe nous avons deux types de signes de notation des voyelles, le premier type sont les voyelles brèves, qui sont notées au moyen de signes diacritiques, le deuxième type sont les lettres d'allongement d'une voyelle.

Les voyelles brèves sont : fatha, damma et kassra, se rajoutent sur les lettres, pour donner le vrai sens (la prononciation précise) de chaque mot et pour éviter les ambiguïtés (les conflits) lorsque le contexte ne suffit pas, il y a aussi d'autres diacritiques moins utilisés comme sukun, shadda et tanwn. Les voyelles longues (lettres d'allongement) sont : "Alif", "waw" et "ya", avec les sons respectivement "aa", "ou", "ii".

2.3 La richesse de la langue arabe

La langue arabe s'écrit au moyen de 28 lettres, le terme "abjad" désigne le système d'écriture, elle dispose du plus grand nombre de mots avec plus de 12 millions de mots contre 600 000 mots en langue anglaise, et 150 000 mots en langue français, et 130 000 mots en langue russe...

Elle est une langue très riche, Il y aurait 80 termes différents pour identifier le miel, 200 pour le serpent, 500 pour le lion, 1000 pour le chameau et l'épée et jusqu'à 4400 pour définir l'idée de malheur, les grammairiens arabes prétendent que toutes les racines ont été originalement des verbes, où le nombre de ces racines en réalité est de 6000.

2.4 Eléments de structure de la langue arabe

Comme toutes les langues sémitiques, l'arabe se caractérise par l'utilisation de certains schémas morphologiques (modèles de formation des mots), ces modèles permettent de fournir des mots à partir de racines abstraites, ces racines se composent d'habitude de trois consonnes qui forment les unités de base pour obtenir de nombreux mots dérivés de cette racine. On trouve par exemple la racine KTB se retrouve dans le verbe dérivé KaTaBa "à écrire" qui peut être conjugué en ajoutant des préfixes et des suffixes convenables. Le schéma KaTaBa peut être représentée par le schéma C1aC2aC3a où C1= K, C2 = T, et C3 = B. Le doublement de la consonne médiane de la racine donne "C1aC2C2aC3a" qui se traduit par le schéma KaTTaBa qui a le sens causal "a fait écrire". Par allongement de la première voyelle on obtient KaaTaBa, par l'ajout du préfixe "ta" nous aurons taKaaTaBa (notion de réciprocité). Chacun de ces schémas peut être modifié pour montrer le passé, le présent…, par exemple KuTiBa " été écrit", yaKTuBo "écrit" et yuKTaB "est écrit. Ainsi que le pluriel, par exemple Kutub "des livres", Kuttab "des écrivains", maKTaBaat "des bibliothèques", en plus, quelques-uns peuvent être mis au féminin par des terminaisons dédiées, par exemple KaaTiba "une femme écrivain", muKaaTiBa "correspondantes" et KaaTiBaat "des femmes écrivains". Ces exemples présentent un modèle riche et varié de schémas dérivés d'un seul mot qui permet d'engendrer des centaines de mots ont référencé à la même racine.

2.5 Darija marocain

Le Darija marocain ou bien le dialecte marocain, est un groupe d'Arabe Nord-Africain, dialectes mélangés avec différentes langues parlées au Maroc, le frottement de plusieurs langues à travers l'histoire de la région a produit un langage complexe et riche comprenant des mots, des expressions et des structures linguistiques, ces langues tel que le Berbère, le Français, l'Italien, l'Espagnol et le Turc ainsi que d'autres langues romanes méditerranéennes. Le dialecte Marocain est fortement influencé particulièrement par le français.

Nous pouvons catégoriser les paroles en ce dialecte comme la suite :

- MSA encodé en lettres arabiques, par exemple : « عيدكم مبارك كل عام وانتم بخير »
- MSA encodé en lettres romanisées, par exemple : « ana said hada elyaoum »
- Dialecte encodé en lettres arabiques, par exemple : « كيراك شريكي »
- Dialecte encodé en lettres romanisées, par exemple : « maysoumouch bla shour »
- Langue étrangère encodée en lettres romanisées, par exemple : « je vais à l'univ »
- Langue étrangère encodée en lettres arabiques, par exemple : « برافووو »

Donc chaque texte est l'une de ces catégories ou un mélange d'eux, on peut aussi trouver une autre écriture particulière, celle de l'utilisation d'un numéro au lieu d'une lettre par exemple : « b1-بيا -bien », « 3adi-عادي -normal », « 5amsa-خمسه -cinq », « 6ariq-طريق -route », « sa7b-صاحب - ami », « 9raya-قراية - étude »...

2.6 Les difficultés de l'arabe et du dialecte

Nous pouvons citer les difficultés suivantes :

- Un seul dialecte (Darija) peut contenir plusieurs sous dialectes.
- La grande distance entre MSA et quelques dialectes.
- Une racine peut prendre plusieurs formes en fonction du contexte.
- La répétition d'une lettre plusieurs fois pour intensifier le sens ou le sentiment (bezeeeef).
- L'arabe a divers signes diacritiques, la présence ou l'absence de tels signes, peut changer totalement le sens des mots.

2.7 Conclusion

Dans ce petit arc, nous avons élucider l'objectif de ce projet, puis nous avons scruté l'histoire de notre langue et quelque caractéristiques, sa richesse de vocabulaire et le plus important, sa multitude d'idiomes et surtout de façon particulière comme nous avons au Maroc, où presque chaque région à son propre accent, c'est une des principales raisons du manque des travaux qui utilisent le dialecte Marocain.

CHAPITRE 2 : CADRE CONCEPTUEL

1 Nomenclature des notions

Avant de détailler le processus d'expérimentation, nous présentons une nomenclature des notions qui sont nécessaires pour comprendre le travail effectué.

- La Régression Logistique : ou modèle logit est un modèle de régression binomiale. Comme pour tous les modèles de régression binomiale, il s'agit de modéliser au mieux un modèle mathématique simple à des observations réelles nombreuses. En d'autres termes d'associer à un vecteur de variables aléatoires une variable aléatoire binomiale. La régression logistique constitue un cas particulier de modèle linéaire généralisé. Elle est largement utilisée en apprentissage automatique.

- Forêt d'Arbres Décisionnels (RF Random Forest) : Le terme Forêts Aléatoires est un ensemble d'arbres de décision. Où chaque arbre a pour but de décomposer le problème en suite de tests correspondant à partitionner l'espace de données en sous-régions homogènes (en terme de classe), aller de la racine à une feuille en effectuent les tests de noeuds où la classe d'une feuille est la majoritaire parmi les exemples d'apprentissage appartenant à cette feuille.

- Naïf Bayésien (NB Naive Bayes) : Un classificateur naïf de Bayes est un classificateur probabiliste basé sur l'application du théorème de Bayes avec l'hypothèse naïve (forte indépendance), c'est-à-dire que les variables explicatives (Xi) sont supposées indépendantes conditionnellement à la variable cible (C), il appartient à la famille des classificateurs linéaires (son rôle est de classer dans des classes les échantillons qui ont des propriétés similaires). Ce classificateur est souvent utilisé sur les flux de données pour la classification supervisée.

- Multilayer Perceptron (MLP) : Le perceptron multicouche est un type de réseau neuronal artificiel organisé en plusieurs couches au sein desquelles une information circule de la couche d'entrée vers la couche de sortie uniquement ; il s'agit donc d'un réseau à propagation directe (feedforward).

- Mesures de performance : Soit :

  - ✓ PS : documents pertinents sélectionnés,
  - ✓ PN : documents pertinents non sélectionnés,
  - ✓ NS : documents non pertinents sélectionnés,
  - ✓ NN : documents non pertinents non sélectionnés.

- Rappel (recall) = PS / (PS + PN)
- Précision = PS/(PS + NS),
- F-mesure (F-score) = 2*précision * rappel/(précision + rappel)
- Accuracy = (PS + NN) / (PS + NS + PN + NN)

CHAPITRE 3 : PROCESSUS D'EXPÉRIMENTATION

1 Introduction

Dans ce chapitre, nous commençons par décrire la source de données (Dataset) sur laquelle nos modèles seront appliqués. Puis, on passera au prétraitement, l'annotation du corpus, et la construction du lexique, opérations qui consomment beaucoup de temps et nécessitent un grand effort

2 Source de données

Pour gagner du temps, nous avons exploité un Dataset publié sur la plateforme Kaggle. Les auteurs de ce Dataset ont construit leur propre corpus en dialecte marocain, par le développement d'un outil avec le langage de programmation Python, qui permet d'interroger l'API de Facebook pour récupérer des postes et des commentaires.

Leur Dataset est un mélange de textes avec des sentiments positifs ou négatifs.

7,سي جلول يعطيك ألف صحة و ربي يبارك فيك وكأنك شرهة في شباب باش يزيد يقرأ أكثر,Positive
8,Ya mama mali masta lasta :p :p,Negative
9,Bravo Neji jelloul,Positive
10,نسحو يفهم في كل شيء,Positive
11,Rak taaafeh wkan tnaber beeef,Negative
12,Weld asel et metrobi,Positive
13,Dima classe fi kemla mahlek,Positive
14,ma7léhom <3,Positive
15,Berrasmi khiit lotfi,Negative
16,mella marek krahnekom,Negative

Tableau 1: Aperçu du DataSet utilisé

3 Opérations de prétraitement

3.1 Dataset Cleaning

Elimination de la ponctuation :

'"`÷×؛<>_()*&^%][ـ/:"ـ“…”!|+¦~{}',.؟"'

Elimination des caractères spéciaux :

ً | # Shadda ٌ | # Fatha ٍ | # Tanwin Fath ُ | # Damma
ٌ | # Tanwin Damm ِ | # Kasra ٍ | # Tanwin Kasr ْ | # Sukun

Elimination des elongations :

"إأآا", "ا", "ى", "ي", "ؤ", "ء", "ئ", "ء", "ة", "ه", "گ", "ك",

3.2 Dataset Reduction

On a procédé comme suit à la réduction du DataSet :

- ✓ Elimination des stops-words
- ✓ Elimination des mots peu fréquents

3.3 Dataset Transformation

Nous avons utilisé pour transformer le DataSet le TF-IDF qui est une abréviation pour « Term Frequency Inverse Document Frequency»

Text → vector
Document → numeric matrix

Par exemple :

data = ['my', 'name ', 'is',omar’] --- tfidf vectorize --- >

my 1.916290731874155
name 1.916290731874155
is 1.916290731874155
omar 1.916290731874155

3.4 Dataset Balancing

Pour le data balancing nous avons fixé la longueur des variables on ajoutant des « batchs » .

Hello my name is omar
Good morning

→ <bat> Hello my name is omar <bat>
→ <bat> Good morning <bat> <bat> <bat> <bat>

4 Implémentation

4.1 Algorithme

Dans cette partie, nous allons implémenter les algorithmes d'analyse de sentiments. Nous allons principalement implémenter quatre algorithmes. Ces algorithmes sont : Régression logistique, Random Forest (RF), Perceptron multicouche et Naïve Bayes (NB). Nous allons utiliser des publications qui ont été normalisées et un corpus prêt à utiliser. Enfin, nous allons discuter les résultats de la classification.

Nous allons utiliser l'algorithme général de l'analyse suivant :

Algorithme

Importer les bibliothèques

Lire le Dataset

Lire le dictionnaire

Début

Extraire les fonctionnalités

Préparation

Appeler aux classificateurs

Afficher les résultats

Fin

4.2 Ressources utilisées

Dans notre expérimentation, nous avons utilisé deux PCs, le premier était de marque HP, le deuxième était de marque SONY VAIO, avec des

processeurs I3, I7, des horloges de fréquence de 4.40 GHZ et des RAM de 8 GO.

Pour la programmation, nous avons utilisé l'environnement Python. Python est un langage de programmation portable, dynamique, extensible, gratuit, d'une syntaxe très simple, orienté objet, évolutif ….

Egalement, nous avons utilisé les packages suivants :

- ✓ **CSV** : Une bibliothèque avec laquelle nous pouvons manipuler les fichiers de format csv.
- ✓ **Gensim** : Est une bibliothèque gratuite Python pour l'extraction automatique des sujets sémantiques des documents.
- ✓ **Scikit-learn** : Est une bibliothèque d'apprentissage automatique en Python, c'est le moteur qui alimente de nombreuses applications de l'intelligence artificielle et de la fouille des données.

4.3 Exemples de code sources

Dans cette section, nous allons présenter quelques exemples de codes sources.

La Figure 1 présente un morceau de code qui permet d'appeler les bibliothèques nécessaires pour compiler notre code.

```
[ ] import pandas as pd
    import numpy as np
    import matplotlib.pyplot as plt
    import string
    import re
    from nltk.corpus import stopwords
    from sklearn.model_selection import train_test_split, GridSearchCV
    from sklearn.feature_extraction.text import TfidfVectorizer
    from sklearn.pipeline import make_pipeline
    from sklearn.linear_model import LogisticRegression
    from sklearn.ensemble import RandomForestClassifier
    from sklearn.naive_bayes import MultinomialNB
    from sklearn.svm import SVC
    from sklearn.metrics import confusion_matrix,accuracy_score, classification_report
```

Figure 1 Appel des bibliothèques

```
[ ] import csv

    Train= pd.read_csv('Train.csv',
                       lineterminator='\r')
    Test= pd.read_csv('Train.csv',
                       lineterminator='\r')
```

Figure 2 Lecture du DataSet

```
[ ] n_pos =0
    n_neg =0


    for i in range(0,len(Train)):
        if (Train.iloc[i]['Sentiment']== 'Positive') : n_pos=n_pos+1
        else : n_neg=n_neg+1

    print(n_pos,"    ",n_neg)

    4382      4020
```

Figure 3 Comptage du nombre de commentaires positifs et négatifs

```
# splitting into train and tests
#X_train, X_test, Y_train, Y_test = train_test_split(feature, target, test_size =.2, random_state=100)
X_train, Y_train= Train.text.apply(str),Train.label.apply(str)
X_test, Y_test =Test.text.apply(str) ,Test.label.apply(str)


# make pipeline
pipe = make_pipeline(TfidfVectorizer(),
                     LogisticRegression(max_iter=1000),
                     verbose=True)
# make param grid
param_grid = {'logisticregression__C': [0.01, 0.1, 1, 10, 100]}

# create and fit the model
model = GridSearchCV(pipe, param_grid, cv=5)
model.fit(X_train,Y_train)

# make prediction and print accuracy
prediction = model.predict(X_test)
print(f"Accuracy score is {accuracy_score(Y_test, prediction):.2f}")
print(classification_report(Y_test, prediction))
```

Figure 4 Implémentation de la Régression logistique

Random forrest

```
[ ] pipe = make_pipeline(TfidfVectorizer(),
                         RandomForestClassifier())

    param_grid = {'randomforestclassifier__n_estimators':[10, 100, 1000],
                  'randomforestclassifier__max_features':['sqrt', 'log2']}

    rf_model = GridSearchCV(pipe, param_grid, cv=5)
    rf_model.fit(X_train,Y_train)

    prediction1 = rf_model.predict(X_test)
    print(f"Accuracy score is {accuracy_score(Y_test, prediction):.2f}")
```

Figure 5 Implémentation du Random Forest

naive bayes

```
[ ] pipe = make_pipeline(TfidfVectorizer(),
                         MultinomialNB())
    pipe.fit(X_train,Y_train)
    prediction2 = pipe.predict(X_test)
    print(f"Accuracy score is {accuracy_score(Y_test, prediction):.2f}")
    print(classification_report(Y_test, prediction))
```

Figure 6 Implémentation du Naïve bayes

```
pipe = make_pipeline(TfidfVectorizer(),
                     MLPClassifier(solver='lbfgs'))

clf = MLPClassifier(solver='lbfgs', alpha=1e-5,
                    hidden_layer_sizes=(5, 2), random_state=1)
pipe.fit(X_train,Y_train)
prediction2 = pipe.predict(X_test)
print(f"Accuracy score is {accuracy_score(Y_test, prediction2):.2f}")
print(classification_report(Y_test, prediction))
```

Figure 7 Implémentation du perceptron multicouche

4.4 Fonctionnalités

Nous avons basé notre étude sur 06 fonctionnalités qui sont :

- ✓ l'existence de mots positifs (et/ou négatifs),
- ✓ le nombre de mots positifs (et/ou négatifs) dans le commentaire,
- ✓ la longueur du texte,
- ✓ le niveau du sentiment.

A dessein d'appliquer la fonctionnalité du niveau de sentiment nous avons utilisé le document (SemEval2016 Arabic Twitter Sentiment Lexicon). La figure 7 présente une partie de ce document.

1	0.963	حب
2	0.925	فرح
3	0.925	نجاح
4	0.912	سرور
5	0.912	مبروك
6	0.900	صادق
7	0.900	متفائل
8	0.900	سعادة
9	0.875	نعيم
10	0.875	مبدع
11	0.875	سلام
12	0.838	رائع

1	0.000	وكيل
2	0.000	عند
3	0.000	احصي
4	0.000	منزمان
5	0.000	ليل
6	0.000	دخل
7	0.000	مكافحة

1	-0.087	غبن
2	-0.250	تانيب
3	-0.275	طفش
4	-0.287	غصة
5	-0.287	دمع
6	-0.287	زكام
7	-0.300	افقد
8	-0.313	هروب
9	-0.325	صعب
10	-0.375	عيب
11		

Figure 8 Exemple de niveau de sentiment

CHAPITRE 4 : DISCUSSION DES RÉSULTATS

1 Résultats

Tant que nous utilisons la méthode d'apprentissage supervisé et l'approche hybride, nous avons divisé le corpus en deux parties, 80% pour l'entrainement et 20% pour le test. Nous avons fait plusieurs tests, et les résultats Accuracy sont présentés dans le tableau ci-dessous.

Test	Fonctionnalités/Classificateur	RF	LR	NB	MLP
1	Toutes les fonctionnalités	84,31	90,00	89,28	83,13
2	HPW, HNW, PWC, NWC	84,28	84,11	82,38	81,02
3	PWC, NWC, CL, SL	85,31	84,62	83,76	82,23
4	PWC, NWC	84,45	84,11	67,87	65,76
5	HPW, HNW, CL, SL	84,11	84,11	48,35	44,87
6	HPW, HNW	84,11	84,11	72,30	69,33
7	CL, SL	48,18	47,49	48,18	44,90

Tableau 2 Résultats de classification

Le tableau ci-dessous, présente en détail les résultats de classification avec l'utilisation de toutes les fonctionnalités :

MODEL	ACCURACY SCORE		PRECISION	RECALL	F1-SCORE	CONFUSION_MATRIX	
						FALSE	TRUE
Logistic regression	0.90	N	0.87	0.93	0.90	758	58
		P	0.93	0.88	0.90	108	756
MLP	0.83	N	0.87	0.93	0.90	655	161
		P	0.93	0.88	0.90	22	842
Random Forrest	0.84	N	0.95	0.70	0.80	568	248
		P	0.77	0.97	0.86	27	837
naive bayes	0.89	N	0.97	0.80	0.88	665	161
		P	0.84	0.97	0.90	22	842

Tableau 3 Résultats de classification détaillés avec toutes les fonctionnalités

2 Discussion

Le meilleur résultat dans tous les tests est 90.00%, il a été obtenu par la régression logistique avec l'utilisation de toutes les fonctionnalités, c'est la même chose pour RF, MLP et NB, leurs meilleurs résultats étaient avec le premier test (85.14%, 84.28% respectivement).

D'après les tests (2), (4) et (6) nous avons remarqué que les deux couples (PWC, NWC) et (HPW, HNW) ont eu presque le même poids d'influence, c'est logique parce que le nombre de commentaires mixtes (qui contiennent des mots positifs et au même temps des mots négatifs) est usuellement petit, c'est pour ça quand nous avons exploité le nombre de mots d'une polarité, la mesure était un peu plus grande (de 84.11% à 84.45%) à part NB et MLP qui ont eu une variation considérable.

Selon les tests (3) et (4), nous avons constaté que l'ajout des fonctionnalités (CL, SL) aux fonctionnalités (PWC, NWC) a amélioré les résultats de la classification.

Dans les tests (5) et (6), nous avons constaté que l'ajout des fonctionnalités (CL, SL) aux fonctionnalités (HPW, HNW) n'a fait rien sauf que la mesure de NB et MLP a diminué.

D'après le dernier test, nous avons trouvé que les fonctionnalités (CL, SL) ne peuvent pas être seules, puisque si c'est le cas, ils vont donner le plus mouvais résultat (moins de 49%).

Ces résultats montrent que la Régression Logistique est généralement considéré comme un meilleur classificateur.

3 Exemples de sortie

Malgré que nous ayons abouti à de très bonnes mesures, notre modèle a soulevé quelques erreurs. Ces erreurs sont listées dans le tableau ci-dessous :

Exemple	Commentaire	Notre annotation	Résultat du modèle
1	عيد سعيد و مبارك و كل عام و أنت بألف خير إن شاء الله	1	1
2	سفيان فيغولي يتطوع لإحدى المائدات الإفطارية للمسلمين في فرنسا ، برافو سفيان	1	1
3	هذا فعل لا اخلاقي اين تعليم سيدنا محمد(ص)عندما كان جاره يهودي وعندما مرت جنازة يهودي بالله عليكم ماذا فعل؟؟؟	-1	-1
4	مالقيتو ما ديرو جايبنا هاد خماج	-1	-1
5	انا باغي كاس رايب و ربع خبزة حسن من لفريت	1	1
6	الى عرفتيه في اقل من 5 ثواني دير جام	1	1
7	الجميع كيشهد أن # الدوون هو الافضل ولكن انا مبانش ليا	-1	1
8	حماق وتلاقاو	1	-1
9	أاش هاد اضحيكة	1	1
10	هذا الانسان ماشي رجل وطني	-1	1

Tableau 4 Exemples des résultats de l'analyse

Nous pouvons expliquer les causes de ces erreurs dans ce que suit.

1) Dans le premier exemple (7), l'erreur se produit par ce que le modèle compte dans le texte un mot positif 'الافضل ' et il le classe comme positif alors que l'existence d'un mot de polarité positive ne signifie pas toujours que le propos est positif entièrement.

2) Dans le deuxième exemple (8), nous avons annoté cette publication par positive, l'erreur survenait quand le système a trouvé le mot négatif 'حماق ' alors ce dernier ne rend pas toujours le texte d'un sentiment négatif.

3) Dans le troisième exemple (9), cette erreur parmi les erreurs qui sont difficiles à régler, on a le mot "اضحيكة" est un mot positif, mais le système ne l'a pas détecté car il est écrit de façon anormal (inattendue), la façon

attendue est ‘الضحيكة ’, puisque dans les réseaux sociaux chacun écrit le mot comme il veut, donc il est dur pour le modèle à analyser.

4) L’erreur du quatrième exemple (10) est due au contexte, où le contexte dit que cette phrase est négative, alors qu’elle n’a aucun mot négatif.

CONCLUSUION

Dans ce chapitre, nous avons fait l’analyse de sentiments sur un corpus qui contient 8402 commentaires en dialecte marocain étiquetées comme suivant : 4382 textes positifs, 4020 textes négatifs. Nous avons exploité quatre classificateurs d’apprentissage automatique qui sont la régression logistique, forêt d’arbres décisionnels (RF), perceptron multicouche et naïve bayes (NB) où l’évaluation de ces classificateurs se fait par 20% du corpus. Nous avons utilisé six fonctionnalités qui sont HasPositifWord, HasNegatifWord, PositiveWordCount, NegativeWordCount, CommentLength et SentimentLevel. Nous avons fait sept tests différents, le premier s’est fait en utilisant toutes les fonctionnalités et les autres tests se sont faits par la substitution entre ces fonctionnalités. Nous avons comparé les résultats de tests pour les quatres classificateurs. Nous avons trouvé que la bonne Accuracy (90.00%) est atteinte par le classificateur Régression Logistique. En fin, nous avons cité quelques exemples d’erreurs d’analyse dans notre modèle et nous avons expliqué comment le modèle les a faites.