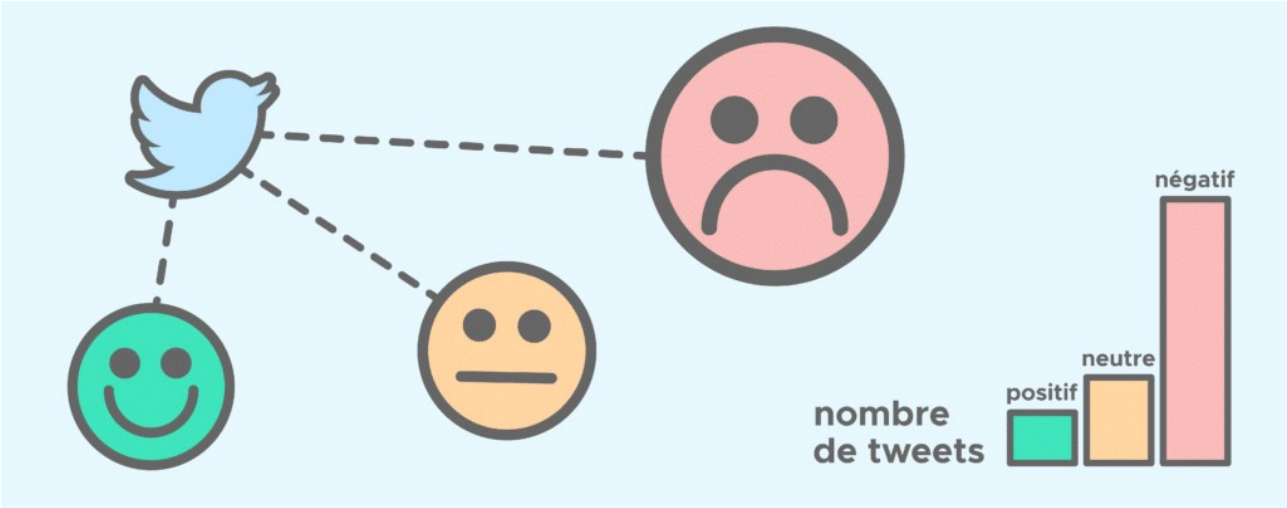
<u>Binôme :</u> Cloé D'HAILLECOURT & Ahmad ALMOSA ALMAKSOUR

PJE C
Analyse de comportements avec Twitter



M1 INFORMATIQUE, 2020

SOMMAIRE

DESCRIPTION DU SUJET

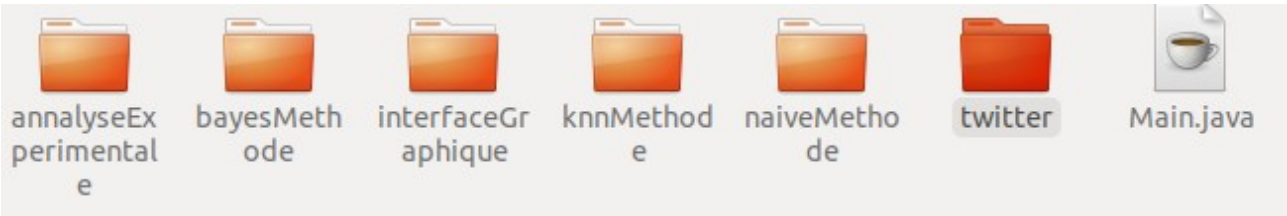
Ce projet encadré était sur le thème de l'analyse de comportements avec Twitter. À l'aide de l'API Twitter, nous devions parvenir à classer différents Tweets recherchés via un mot clé selon si ils sont 'positifs', 'négatifs' ou 'neutre'. Grâce à cela, nous pouvons déterminer le pourcentage de personnes qui ont un avis 'positif', 'négatif' ou 'neutre' sur les Tweets composés du mot clé entré au préalable. L'annotation des Tweets peut se faire à l'aide de diverses méthodes que nous développerons dans ce rapport. Le projet a été traité dans le langage de programmation java. Le résultat final est une interface graphique dans laquelle on peut entrer un mot clé, lorsque celui-ci est entré nous pouvons choisir la méthode de notation qui sera utilisée, une quinzaine de Tweets notés s'afficheront alors. De plus, le pourcentage de personnes qui ont un avis 'positif', 'négatif' ou 'neutre' sera affiché. Dans ce projet, il y a également une partie d'analyse expérimentale qui affiche le taux d'erreur de chaque méthode.

POURQUOI AVOIR CHOISI CE PJE ?

Ce PJE met en lumière le Machine Learning et la Data Sciences et nous étions tous deux intéressés par ces domaines. De plus, c'est un projet très connu et qui pourrait être un atout pour notre CV car les entreprises aiment ce type de projet. Avant toute chose, c'est tout simplement un projet qui a attiré notre attention et attisé notre curiosité.

DESCRIPTION GÉNÉRALE DE L'ARCHITECTURE DE NOTRE APPLICATION

Le dossier de notre projet se nomme 'pje_twitter_dhaillecourt_almosaalmaksour', vous y trouverez directement ce rapport 'RAPPORT_PJE.pdf'. Le dossier 'target' est composé de tous nos fichiers class. Notre projet java se trouve dans le dossier 'src/main/java'.



Notre projet utilise trois méthodes de classification différentes et pour certaines, des sous-méthodes de classification.

Dans le package 'naiveMethode' se trouve la méthode Naïve de classification.
Ce package est composé de deux classes et d'un dossier 'Liste'. Dans le dossier 'Liste' se trouve les deux fichiers 'positive.txt' et 'negative.txt' qui vont permettre d'annoter les Tweets à l'aide de la méthode naïve.
La classe ExtractorSentimatWorld permet de créer une liste des mots se trouvant dans les fichiers de 'Liste'.

La classe NaiveMethode prend en paramètres les fichiers de ‘Liste’ et permet de noter un Tweet avec la méthode naïve.

Le package knnMethode est composé de la classe KnnMethode qui classifie un Tweet avec la méthode KNN.

Le package bayesMethode est composé de la classe BayesMethode qui regroupe les fonctions communes au méthode de bayes (fréquence et présence.
La classe BayesMethodePresence permet de classer un tweet avec la méthode bayes présence.
La classe BayesMethodeFrequance permet de classer un tweet avec la méthode bayes fréquence.

Le package interfaceGraphique est composé des deux classes permettant de créer notre interface graphique.

Le package Twitter est composé d’une classe App qui va chercher dans l’API tweeter différents tweets.

Le package annalyseExperimentale est composé d’une class CrossValidation qui va permettre d’afficher de calculer le taux d’erreur de chaque méthode vue précédemment par validation croisée.

API Twitter

Nous avons téléchargé la librairie Twitter4J pour pouvoir accéder à l’API Twitter afin de faire la recherche des tweets et extraire les informations utiles.
Nous avons créé un compte développeur sur Twitter pour avoir les différents codes secrets qui nous permettront de faire l’authentification et pouvoir extraire les tweets de l’API.

```
package twitter;
import java.util.Scanner;


public class App{
    private String a;
    public  App(String a) throws TwitterException {
        this.a=a;
        ConfigurationBuilder cb = new ConfigurationBuilder();
        cb.setDebugEnabled(true)
        .setOAuthConsumerKey("AzgK1voXeIWmBFLKa6uMF13C1")
        .setOAuthConsumerSecret("PY1OJwUGOsynIc1zCm8C7f07Zok3Dr7l6jsbH7m7ImBB0dvljb")
        .setOAuthAccessToken("1305467268038635520-mwc0rY3Lf2Zam7mlc049Kp4ZEr011o")
        .setOAuthAccessTokenSecret("v8XEwErX1g2cjEP55v03aZBtLuqF1lHYRETg5Dfn2iqrZ")
        .setTweetModeExtended(true)
        .setHttpProxyHost("cache-etu.univ-lille1.fr")
        .setHttpProxyPort(3128);

        Twitter twitter = TwitterFactory.getSingleton();
```

NETTOYAGE DE LA BASE D'APPRENTISSAGE

Nous avons créé une fonction filtres dans la class App qui permet de filtrer chaque Tweet passé en paramètres.

```
public String filtres(String text) throws TwitterException {
    text = text.toLowerCase();

    Pattern p0 = Pattern.compile("\\b(https?|ftp|file)://[-a-zA-Z0-9+&@#/%?=~_|!:,.;]*[-a-zA-Z0-9+&@#/%=~_|]");
    Matcher m0 = p0.matcher(text);
    text=m0.replaceAll("");

    Pattern p = Pattern.compile("@\\w+ *");
    Matcher m = p.matcher(text);
    text=m.replaceAll("");

    Pattern p1 = Pattern.compile("#[-a-z]*");
    Matcher m1 = p1.matcher(text);
    text=m1.replaceAll("");

    text=text.replaceAll("\'", " ");
    text=text.replaceAll("’", " ");
    text=text.replaceAll("\"", " ");
    text=text.replaceAll("«", " ");
    text=text.replaceAll("»", " ");
    text= text.replaceAll("\n", " ");
    text=text.replaceAll("\t"," ");
    Pattern p2 = Pattern.compile("\\p{Punct}");
    Matcher m2 = p2.matcher(text);
    text=m2.replaceAll(" ");
    text=text.replaceAll("ç","c");
    return Normalizer.normalize(text, Normalizer.Form.NFD).replaceAll("[^\\p{ASCII}]", "");

}
```

Nous avons commencé par garder uniquement les Tweets de langue française. Ensuite, nous avons suivi les recommandations de nos professeurs et à l'aide des expressions régulières nous avons supprimer tous les @, les #, les liens.
Plus tard, dans la création de méthode nous nous sommes rendus compte que d'autres choses pouvaient poser problème. Nous avons donc décidé de convertir en caractères minuscules tous les Tweets passés en paramètres et également du supprimer absolument toutes les ponctuations, en effet, ces éléments nous gênait pour certaines comparaisons.
Ensuite, nous avons remarqué que notre code considéré que 'garcon' était différent de 'garçon' ce qui nous gênait. Nous avons donc supprimé tous les accents et les ç.

CONSTRUCTION DE LA BASE D'APPRENTISSAGE

Nous avons choisi d'utiliser un fichier CSV pour notre base d'apprentissage. Vous pouvez trouver ce fichier dans le dossier 'tweets', 'tweets_ManualMethod.csv'. La construction de la base d'apprentissage se fait dans la classe App. Pour chaque tweet de la base de données est recensé le ID de l'utilisateur, son nom, le Tweet, la date de création du Tweet, le mot clé

qui aura servi à la recherche de Tweet et enfin la notation de ce Tweet, tout d'abord initialisé à -1 qu'il faudra alors changé manuellement.

	B	C	D	E	F
1	JaxTellee:	les amis si ça peut vous rassurer peut etre qu on se retrouvera tous pour la plupart dans	Fri Nov 27 15:06:02 CET 2020	covid	4
2	imadabada4:	jvais m attacher au covid comme ça lui aussi il partira	Fri Nov 27 15:05:36 CET 2020	covid	0
3	weightlose51:	je suis grave malade j ai meme peur d avoir le covid et moi je fais qupi le tbc je veu	Fri Nov 27 15:05:29 CET 2020	covid	0
4	aminat_shh:	le test covid il fait mal wsh	Fri Nov 27 15:04:18 CET 2020	covid	0
5	OrbanDoc:	quid du suivi des sequelles respiratoires post covid et le plus souvent post hospit en m	Fri Nov 27 15:03:23 CET 2020	covid	0
6	barberousseomax:	qu est ce que j ai envie de retrouver une vie normale sans covid	Fri Nov 27 15:02:27 CET 2020	covid	0
7	camsidie_:	je rentre a nouveau dans ma periode ou y a plus rien qui m interesse genre quoi que je fa	Fri Nov 27 15:01:55 CET 2020	covid	0
8	MxxxRayane:	j pense si tu fais ecouter stamina de au covid il part en corbillard direct en enfer le job es	Fri Nov 27 15:01:16 CET 2020	covid	0
9	lauraapmo:	j avais oublie a quel point c est pas agreable le test covid	Fri Nov 27 14:57:49 CET 2020	covid	0
10	pichettepaillet:	quelle mediocrite deux colis qui se baladent dans la nature et ne me sortez pas l excuse	Fri Nov 27 14:56:02 CET 2020	covid	0
11	cpa_breda:	27 11 20 14 55 51 0126501 ambulance pour un covid 19 43 87 2 193	Fri Nov 27 14:55:56 CET 2020	covid	2
12	psgeneve:	le gel hydroalcoolique est un indispensable dans la lutte contre la covid 19 le ps ville de	Fri Nov 27 15:12:02 CET 2020	covid	2
13	KonamaKoprowski:	a propos du vaccin du covid se serai bien et s ils en avaient le courage et la franchise	Fri Nov 27 15:12:01 CET 2020	covid	0
14	OhMyVHope:	je suis deja pas super fan de mes cours d allemand alors savoir que le partiel sera majo	Fri Nov 27 15:10:09 CET 2020	covid	0
15	CCabimael91:	les delegues en sueur quand ils devront accompagner leurs camarades malades avec su	Fri Nov 27 15:09:47 CET 2020	covid	0

A la création de notre base, nous avons rencontrés plusieurs problèmes. En effet, il fallait déterminer un séparateur pour les colonnes et un séparateurs pour les lignes. Or, notre base à ce moment là n'était pas encore filtrée correctement. Nous avons donc décidé d'ajouter des filtres sur les tabulations et saut de lignes afin d'utiliser comme séparateur de colonne une tabulation et comme séparateur de ligne un saut de ligne.
Notre base est constitué de 181 Tweets choisis avec les mots clés 'noël', 'covid' et 'argent'. Nous avons également récupérer la base de données du binôme Guillaume & Etienne 'tweets_DB.csv' qui a été formé un petit peu différemment mais que nous avons adapté à notre code. Elle est composée de 321 Tweets obtenus avec les mots clés 'Microsoft', 'Vacances', 'Luigi', 'NBA', 'Playstation', 'Apple', 'PS5', 'asso', 'sexion', 'informatique', 'NIVIDIA', 'AMD', 'Kinder', 'Noel', 'Amazon', 'Pétrole', 'Télévisions', 'Musique', 'Cyberpunk'.*

L'ANNOTATION

Nous avons décider de noter les Tweets positifs avec 4, les Tweets neutres avec 2 et enfin, les Tweets négatifs avec 0.

LA MÉTHODE NAÏVE

Pour la méthode naïve, nous avons deux listes de mots dans les deux fichiers respectifs 'positive.txt' et 'negative.txt'. Ces deux fichiers nous étaient fournis mais nous avons pris la décision de retirer ce qui ne nous semblait pas cohérent et d'ajouter nous-même quelques points. Comme énoncé plus tôt nos avons décidé de garder uniquement les Tweets français nous avons donc supprimé la première partie des deux fichiers qui étaient composés de listes de mots de langue anglaise, cependant, nous avons gardé quelques mots anglais souvent utilisés en français dans notre liste. Pour définir la note d'un Tweet, il faut compter le nombre de mots du Tweet se trouvant dans la liste 'positive.txt' et le nombre de mots se trouvant dans la liste 'negative.txt'. Si le compteur positif est supérieur au compteur négatif alors la notation sera positive est vice-versa. En cas d'égalité, la notation sera neutre.
La première difficulté est de convertir la liste de mots du fichier en liste java. Ceci se fait dans la classe ExtractorSentimentWorld. Nous avons bien veillé à supprimer les espaces entourant le mot dans la liste.

Ensuite, la classe NaiveMethode prend en paramètres les deux listes établies par la classe ExtractorSentimentWorld. Puis la note est calculée avec la méthode expliquée précédemment.

LA MÉTHODE KNN

Pour la méthode KNN, le but est de prendre les k voisins les plus proches du Tweet passé en paramètres et leur notation afin de déterminer sa notation. La class KnnMethode permet de calculer la distance entre deux Tweets puis de donner sa notation.
Nous avons calculé manuellement la qualité de classification de KNN avec k = 5.

		Classe estimée		
		Positif	Négatif	Neutre
Classe réelle	Positif	6	16	1
	Négatif	3	21	2
	Neutre	1	9	3

LA MÉTHODE BAYES

Cette méthode se base sur trouver la classe la plus probable pour un tweet donné en appliquant le théorème de Bayes.

On a créé plusieurs types pour la méthode bayésienne, on a créé une classe pour classer les tweets en prenant compte que la présence des mots dans le tweet, on ne se préoccupe donc ni l'ordre des mots, ni de leur organisation dans le tweet, ni de leur nombre d'occurrences.

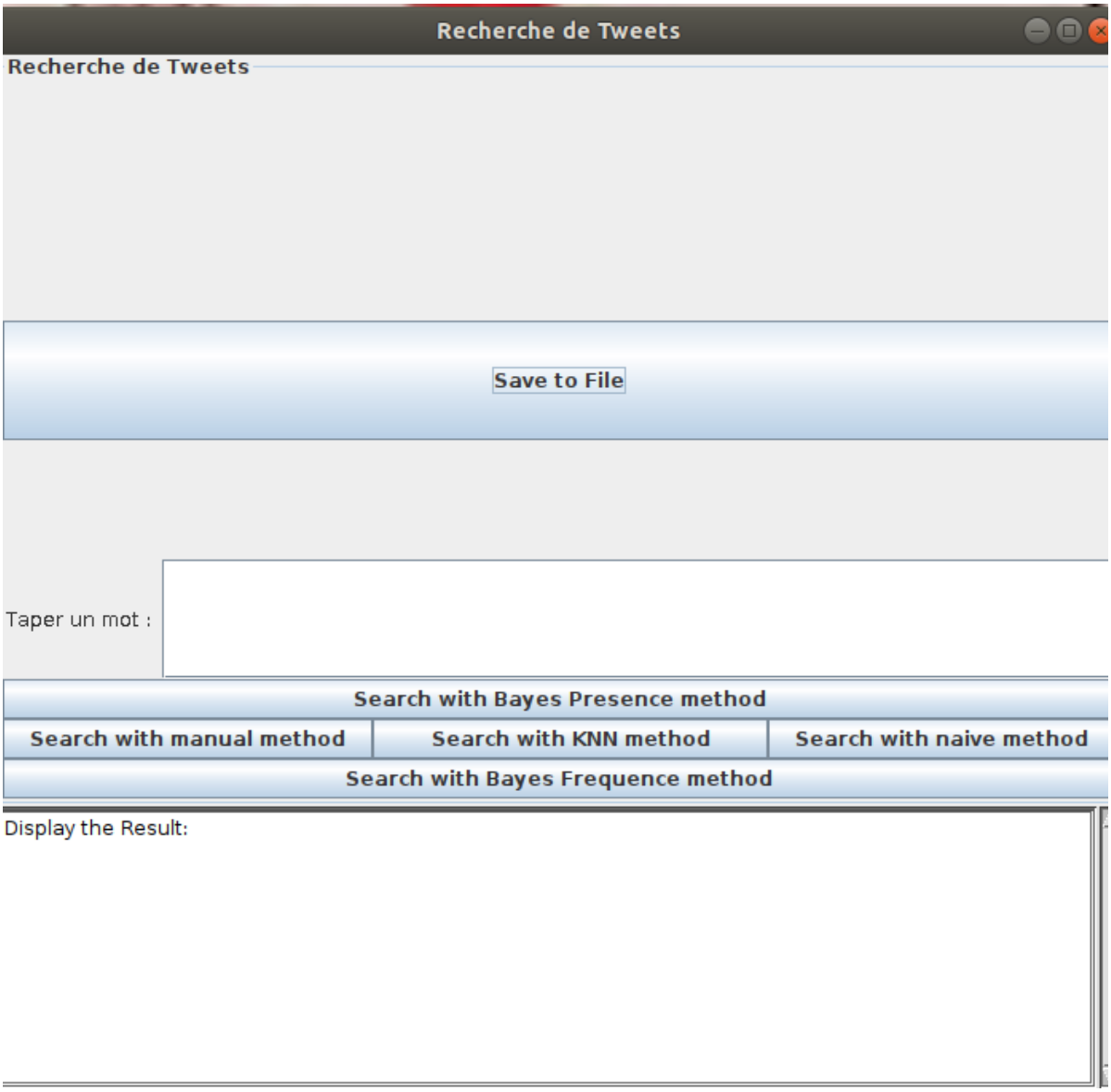
Ensuite on a classé les tweets selon le nombre d'occurrences pour chaque mot présenté dans le tweet et on a adapté l'équation de la probabilité.

On a fait le choix entre choisir tous les mots dans le tweet et calculer la probabilité dans par rapport à notre base de données et choisir que les mots qui sont de longueur supérieure à 3.

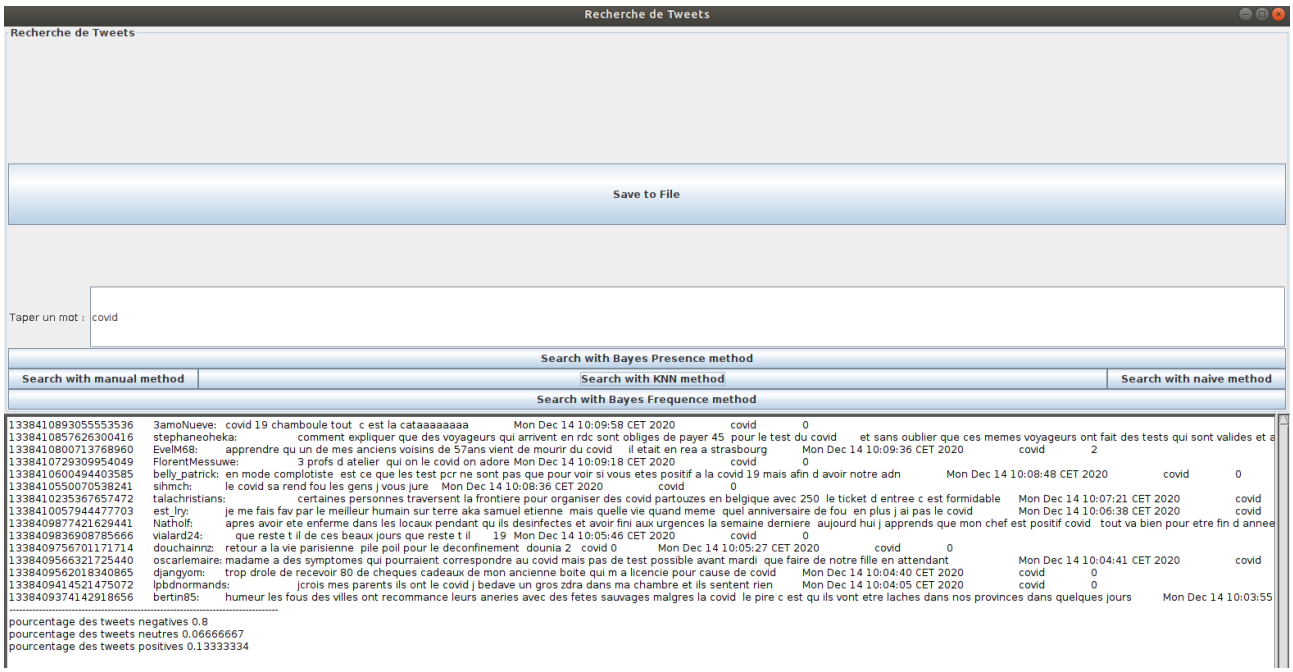
On a modifié notre algorithme afin de considérer les bi grammes (deux mots ensemble) et uni-grammes + bi-grammes.

L’INTERFACE GRAPHIQUE

Voici l’affichage de notre interface graphique :



Notre interface graphique permet d’entrer un mot clé dans la case ‘Taper un mot’, suite à cela, vous allez rechercher les Tweets en choisissant la méthode avec laquelle vous voulez qu’ils soient notés parmi les différentes méthodes vues précédemment.
Votre interface affichera alors une quinzaine de Tweets notés avec la méthode choisie avec également l’id et le nom de l’utilisateur, la date exacte du Tweet et le mot clé choisi.
L’interface graphique affichera également la tendance des Tweets en ce moment par rapport au mot clé entré en paramètres. Autrement dit, elle affichera parmi la quinzaine de Tweets, le pourcentage de Tweets positif, négatif ou neutre.
Par exemple, si vous entrez le mot ‘covid’ puis cliquez sur ‘search with KNN methot’ voici l’affichage que vous aurez :



L'interface offre également une possibilité de sauvegarder les résultats dans un fichier propre à chaque méthode en cliquant sur 'Save to file'. Dans notre exemple, en cliquant sur 'Save to file', si le fichier 'KNNMethode.csv' existe déjà nos résultats vont s'enregistrer à la suite des Tweets déjà présent, sinon il va créer ce fichier 'KNNMethode.csv' et y enregistrer les résultats.

LA CLASSIFICATION

Nous avons utilisé la méthode de Cross Validation pour la classification de nos méthodes. Ceci consiste à découper notre base de données en k sous-ensembles disjoints de même taille. Nous avons choisi de prendre un k = 10.
Dans la classe CrossValidation, ce découpage de la base de données va se faire aléatoirement ce qui va engendrer que pour une même base des résultats de classification peuvent varier.
Dans la class CrossValidation, il y a une fonction d'affichage des résultats de la validation croisée que nous avons mis dans le Main, nous avons pris la décision de ne pas l'afficher directement dans l'interface graphique.
Voici les résultats que nous avons eu avec notre base de données.

```
La taux d'erreur pour La classification KNN est : 0.27222222222222225
------------------------------------------------------------------------
La taux d'erreur pour La classification Naive Methode est : 0.6388888888888888
------------------------------------------------------------------------
La taux d'erreur pour La classification Bayes Presence uni gramme est : 0.7222222222222221
La taux d'erreur pour La classification Bayes Presence bi gramme est : 0.6777777777777778
La taux d'erreur pour La classification Bayes Presence uni gramme et bi gramme est : 0.7
------------------------------------------------------------------------
La taux d'erreur pour La classification Bayes Frequance uni gramme est : 0.7611111111111112
La taux d'erreur pour La classification Bayes Frequance bi gramme est : 0.6
La taux d'erreur pour La classification Bayes Frequance uni gramme et bi gramme est : 0.5722222222222222
```

On remarque rapidement que le méthode KNN est la meilleure. Toutes les autres méthodes sont très moyennes et il reste un grand taux d'erreur.
La méthode naïve donne un taux d'erreur aux alentours des 60%.
On ne trouve pas une très grande différence avec les variantes de la méthode Bayes. De plus, les résultats peuvent varier avec une même base de données comme expliqué précédemment.
Nous avons comparé nos résultats avec une autre base de données :

```
La taux d'erreur pour La classification KNN est : 0.371875
------------------------------------------------------------------------
La taux d'erreur pour La classification Naive Methode est : 0.603125
------------------------------------------------------------------------
La taux d'erreur pour La classification Bayes Presence uni gramme est : 0.675
La taux d'erreur pour La classification Bayes Presence bi gramme est : 0.73125
La taux d'erreur pour La classification Bayes Presence uni gramme et bi gramme est : 0.66875
------------------------------------------------------------------------
La taux d'erreur pour La classification Bayes Frequance uni gramme est : 0.7944444444444444
La taux d'erreur pour La classification Bayes Frequance bi gramme est : 0.5
La taux d'erreur pour La classification Bayes Frequance uni gramme et bi gramme est : 0.6722222222222222
```

On remarque que les résultats varient un peu mais restent tout de même similaires.

CONCLUSION

Nous avons beaucoup aimé ce projet. Nous avons pu travailler ensemble malgré la situation, et cela nous a permis de fusionner nos idées et mieux avancer dans le projet. La plupart des tâches ont été réalisées avec succès. Malgré tout, nous sommes un peu déçu car notre taux d'erreur reste relativement grand. On conclue de cette expérience que le travail d'équipe prime et que les idées des uns et des autres sont toujours bonnes pour être entendues.
Merci à nos professeurs pour ce projet.