LA Joséphine

2021-2022 **SITBON Benjamin**

Rapport du projet Data Science

I)Introduction

Pour le choix de notre projet, nous avons choisi un domaine qui nous intéressait mais qui était encore tout nouveau pour nous, le Natural Langage Processing (NLP). Cette branche de l'intelligence artificielle consiste à traiter du langage humain, ainsi que le traitement de ce langage grâce à différents outils informatiques. On utilise le NLP au quotidien, comme pour faire une traduction, où intervient la compréhension afin de comprendre le sens d'un texte dans sa langue originale puis la génération de texte dans une langue cible, ou bien par exemple pour faire un référencement de sites web, comme le permet l'algorithme mis en place par Google BERT. Du fait de l'unicité de chaque langage, chaque modèle doit être ré-entrainé pour chaque langue, la langue la plus avancée étant l'Anglais. Nous avons donc utilisé des données dont le contenu est exclusivement en Anglais.

II)Mental Health Chatbot

Depuis quelques années, les chatbots sont devenus de plus en plus populaires. Un chatbot est un programme informatique qui simule et traite une conversation humaine, qui peut être écrite ou parlée, et permettant aux humains d'interagir avec des terminaux digitaux comme s'ils communiquaient avec une personne réelle.

Nous avons pensé qu'il pouvait être intéressant de découvrir ce milieu, comment fonctionne un chatbot, et comment appliquer du Machine Learning afin de pouvoir interagir de manière autonome avec un chatbot.

De nombreuses bibliothèques sont utilisées afin de traiter les problèmes qui sont basés sur du NLP. Nous allons utiliser principalement **NLTK**, mais on pourrait également se servir de **Spacy**, qui sont des bibliothèques de niveau supérieur utilisées pour effectuer des tâches en langage naturel telles que la suppression de mots vides, la reconnaissance d'entités, la correspondance des phrases, etc... Nous utiliserons également **numpy**, **pandas**, **scikit learn** , **matplotlib**, et d'autres que nous verrons dans la suite de notre projet.

1)Choix du dataset

Notre première étape était de trouver un dataset où étaient recensées des questions et des réponses. Après diverses recherches sur Kaggle, notre choix s'est porté sur des données sur la santé mentale. Cet ensemble de données représente une FAQ sur la santé mentale. Nous avons donc ensuite essayé d'explorer le NLP pour construire un chatbot pouvant répondre à certaines questions. Notre base de données est constituée de 98 questions pour 98 réponses. Les questions traitent différents sujets sur la santé mentale tels que les problèmes d'alcool, de drogue, la dépression, etc...

	Questions	Answers
0	What does it mean to have a mental illness?	Mental illnesses are health conditions that di...
1	Who does mental illness affect?	It is estimated that mental illness affects 1 ...
2	What causes mental illness?	It is estimated that mental illness affects 1 ...
3	What are some of the warning signs of mental i...	Symptoms of mental health disorders vary depen...
4	Can people with mental illness recover?	When healing from mental illness, early identi...

2)Preprocessing

Une fois la base de données trouvée, nous avons effectué des recherches afin de comprendre comment fonctionne le NLP, quelles sont les techniques à utiliser, mais surtout comment bien préparer notre donnée pour pouvoir l'exploiter par la suite. Le preprocessing est une étape cruciale en NLP, car la qualité de notre modèle dépendra de cette partie. En NLP, nettoyer la donnée consiste à enlever les caractères qui ne représentent pas des lettres, tels que "!","#", ou bien "?"(qui sont la plupart du temps présents lorsque l'on pose une question), supprimer les chiffres, ou encore enlever des mots qui sont considérés comme inutiles. Nous avons fait le choix d'enlever les caractères spéciaux car ils nous sont inutiles, mais il est bon de rappeler que dans certains cas d'usage, garder ces caractères peut être intéressants, comme lorsqu'on souhaite analyser si un email est un spam ou non, où "!" peut être un bon indicateur.

Premièrement, à l'aide d'une fonction to_lower(), nous avons modifié chacune des majuscules présentes dans les questions et les réponses. En effet, les mots "Mental" et "mental" ont la même signification, cependant si on n'enlève pas la majuscule, ces deux mots seront représentés comme étant deux mots complètement différents, ce qui pourra induire en erreur notre modèle par la suite.

De plus, dans la langue anglaise, il est fréquent que certains mots soient sous une forme contractée, c'est à dire des mots raccourcis en supprimant des lettres et en les remplaçant par une apostrophe. Par exemple "I'm", "You're", "He's",… Nous avons pour cela implémenté une fonction contractions_w(text) qui à l'aide de la librairie **contractions** permet de regarder chacun des mots et de le transformer sous sa forme initiale.

[It's estimated that mental illness …] ----> [It is estimated that mental illness …]

Ensuite, nous nous sommes occupés de retirer les ponctuations grâce à notre fonction remove_punctuations(text). Nous utilisons la librairie **string** afin d'utiliser **string.punctuation** qui permet de retirer les caractères suivants :

```
: 1 print(string.punctuation)
```

```
!"#$%&'()*+,-./:;<=>?@[\]^_`{|}~
```

"What is ! Mental illness?" ----> "What is Mental illness"

Pour la suite de notre nettoyage, nous faisons également le choix de supprimer les chiffres, car nous considérons qu'ils n'ont pas d'importance pour trouver les mots principaux dans une phrase. Nous utilisons la fonction remove_number(text). Si jamais les chiffres sont des indicateurs importants pour un autre projet, certains outils de Machine Learning permettent de transformer des chiffres en leur valeur textuelle.

Nous continuons ensuite par traiter les stopwords. Mais que sont-ils ? Les stopwords correspondent aux mots qui sont les plus courants dans un texte, et qui n'ont en réalité que très peu d'information valable. Ces mots ne sont pas significatifs. Quelques exemples en anglais sont "a", "the", "is", "are", et la bibliothèque NLTK permet de supprimer environ 180 stopwords, et peut également, si on le souhaite, rajouter un mot considéré comme stopword grâce à sa fonction add.

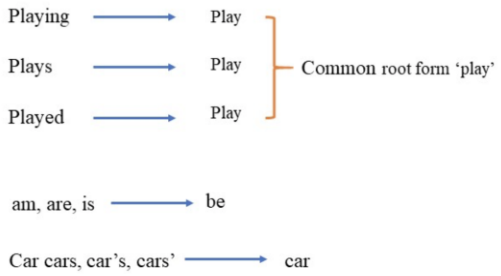
"what is the meaning of mental ilness" ----> "meaning mental illness"

Liste des mots considérés comme des stopwords par la librairie NLTK , et le nombre de mots présents :

```
{'here', 'we', 'isn', 'for', 'were', 're', 'been', 'o', "doesn't", 'y', 'the', 'over', "you're", 'themselves', 'his',
'hasn', 'being', 'not', 'had', 'while', 'don', 'same', 'should', 'before', 'was', 'only', 'wouldn', 'weren', 'myself'
, 'by', 'into', "hasn't", 'yourself', 'there', 'll', "should've", 'her', 'won', "you've", "shan't", 'until', 'yours',
'm', "hadn't", 'him', 'will', 'each', "you'd", 'below', 'itself', 'no', 'hadn', 'haven', 'did', 'now', 'doing', "need
n't", 'and', 'other', 'such', 'ain', 'to', 'herself', 'himself', 'having', 'so', 'you', 'shouldn', "that'll", 'that',
'down', 'against', 'if', 'wasn', 'needn', 'didn', 'mustn', 'through', 'when', 'between', 'these', 'or', 'then', "woul
dn't", 'nor', 'what', 'theirs', 'where', 'how', 'does', "couldn't", 'whom', 'why', "she's", 'during', "shouldn't", 'w
hich', 'as', 'couldn', 'once', 'under', 'are', 'a', 'than', 'because', 'who', "it's", 'more', 'shan', 'doesn', "wasn'
t", 'on', "mustn't", 'she', 'them', 'some', "don't", "you'll", 'ours', 'too', 'd', 'further', 'ma', 'but', 'off', 'i'
, 'do', 'has', 'ourselves', 'any', "aren't", 'from', 'just', 'am', "haven't", 'my', 'most', 's', 'he', 'can', 'above'
, 'their', 'our', 'those', "didn't", 'both', 've', 'hers', 'of', 'with', 't', 'after', 'is', 'an', "isn't", 'have', '
this', 'me', 'up', 'aren', "won't", 'your', 'all', 'they', 'very', 'in', 'yourselves', 'mightn', 'at', 'about', 'own'
, 'be', 'its', 'out', 'few', "mightn't", "weren't", 'it', 'again'}
**********
179
```

Nous avons également remarqué que nos dans réponses, il y'avait beaucoup de lien vers des sites internet, nous nous retrouvions donc avec des mots comme wwwexemplesitecom, et nous avons fait le choix de supprimer chaque mot contenant "www" grâce à la fonction remove_web(text). En effet ces mots viendront rajouter de l'information inutile. Il en est de même avec des mots exprimant la possession en anglais (avec le 's , comme father's), ou l'apostrophe s'est transformé en les caractères â€™, donc nous avons également choisi de supprimer cette transformation de l'apostrophe.

Nous arrivons maintenant à l'étape du "Stemming et Lemmatization", une étape cruciale pour le nettoyage de nos données. Ces deux processus sont des techniques de normalisation de texte. Des algorithmes informatiques ont été mis en place dès 1960. Lorsque nous parlons, ou écrivons, nous utilisons souvent des mots qui sont souvent dérivés les uns des autres, avec un degré qui diffère plus ou moins selon le sens d'un mot, mais dont la racine reste commune.



Stemming :

Cela consiste à réduire un mot à sa forme racine, c'est à dire tronquer le mot de toute déclinaison, ou dérivation, et consiste la plupart du temps à enlever une partie de la fin du terme, quitte à en enlever trop ou pas assez. Nous importons **PorterStemmer** de la librairie NLTK, et appliquons la fonction stem_words(text) sur chacune de nos questions et réponses.

"meaning mental illness" ----> "mean mental ill"

Lemmatization :

Ce processus consiste à ramener un terme, quels que soient ses accords, ou déclinaisons, à sa forme la plus simple. On utiliser **WordNetLemmatizer(),** et appliquons la fonction lemmatize_words(text) sur chacune de nos questions et réponses.

"when was the meeting?" ----> "when be the meet"

Après avoir appliquées toutes ces fonctions, nous pensons que l'étape du préprocessing est finie, nous avons ramené chacune de nos questions et réponses à leur forme la plus simple,et permettant d'être comprises par nos futurs modèles linéaires.

Voici une représentation de notre dataset après nettoyage :

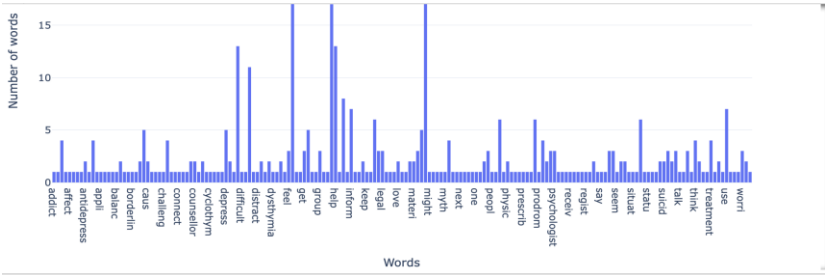
	Questions	Answers
0	mean mental ill	mental ill health condit disrupt personâ€™ tho...
1	mental ill affect	estim mental ill affect adult america adult se...
2	caus mental ill	estim mental ill affect adult america adult se...
3	warn sign mental ill	symptom mental health disord vari depend type ...
4	peopl mental ill recov	heal mental ill earli identif treatment vital ...

3)Visualisation des données

Parmi nos réponses et nos questions, nous avons pensé qu'il serait intéressant de voir quels sont les mots les plus courants. Pour une meilleure compréhension, nous avons utilisé différents moyens de représentations, afin d'avoir une vision globale et parfois plus représentatives dans certains cas.

Visualisation des questions :

- **Par un histogramme :** C'est en effet par un histogramme que nous avons choisi de représenter la récurrence des mots dans notre dataset. Nous avons d'abord transformé notre colonne Questions en une liste de questions, puis avons split chaque question en une liste de mot dans une nouvelle liste, et enfin nous avons regroupé tous les mots présents en une seule et même liste. Grâce à la bibliothèque **plotly.express**, nous avons pu créer l'histogramme suivant :



On peut observer que les mots sont triés par ordre alphabétique, mais du fait du grand nombre de mot, tous n'apparaissent pas. Cet histogramme n'est pas assez représentatif de notre dataset.Ce que l'on souhaite observer, ce sont principalement les mots qui apparaissent plus d'une fois.

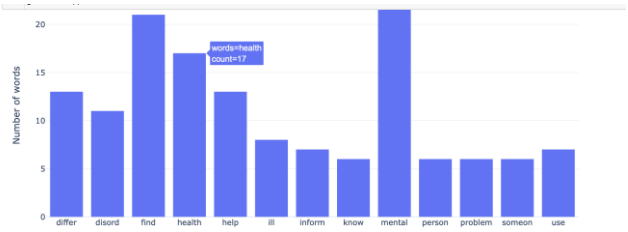
C'est pour cela que nous avons créé un nouveau DataFrame, sub_df, qui est une copie de notre DataFrame initial, cependant, ici les mots sont regroupé par leur nombre d'apparition.

	words	Counts
0	mental	218
1	ill	97
2	health	263
3	condit	22
5	person	77
...	...	...
13798	whether	8
13799	use	140
13800	substanc	57
13801	visit	36
13802	wwwcisurca	6

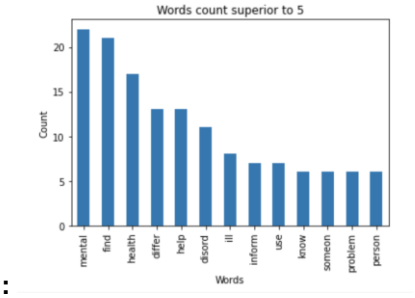
[11087 rows x 2 columns]

Nous avons ensuite fait le choix de n'afficher dans un nouvel histogramme que les mots dont l'apparition est supérieure à 5. Nous avons affiché les résultats sous la forme :

D'un histogramme :

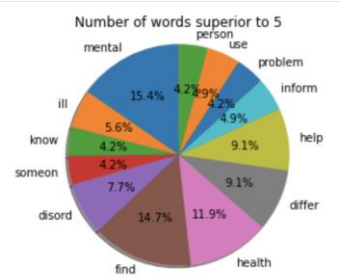


En utilisant **matplotlib.pyplot** où le paramètre kind = 'Bar' **:**

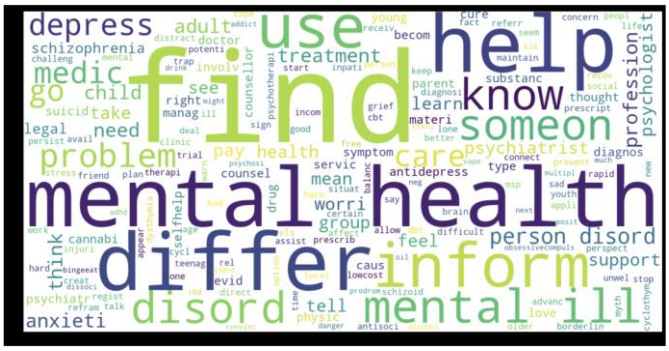


:

- **Par plt.pie :** Nous avons trouvé intéressant, et amusant, d'afficher la récurrence d'un mot présent dans nos questions à travers cette forme, car cela nous indique le pourcentage, et donc la proportion, d'un mot par rapport aux autres.



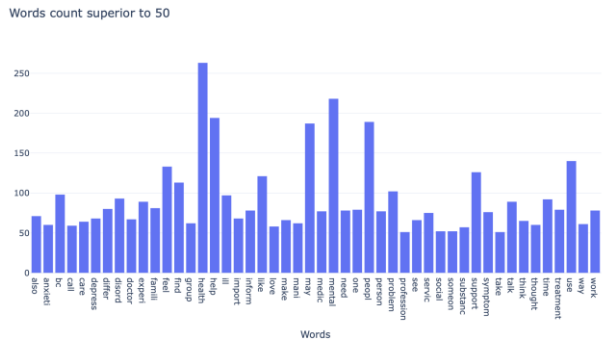
- **En utilisant WordCloud:** Cette manière d'afficher de la donnée est nouvelle pour nous. Nous avons trouvé ce moyen réellement intéressant pour le traitement de texte, car il nous affiche les mots les plus fréquents avec des lettres plus grosses et des couleurs différentes. Plus la taille du mot est petite, moins il est important. Grâce à cela, nous pouvons voir d'un coup d'œil que les mots "find","help", "mental",… sont les mots les plus importants, ce qui parait cohérents du fait que nous sommes en train d'analyser des questions sur la santé mentale.



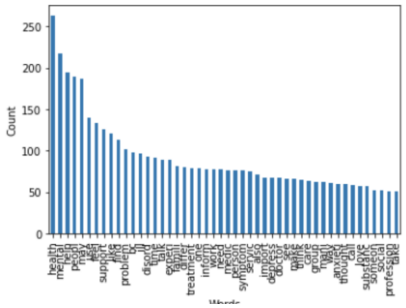
Visualisation des réponses :

Pour ce qui est des réponses de notre dataset, nous avons appliqué le même processus, et les mêmes affichages de données que pour les questions, à la différence qu'ici nous avons fait le choix de n'afficher que les mots dont la fréquence est supérieure à 50. En effet, les réponses sont logiquement plus longues que les questions, et donc composées de beaucoup plus de mots. Il n'aurait pas été pertinent de garder seulement les mots qui n'apparaissent plus que 5 fois, comme c'était le cas pour les questions.

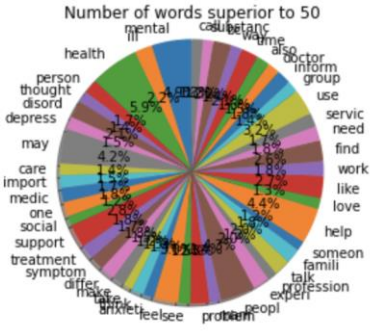
- **Par un histogramme :**



Nous voyons facilement quels sont les mots les plus fréquents, cependant, même avec une fréquence de 50, il reste beaucoup de mots, et donc l'information que l'on souhaite visualiser reste difficile à percevoir.

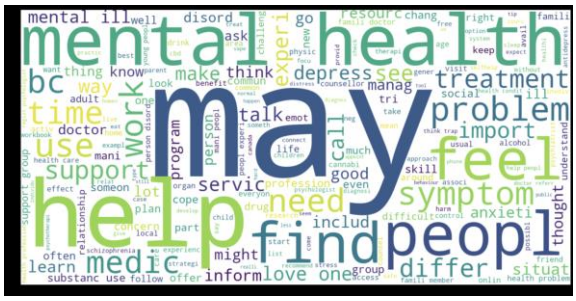


- **Par plt.pie:** Même remarque que précédemment, la tarte ici n'est pas aussi représentative que pour les questions, car il y a énormément de mots.

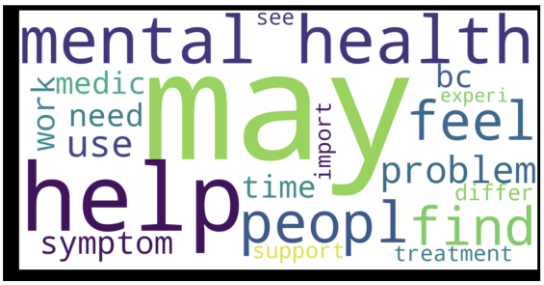


Une solution pourrait être de chercher les mots dont la fréquence d'apparition est supérieure à 50.

- **En utilisant WordCloud :** Mais c'est ici que nous avons trouvé réellement intéressant wordcloud. En effet, bien que le nombre de mots soit le même que pour les affichages précédents, ici nous remarquons rapidement quels sont les mots les plus fréquents, grâce à leur taille et les différences de couleurs.



Afin d'avoir une meilleure représentation, nous avons utilisé le paramètre max_words = 20 afin de n'afficher que 20 mots sur notre image.



4)Modèles linéaires

Arrivés à cette partie, il nous a fallu une grande réflexion sur comment aborder le problème.

La première étape était de comprendre qu'un modèle linéaire ne peut pas interpréter des mots. Il faut donc transformer chaque mot en une valeur capable d'être comprise par l'ordinateur. Il y a différents moyens de faire cela, mais nous nous sommes orientés vers la méthode TF-IDF (Term Frequency – Inverse Document Frequency) . C'est une mesure statistique qui permet, à partir d'un ensemble de textes, de connaitre l'importance relative de chaque mot. Le TF-IDF d'un mot est obtenu en multipliant deux métriques :

- Term Frequency (TF) : Il y a différentes manières de calculer cela, mais la façon la plus simple est de calculer le nombre de fois qu'un mot apparait dans un document divisé par le nombre total de mots dans le document. Chaque document à sa propre fréquence.

$$tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{i,j}}$$

- Inverse Data Frequency (IDF) : Il s'agit du logarithme du nombre de documents divisé par le nombre de document contenant le mot. La fréquence inverse des données détermine le poids des mots rares dans tous les documents du corpus.

$$idf(w) = log(\frac{N}{df_t})$$

Le résultat final est obtenu en multipliant les deux métriques :

$$w_{i,j} = tf_{i,j} \times \log\left(\frac{N}{df_i}\right)$$

La bibliothèque **scikit-learn** offre une fonction **TfidfVectorizer()** qui à travers différents paramètre permet convertir une collection de documents en une matrice de TF-IDF selon les mots présents. Nous avons appliqué ce processus pour convertir nos questions, qui ont été préalablement nettoyées comme vu précédemment. En ce qui concerne les réponses, nous sommes partis dans une optique de labeliser notre dataset, en utilisant un **LabelEncoder().** Comme il y a 98 données dans notre dataset, nous avons donc labélisé les questions en attribuant une valeur entre 0 et 97 à chacune des réponses dans une nouvelle colonne “Answers_Encoded”.

	Questions	Answers	Answers_Encoded
0	mean mental ill	mental ill health condit disrupt person though...	51
1	mental ill affect	estim mental ill affect adult america adult se...	23
2	caus mental ill	estim mental ill affect adult america adult se...	24
3	warn sign mental ill	symptom mental health disord vari depend type ...	89
4	peopl mental ill recov	heal mental ill earli identif treatment vital ...	33

Nous avons donc nos données d’entrainement (df[“Questions”]) ainsi que nos target (df[“Answers_Encoded”]). Nous avons choisi d’utiliser deux algorithmes, **LinearSVC** et **MultinomialNB**. L’idée est de transformer chaque nouvelle question en tfidf, et d’utiliser un des modèles précédents entraînés afin de trouver une réponse valide.

Voici un exemple avec les questions "Can I have danger with canabis?" Et “Is cannabis dangerous?”, on obtient respectivement les réponses suivantes :

- LinearSVC :

```
Answer:  ['Cannabis smoke, for example, contains cancer-causing toxins. However, the risk of developing some cancers
(e.g., mouth, tongue and lung) is less for cannabis smokers than tobacco smokers, partly because they tend to smoke l
ess than tobacco users. And, while all drugs have an effect on the brain, the particular properties of the drug influ
ence the level of risk of harmful consequences. The negative effects of cannabis on the brain, for example, seem to b
e less than the effects of some substances such as alcohol. \n Legalizing cannabis provides an opportunity to put in
place regulations to minimize potential harms. The danger of buying and using any illegal drug is that we can never k
now for sure what exactly is in it. Cannabis is legal in Canada as of October 17, 2018. Adults (over age 19 in BC) ar
e now permitted to possess up to 30 grams of cannabis in public. Cannabis is regulated by the Province of British Col
umbia and will be sold through the Liquor Distribution Branch. Cannabis will be tested for quality. \n When drugs are
produced and obtained inside a regulated system, it is possible for us to know about the contents and dosage of what
we are taking. This helps us manage the risks. However, it is likely that cannabis will still be available outside th
e government system. It is important to know that the quality of cannabis obtained from a dealer or a friend is unkno
wn and may contain contaminants like mold, mildew, or fillers that may be toxic. \n The legalization of cannabis also
provides us with openings to engage in honest and thoughtful discussions about drug use with our families and communi
ties. When dealing with complex issues, like cannabis policy, no one has all the answers. But as community members, w
e all have thoughts, feelings and experiences around drugs and drug use to share with each other. Engaging together t
o explore and share ideas will help us discover how to manage use, as individuals and communities, in ways that maxim
ize benefit and minimize harm. \n The Canadian Institute for Substance Use Research, formerly CARBC, is a member of t
he BC Partners for Mental Health and Addictions Information. The institute is dedicated to the study of substance use
```

```
Answer:  ['Different kinds of therapy are more effective based on the nature of the mental health condition and/or sy
mptoms and the person who has them (for example, children will benefit from a therapist who specializes in children’s
mental health). However, there are several different types of treatment and therapy that can help.']
```

- MultinomialNB :

```
Answer:  ['Cannabis smoke, for example, contains cancer-causing toxins. However, the risk of developing some cancers
(e.g., mouth, tongue and lung) is less for cannabis smokers than tobacco smokers, partly because they tend to smoke l
ess than tobacco users. And, while all drugs have an effect on the brain, the particular properties of the drug influ
ence the level of risk of harmful consequences. The negative effects of cannabis on the brain, for example, seem to b
e less than the effects of some substances such as alcohol. \n Legalizing cannabis provides an opportunity to put in
place regulations to minimize potential harms. The danger of buying and using any illegal drug is that we can never k
now for sure what exactly is in it. Cannabis is legal in Canada as of October 17, 2018. Adults (over age 19 in BC) ar
e now permitted to possess up to 30 grams of cannabis in public. Cannabis is regulated by the Province of British Col
umbia and will be sold through the Liquor Distribution Branch. Cannabis will be tested for quality. \n When drugs are
produced and obtained inside a regulated system, it is possible for us to know about the contents and dosage of what
we are taking. This helps us manage the risks. However, it is likely that cannabis will still be available outside th
e government system. It is important to know that the quality of cannabis obtained from a dealer or a friend is unkno
wn and may contain contaminants like mold, mildew, or fillers that may be toxic. \n The legalization of cannabis also
provides us with openings to engage in honest and thoughtful discussions about drug use with our families and communi
ties. When dealing with complex issues, like cannabis policy, no one has all the answers. But as community members, w
e all have thoughts, feelings and experiences around drugs and drug use to share with each other. Engaging together t
o explore and share ideas will help us discover how to manage use, as individuals and communities, in ways that maxim
ize benefit and minimize harm. \n The Canadian Institute for Substance Use Research, formerly CARBC, is a member of t
he BC Partners for Mental Health and Addictions Information. The institute is dedicated to the study of substance use
```

```
["Although this website cannot substitute for professional advice, we encourage those with symptoms to talk to their
friends and family members and seek the counsel of a mental health professional. The sooner the mental health conditi
on is identified and treated, the sooner they can get on the path to recovery.\nIf you know someone who is having pro
blems, don't assume that the issue will resolve itself. Let them know that you care about them, and that there are tr
eatment options available that will help them heal. Speak with a mental health professional or counselor if you think
your friend or family member is experiencing the symptoms of a mental health condition. If the affected loved one kno
ws that you support them, they will be more likely to seek out help."]
```

Bien que nous obtenions des réponses correctes pour certaines questions, nous sommes arrivés à un problème fondamental. On remarque certaines fois, qu'avec nos deux modèles, pour une même question posée, les réponses diffèrent. En effet, avec la stratégie adoptée de labéliser notre dataset, il n'est pas possible de séparer nos données en données d'entrainement et de test. Cela n'a pas de sens, car chaque question a sa propre réponse, pour le moment nous ne pouvons pas sortir une nouvelle réponse pour une nouvelle question, nous pouvons seulement trouver la réponse la plus proche. De ce fait, nous ne pouvions pas calculer la performance, ni évaluer nos modèles. Ce manque d'évaluation ne nous donnait pas la possibilité de remettre en cause les modèles utilisés.

L'idée pour résoudre notre problème serait de résoudre le problème suivant :

$$\min \Sigma||A\Phi(q)-\Phi(a)||^2 \qquad \text{où } \Phi : q \rightarrow a$$

$$A \qquad\qquad \text{Questions} \rightarrow \text{Réponses}$$

Malheureusement, par manque de temps, nous avons trouvé judicieux qu'il serait sans doute mieux pour nous de nous ramener à un problème que nous avons déjà étudié, se ramener à de l'apprentissage supervisé avec une base de données contenant un label.

II)Chatbot Intent Recognition

1)Choix du dataset

Afin de simplifier le problème précédent, nous avons jugé judicieux de choisir une nouvelle base de données. Mais pas n'importe laquelle, car en effet, cette fois ci nous avons des questions et réponses comme précédemment, à la différence que cette fois ci elles sont regroupées par "intention". Notre document est initialement sous la forme d'un document **json.** Pour une meilleure visualisation, nous l'avons ramené sous forme d'un DataFrame, avec comme colonnes "Text" correspondant à des réponses de l'utilisateur, et "Intent", l'intention d'une question, qui servira ici target.

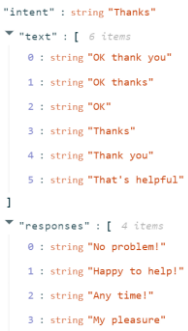
2) Description du dataset

Ainsi, à l'aide de cette base de données, il était possible de labéliser notre base de données, avoir une target qui est l'intention et ainsi faire de la classification. Il y a 22 intentions différentes.

Les numéros à droite correspondent au nombre de phrases existantes associées à l'intention correspondantes. Par exemple la catégorie 'NotTalking2U' possède les phrases suivantes :

	Text	Intent
63	I am not talking to you	NotTalking2U
64	I was not talking to you	NotTalking2U
65	Not talking to you	NotTalking2U
66	Wasn't for you	NotTalking2U
67	Wasn't meant for you	NotTalking2U
68	Wasn't communicating to you	NotTalking2U
69	Wasn't speaking to you	NotTalking2U

Ainsi, dans notre chatbot, le but sera d'appliquer le même préprocessing que nous allons appliquer à la base de données, puis d'appliquer un modèle de Machine Learning qui analysera l'intention. Puis dans notre base de données, il existe des réponses à ces intentions, nous répondrons donc par l'une d'entre elles de manière aléatoire. Par exemple, ci-dessous nous avons l'intention 'Thanks' le 'text' correspond aux phrases de l'utilisateur et nous prendrons une réponse au hasard dans 'responses'.



3)Modèles

Avant d'appliquer les modèles que nous avons choisi, il faut refaire le même procédé de nettoyage de nos données. Cependant, nous allons utiliser **Tensorflow** et **Keras,** qui sont des bibliothèques permettant un apprentissage automatique plus rapide et plus efficace. En effet, nous important de ces bibliothèques la classe **Tokenizer**, qui permet de nettoyer chaque corpus de texte, mais également de vectoriser ces corpus en transformant chaque texte en une séquence d'entiers, ou en vecteur où le coefficient de chaque "token" peut être basé sur TF-IDF(mais pas seulement, cela aurait pu être binaire, basé sur le nombre de mots, etc.)

```
tf.keras.preprocessing.text.Tokenizer(
    num_words=None,
    filters='!"#$%&()*+,-./:;<=>?@[\\]^_`{|}~\t\n',
    lower=True,
    split=' ',
    char_level=False,
    oov_token=None,
    document_count=0,
    **kwargs
)
```

Une fois le pre processing effectué, nous allons utiliser plusieurs méthodes que propose cette classe afin de transformer nos données afin qu'elles puissent être comprises par la suite par nos réseaux de neurones.

Nous appelons d'abord fit_on_texts(texts), qui met à jour notre vocabulaire interne.

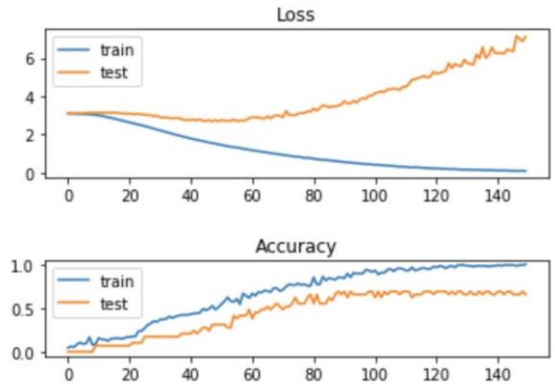
Cette méthode est requise avant d'utiliser text_to_sequences(texts), qui transforme chacun de nos "Text" en séquence de nombres. Par exemple df["Text"][1] == "Hi there" et si on pose : sequences = tokenizer.texts_to_sequences(df["Text"]), alors avec sequences[1], on obtient [51,52]

En regardant dans l'image juste au-dessus, on remarque que 51 : 'hi' et 52 :'there'.

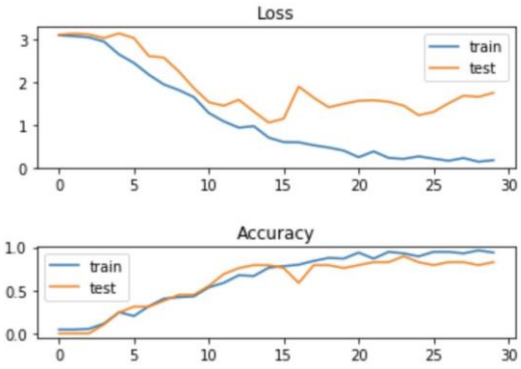
Enfin nous utilisons pad_sequences(sequences) qui transforme des séquences de nombres en un tableau Numpy2D.

Nous avons testé deux réseaux de neurones : pour cela nous avons utilisé le modèle Sequential qui consiste à un perceptron à plusieurs couches. Ce sont deux récurrents neural networks.

Le premier réseau de neurones contient 4 layers fully connected. La première et deuxième couche contient 128 neurones, la troisième 64 et la dernière contient 22 neurones correspondant au nombre d'intentions différentes. Pour la dernière fonction d'activation nous avons utilisé softmax. Nous avons effectué des tests afin de déterminer le nombre d'epochs optimal et nous avons finalement choisi 150 epochs. Pour cela, nous avons affiché le graphique de la loss et de l'accuracy durant le training. Puis nous devions trouver un compromis entre la loss et l'accuracy. Nous avons donc pris le coude la fonction. Avec ce réseau nous obtenons une accuracy de 65%.



Notre second réseau de neurones consiste en 3 hidden layers. Comme le premier réseau, la dernière couche contient 22 neurones correspondant aux nombres d'intentions. Nous avons également ajouté un Dropout de 0.5 qui permet d'éviter l'overfitting. Elle permet de remettre de manière aléatoire des unités à 0 à une fréquence de 0.5. Cette part d'aléatoire fait varier notre modèle entre 72% et 82% d'accuracy.



Nous sommes parvenus à trouver un modèle qui reste plutôt efficace pour du NLP. Nous avons mis en production notre modèle en faisant un chatbot. Ainsi à chaque input de l'utilisateur, nous appliquons le même pré-traitement que mentionné plus tôt et utilisons le modèle qui a donné la meilleure accuracy. Nous avons enregistré le modèle donnant le meilleur résultat grâce à la libraire **pickle,** qui permet de stocker et de restaurer un objet Python tel quel, sans aucune manipulation supplémentaire.

<u>Voici un exemple de conversation avec Genisys, notre chatbot :</u>

```
Start talking to the chatbot, type 'exit' to end conversation:
User: Hi!
ChatBot: Hi human, please tell me your GeniSys user
User: How are you?
ChatBot: Hi, I am great, how are you? Please tell me your GeniSys user
User: I'm fine
ChatBot: See you later
User: Wait, can you tell me your name?
ChatBot: You may call me Geni
User: And what time is it?
ChatBot: One second
User: Ok ok
ChatBot: Any time!
User: Before I left, tell me a joke
ChatBot: A woman goes into a clothes shop, 'Can I try that dress on in the window please?' she asks. 'I'm sorry madam
', replies the shop assistant, 'but you'll have to use the changing-rooms like everyone else.'
User: Ok tanks, bye
ChatBot: No problem, goodbye
User:
```

III)Conclusion

Ce projet nous a permis de découvrir une nouvelle discipline du Machine Learning, le Natural Langage Processing. Bien qu'il nous reste de nombreuses choses à apprendre sur le sujet, nous avons néanmoins étudié les différentes étapes nécessaires pour nettoyer et transformer nos données afin qu'elles puissent être utilisées. Nous avons également, à travers de nombreuses recherche, eu une première expérience avec les réseaux de neurones, que l'on a pu mettre en œuvre grâce aux librairies Keras et Tensorflow, et comment modifier chaque paramètre afin d'obtenir la meilleure précision. Nous avons eu beaucoup de plaisir à réaliser ce projet, et pourquoi pas tenter de l'améliorer plus tard pour obtenir un chatbot plus performant.

En effet, un moyen de développer le projet serait de fusionner avec une autre base de données. En effet, la nôtre reste petite et ainsi limitée. De plus, il serait intéressant d'explorer d'autres réseaux de neurones utilisés pour le NLP. Nous avons fait des recherches d'articles scientifiques mais avec plus de temps, il serait possible de l'améliorer davantage.

IV)Annexes

Liens de nos dataset :

https://www.kaggle.com/datasets/elvinagammed/chatbots-intent-recognition-dataset

https://www.kaggle.com/datasets/narendrageek/mental-health-faq-for-chatbot?datasetId=903283

Liens et articles :

https://machinelearningmastery.com/how-to-choose-loss-functions-when-training-deep-learning-neural-networks/

https://towardsdatascience.com/how-to-create-a-chatbot-with-python-deep-learning-in-less-than-an-hour-56a063bdfc44

https://shanebarker.com/blog/deep-learning-chatbot/

https://www.datacamp.com/community/tutorials/stemming-lemmatization-python

https://www.analyticsvidhya.com/blog/2021/06/text-preprocessing-in-nlp-with-python-codes/

https://www.tensorflow.org/api_docs/python/tf/keras/preprocessing/text/Tokenizer