**MASTER THESIS**

Graduate School of Project Design, Miyagi University
Department of Information Systems


Thierry Roger BAYALA

# Early Discovery Of Epidemic By Big Data Analysis

Supervisor: Prof. Atsushi Togashi

March 2019

Firstly, I would like to express my sincere gratitude to JICA (Japan International Cooperation Agency) for the continuous financial support of my master thesis..

Besides JICA, I would like to thank JICE and Miyagi University staffs for their assistance during my thesis.

My sincere thanks also goes to Prof. Togashi Atsushi , Prof.Suguri Hiroki. , and Prof. Koji Makanae, who provided me an opportunity to join their laboratory, and who gave access to the laboratory and research facilities. Without they precious support it would not be possible to conduct this research.

I thank my fellow labmates Sow Aboubacar and Tumukunde Ibrahim for the stimulating discussions during our weekly join seminar. Also, I thank Prof. Malo Sadouanouan for helping to define my research topic.

Last but not the least, I would like to thank my family: my parents and to my brothers and sister for supporting me spiritually throughout writing this thesis and my life in general.

I declare that I carried out this master thesis independently, and only with the cited sources, literature and other professional sources.

In Japan date 2019/03/17

Title: Early Discovery Of Epidemic By Big Data Analysis

Author: Thierry Roger BAYALA

Department: Information Systems

Supervisor: Prof. Atsushi Togashi

**Abstract:**

Epidemic surveillance requires a rapid collection and integration of data and events related to the disease. Adequate measures, including education and awareness, must be rapidly taken to reduce the disastrous consequences of the disease. However, developing countries, especially those in West Africa, face a lack of real-time data collection and analysis system. This situation delays the analysis of risk and decision making. This research aims to contribute to the surveillance of the meningitis epidemic based on Twitter datasets. We divided our approach into two parts. The first part consisted of investigating different methods to convert the tweet data into numerical data that will then be used in Machine Learning Algorithms for the classification tasks. The second step is to evaluate these approaches using different algorithms and to compare their performance in term of training time, Precision, F1-Score, and Recall. As a result, we found that the SVM Machine Algorithm performed a g result with 0.98 of accuracy using the TF-IDF embedding approach while the ANN algorithm performed an excellent good accuracy of 0.95 using the Skip-gram embedding model.

**Keywords:** Natural Language Processing, Twitter, Machine Learning, Meningitis, Neural Network, Word Embeddings, Support Vector Machine, Logistic Regression, Random Forest.

# Contents

# List of Abbreviations

| **ABBREVIATIONS** | **DESCRIPTION** |
|---|---|
| ML | Machine Learning |
| WHO | World Health Organization |
| NN | Neural Network |
| NLP | Natural Language Processing |
| API | Application Programming Interface |
| SML | Supervised Machine Learning |
| DC | Document Classification |
| SVM | Support Vector Machine |
| FBE | Frequency Based Embedding |
| PBE | Prediction Based Embedding |
| TF-IDF | Terms Frequency - Inverse Document Frequency |
| CBOW | Continues Bag Of Word |

Table 1: Abbreviation Table

# List of Figures

# List of Tables

# Introduction

Twitter is a microblogging platform where many interconnected users frequently share information among themselves. Through this platform, users interact through short messages by posting their opinion about a current situation such as the presidential election, the stock market, and healthcare issues that occur in their surroundings. This diversity of topics tackled every day has made Twitter an exciting source of data that was used by researchers to predict events. In recent years, Twitter data has allowed conducting a lot of researches in the fields of epidemic disease prevention such as the epidemic of influenza successfully. [1][2][3][4] In this paper, we tackled the problem of meningitis which is an infectious disease affecting many people in West African countries every year. We decided to work on this disease for the following reasons:

- Meningitis affects mostly young people. In 2017, a report relating elaborated by the United Nations Department of Economic and Social Affairs estimated the percentage of African population who suffer from meningitis between 0 to 14 years old is about 41%. [5]
- There is almost no system for data collection and analysis to prevent the outbreak of meningitis epidemic in real-time.
- The ability of the disease to create severe damages such as hearing loss and sometimes mental disorders in a short period after the exhibition requires to be diagnosed earlier to reduce the disastrous damages.[6]
- There is a need to discover if the country faces a situation of the epidemic to limit the spreading over the country through sensitization of the population.

To reach our goal, we trained a classifier that can be able to identify the label of a given tweet mentioning the keyword "meningitis." Since we approached this problem as a Supervised Machine Learning problem, we proceed to the annotation of our dataset based on the features we defined as meningitis dictionary. Most of the existing machine learning algorithm deals with numbers. Therefore, we trained a word embedding model that consists of converting our tweets into a numerical vector. We used the embedding model to train several classifiers with the target labels "Infection, Concern, Vaccine, Campaign, and News." We investigated the Support Vector Machine (SVM), Artificial Neural Network (ANN), Logistic Regression (LR), Random Forest (RF) models and compared their results.

Our approach involves several concepts such as Meningitis, Machine Learning, Supervised Machine Learning, Document Embedding and Document that need to be clarified before we dive in the methodology. In the first chapter, we presented

each concept with their definition associated. The second chapter of this document contains the methodology we adopted in this thesis. In the third and the fourth chapter, we presented the implementation process and the different results respectively.

# 1. Concepts and Definition

## 1.1 Meningitis

Meningitis is an inflammation of the membranes covering the brain called meninges. According to the World Health Organization (WHO), 400 million people living in Africa are exposed to the high risk of epidemics including meningitis. [17] Significant outbreaks of meningitis occur and mainly affect children and young adults. In general, most of the affected people die within 24 to 48 hours after the onset of the disease symptoms. Among those survivors between 10% and 20% suffer from mental retardation, hearing loss, or learning disabilities. [16]

## 1.2 Machine Learning

Machine Learning refers to creating models that can be able to learn from data called "inputs" and produce a result called "output" for a given new data. These models are a specification of a mathematical (or probabilistic) relationship between the input variables and the output variables. [16]

Based on the approach used, a machine learning model can be supervised or unsupervised. A model is supervised if the dataset analyzed contain a specifics correct answers called *"Target"* or *"Label"*. In unsupervised machine learning, there are no specific labels in the training dataset. These two machine learning approaches can use several algorithms to perform classification tasks. We presented in the following sections, four of them that we used to achieve our classification task.

## 1.3 Classification Models

### 1.3.1 Logistic Regression

Logistic Regression is one of the popular algorithm used for classification purpose in the field of machine learning. The LR method is mostly used to solve binary classification problems but can also be used for multiple classification problems through an approach called *"One Vs All"*. The main idea behind the concept of *"One Vs All"* is to divide the training dataset into sub training datasets, and for each sub training dataset, only one class is consider as *"positive value"*, and the remaining classes are considered as *"negative value"*. So, the multiple classification problems are transformed into a binary classification problem.[21]

The Logistic Regression model is based on probabilistic distribution using a sigmoid function $\sigma$ defined by :

$$\sigma(X) = \frac{1}{1 + e^{-}X}$$

.
The variable $X$ is a linear operation defined by $\theta^T x$ with $x$ the input vector and $\theta^T$ the transpose of the weight of the trained model. The process of classification using the LR can be described as follow:

| **Algorithm** Tweets Classification using the LR model |
|---|
| **Input:** dataset D, learning rate, a trained logistic regression model.<br>**Output:** the class of the tweet.<br><br>**Step1:** receive the input $X$.<br>**Step2:** multiply each input sent to the model by weights.<br>**Step3:** sum all the weighted input.<br>**Step4:** generate output: the output of network is produced by passing that sum through the activation function $\sigma$<br>**Step5:** if the value of $\sigma(X)$ is $\geq 0.5$ set the value of the output at positive else set the value of the output as negative. |

### 1.3.2 Artificial Neural Network

Artificial neural network (ANN) is a machine learning approach that learn from data in the form of connected network unit so called *"neurone"* or *"node"*. Figure 1.2 is a simple illustration of an ANN structure. The connection between the nodes called *"Weights "*are crucial when training an ANN model. The learning process consists of adjusting these weights which can be done using the back-propagation method. [20]

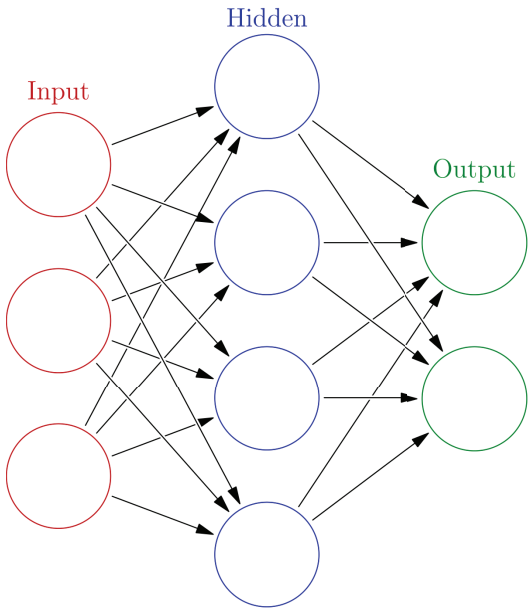



Figure 1.1: Simple ANN achitecture

The classification algorithm using the ANN model is described as following:

| **Algorithm** Tweets Classification using the ANN model |
| --- |
| **Input:** dataset D, a trained neural network.<br>**Output:** the probabilities, the tweet class.<br><br>**Step1:** receive the input $X$.<br>**Step2:** propagate each input in the network and multiply them by the trained model weight.<br>**Step3:** sum all the weighted input.<br>**Step4:** generate output: apply a softmax function to the sum.<br>**Step5:** set the class of tweet by selecting the class name with the highest probability. |

### 1.3.3 Random Forest

The concept of the Random Forest (RF) algorithm is based on the decision tree. Trees are entities made of two parts including the root which are the features and leaves that represent the class of our tweets. The name Rand Forest comes from

the fact that the model contains most of the time many trees which are chosen randomly. Figure 1.3 is an illustration of an RF architecture.

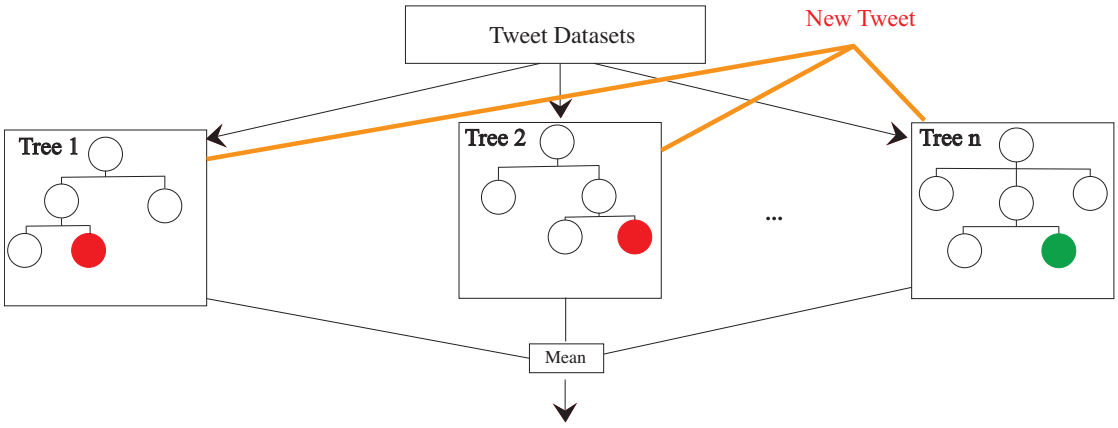



Figure 1.2: Random Forest architecture

| **Algorithm** Tweet Classification using a Random Forest model |
|---|
| **Input:** dataset D, trained model<br>**Output:** vote count, the class of the tweet.<br><br>**Step1:** receive the tweet as input.<br>**Step2:** send the input to all tree in the forest for voting.<br>**Step2:** count the polling result for each class of tweet.<br>**Step4:** generate output: the output is the class with the maximum of vote |

### 1.3.4 Support Vector Machine

Support Vector Machine (SVM) model is a classifier formally determined via tree hyperplanes: The decision boundary defined by the equation $W^T X = 0$, the negative hyperplane defined by the equation $W^T X = -1$, and the positive hyperplane defined by the equation $W^T X = 1$. In each equation, the variable $x$ represents the input vector and the variable $W^T$ is the transpose of the weight of the trained model. Training SVM model on training dataset consists of maximizing the gap between the positive hyperplane and the negative hyperplane, so then the model can categorize new examples of the tweets. Figure 1.3 is an illustration a SVM model.

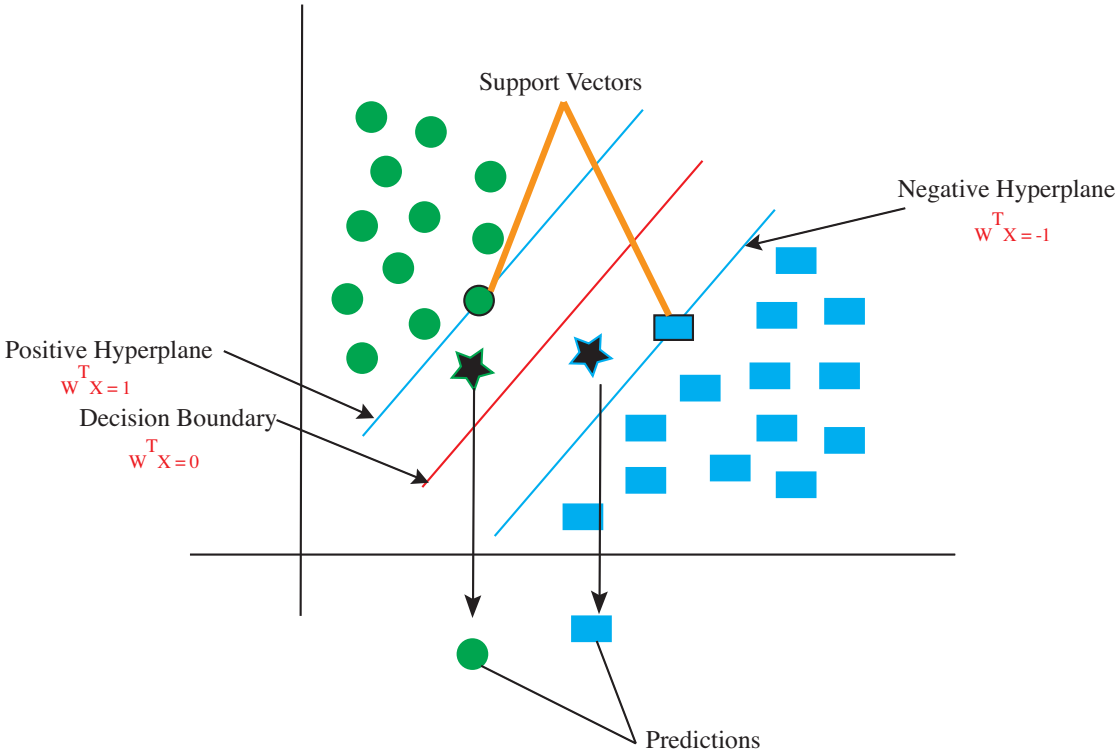



Figure 1.3: Random Forest architecture

| **Algorithm** Tweet Classification using SVM models |
| --- |
| **Input:** dataset D, trained SVM model.<br>**Output:** the class of tweet.<br><br>**Step1:** receive the tweet as input $X$.<br>**Step2:** multiply each input sent to the model by the weights of the trained models. **Step3:** compare the value obtained the decision boundary.<br>**Step4:** generate output: if the value is bigger than 1 , then set the class of the tweet to the positive class else if the output value is smaller than $-1$ set the class of the tweet to the negative class. |

## 1.4 Document Embedding

The different models we present in the previous section operate with numerical therefore text data must be converted into numerical data. The process of converting text data into numerical data is so-called "document embedding" or "word embedding" embedding. One of the earlier works done by G. Salton et al. in 1975. The main idea to represent a document as a vector in a vector space for the purpose of information retrieval. [11] The result of this has allows the progress in information retrieval which is used in many research engines. Recently this concept has been taken up by other researchers to improve it for NLP tasks. [12,13].

There exist several approaches of document embedding, and they can be divided into two groups: The embedding based on count (EBC) and the embedding based on prediction (EBP).

### 1.4.1 Embedding Based on Counting

The process of transforming text into numerical data using the EBC's group is based on counting words in the document. Some examples of the algorithm are TF-IDF, Count Vector, Co-Occurrence Matrix. To illustrate how these algorithms work, let consider these two examples of tweets.

**Doc 1**: my son has meningitis
**Doc 2**: i got meningitis

| | my | son | has | meningitis | i | got |
|---|---|---|---|---|---|---|
| Doc 1 | 1 | 1 | 1 | 2 | 0 | 0 |
| Doc 0 | 0 | 0 | 0 | 2 | 1 | 1 |

Table 1.1: TF-IDF

| | my | son | has | meningitis | i | got |
|---|---|---|---|---|---|---|
| Doc 1 | 1 | 1 | 1 | 1 | 0 | 0 |
| Doc 0 | 0 | 0 | 0 | 1 | 1 | 1 |

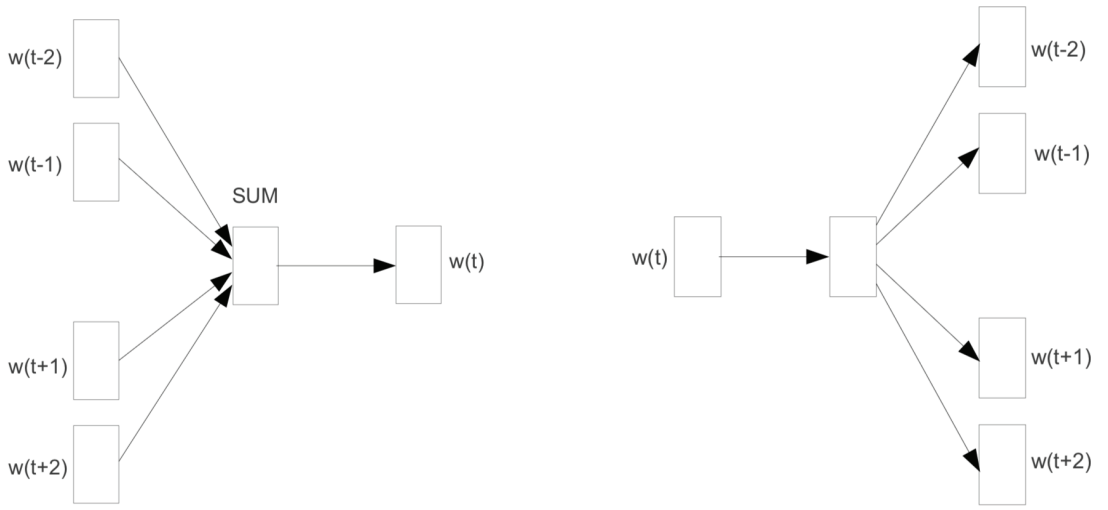
Table 1.2: Count Vector

| | my | son | has | meningitis | i | got |
|---|---|---|---|---|---|---|
| my | 0 | 1 | 0 | 0 | 0 | 0 |
| son | 1 | 0 | 1 | 0 | 0 | 0 |
| has | 0 | 1 | 0 | 1 | 0 | 0 |
| meningitis | 0 | 0 | 1 | 0 | 0 | 1 |
| i | 0 | 0 | 0 | 0 | 0 | 1 |
| got | 0 | 0 | 0 | 0 | 1 | 0 |

Table 1.3: Co-Occurrence Matrix

### 1.4.2 Embedding Based on Prediction

Different from EBC group, the EBP group is composed of two algorithms: The CBOW or Continuous Bag of Word and the Skip-gram algorithms. Both of these algorithms use an artificial neural network to generate the vector representation of a word or document. However, the approaches used by the algorithms are different. While the main concept of Ski-gram model is to predict the surrounding word called context given a corpus, the CBOW' s main idea is to predict a word given a context. These algorithms were introduced by Mikolov et al. in their paper "Distributed Representations of Words and Phrases and their Compositionality". [14] Figure 1.4 is an illustration of the skip-gram and the CBOW models.




(a) CBOW word representation (b) Skip-gram word representation

Figure 1.4: Skip-gram Vs CBOW architecture

We used these two algorithm to train train our word embedding models through a Word2Vec network [11]. We considered each tweet as a single document and the labels was defined as presented in the following examples:

[my prof has meningitis and no more classes rest of the term]
**tweet_0**

[was worried i had meningitis because i ached so much and could not move my head too]
**tweet_1**
...

The process of vectorizing the tweet consists of calculating the probability of having a second word given the first through a neural network as presented in Figure 1.5. Let us assume that we intend to vectorize the tweet: [my prof has meningitis and no more classes rest of the term].

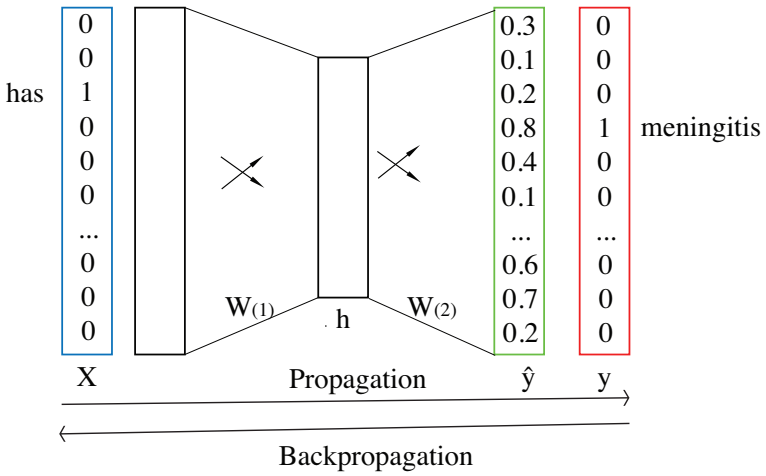



Figure 1.5: Word2Vec Network

The first step in the Word2Vec approach is to split the whole sentence into different tokens and from this list of tokens, create a pair of the tokens from the left to right as follow [(my, prof), (prof, has), (has, meningitis) ...] for the all entire corpus. The second step consists of training a neural network where each node of the neural corresponds to a single token. The neural network has an input layer, a hidden layer and an output layer. The input and the output layers have the same number of nodes representing the size of the vocabulary. For each pair of tokens, the purpose is to find the probability to have the second word given the

first. Let consider the pairs of words (has, meningitis) where the word "has" is represented by $w_1$ and "meningitis" by $w_2$. The input value is the word " **has**" and the target value is "**meningitis**" . All node is initialized with the value "0" except the input word that receives has value "1" for the input layer (The blue box in the architecture). The same approach is used for the output layer where the target value is initialized at "1" and the remaining node at "0" (The red box in the architecture). A first a linear operation is operated between the input layer and the hidden layer through the mathematical expression $W^T x$ where $W_T$ is the transpose of the vector representing the weights of the connection between each input node to every single node of the hidden layer and x the input vector. A second similar linear operation is operated between the hidden layer and the output layer and on the top of each output node, the probability estimate through a non-linear operation given by the following softmax formula :

$$h = W_{(1)}^T X \tag{1.1}$$

$h$ is the linear operation between the input layer and the hidden layer.

$$y = softmax(W_{(2)}^T h) \tag{1.2}$$

$y$ is the output by passing the result of the linear operation between the hidden layer and the output layer to a softmax activation function.

The target vector and the output vector are then compared, and the weight is updated through a back-propagation technique until the minimum error is obtained between these two vectors.

# 2. Research Methodology

The first part of the methodology is related to data collection. It contains our process of gathering the data used in this research. The second part consists of presenting the data preparation before the analysis. The third part of the methodology introduces meningitis features and category used in this research. The fourth part includes the presentation of the algorithm that has been used to define the label of the tweet. Figure 2.1 is a summary of the entire methodology.

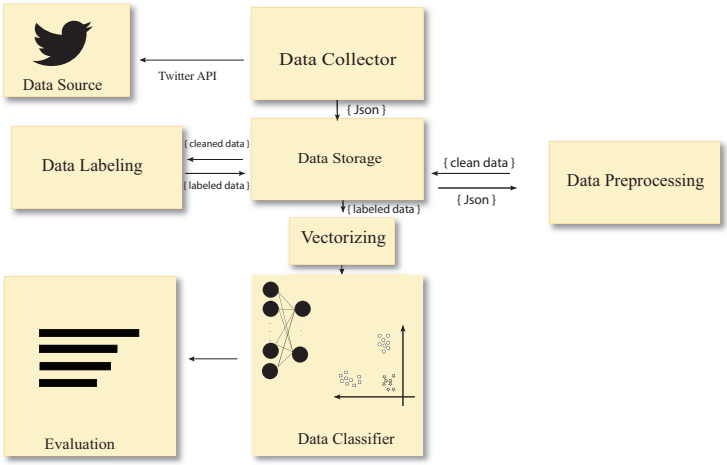



Figure 2.1: Tweet dataset mentioning the keyword meningitis

## 2.1 Data Collection

We collected historical data using the hand chosen keyword "meningitis." We were able to collect 373 765 tweets from 2009 to 2014. Figure 2.2 shows the number of tweets mentioning the keyword "meningitis" with a significant peak in 2012-2014 corresponding to a period when Nigeria has seen an increased rate of meningitis in some of its states. Each entity of tweet is composed of two features: text and date.

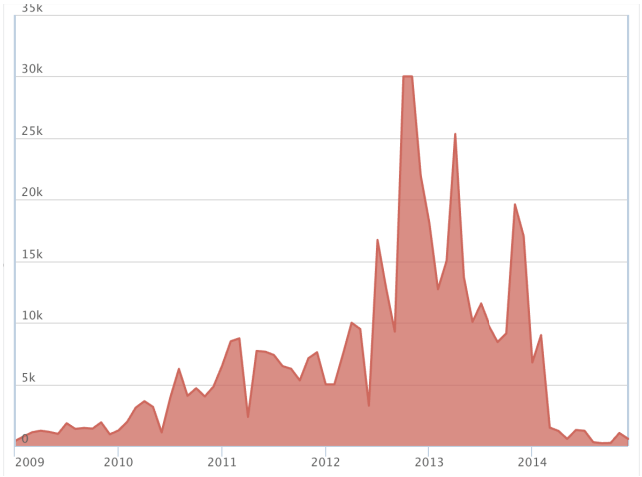



Figure 2.2: Tweet dataset mentioning the keyword meningitis

## 2.2 Data Pre-processing

Twitter data contain a lot of information that affects the quality of data negatively. The presence of this information affects the performance of classifiers in terms of time and memory consumption. The most frequent unnecessary words we found in our dataset are URLs, emoji, hashtags, apostrophes, punctuations and stop words. Our pre-processing tasks removed this information from the collected data.

- **URLs**: The URLs are frequent in the tweets. We used regular expression to identify them. We split the tweet into different tokens, and for each token, we check if it contains a substring "HTTP" , "HTTPS" or "WWW." If so, then we removed that token from the list of tokens, and we join the remaining tokens to reconstruct the sentence.

- **Emojis** : Regarding the limitation of the number of characters for a tweet, users often use emoji to express their feelings. We first decode each emoji into their Unicode representation, and we used the RegEx to find and removed them.

- **Hashtags**: The hashtags are also common in tweets. The purpose of using a hashtag is to make the information retrieval easy. They begin with the special character "#." Here we check the first character of each token and revoke those who start by the special characters mentioned.

- **Apostrophes**: Most of the time people on social network prefers to use the contraction form of the expressions instead of the formal expression. For

example “does not” is often written “doesn’t ”. To solve this problem, we built a dictionary for some of the most common contractions. We used the RegEx to retrieve them and replace by the formal expression.

- **Punctuations**: The punctuations can be “-! ? etc.”We removed all these punctuation except “.”, which is used to delimitate the scope of every tweet in the final corpus.

- **Stop Words**: The stop words are words such as “a, at , i ”etc. Since we are interested in some of them for our analysis such as the pronouns, we defined a list of the words that must not be deleted even if there are considered as stop words.

Beyond the different cleaning described above, we implemented an algorithm to remove duplicated information. Retweeted information is easy to detect. The retweets usually start with the character @RT at the beginning of the tweet. Looking at our dataset a lot of information seems not to be a retweet but a simple copy and paste. The algorithm we defined allowed to detect and remove those type of duplicated information by analyzing them to avoid a biased result during the training stage.

| **Algorithm** Remove duplicated tweet |
|---|
| **Input:** dataset D<br>**Output:** dataset with no duplicated tweet<br><br>**Step1:** receive the tweet.<br>**Step2:** each input sent to the method is compare to the remaining tweet in the dataset using the *difflib* library.<br>**Step3:** for all tweet similar to the input tweet, remove them from the dataset.<br>**Step4:** generate output: csv file with tweets having different sequence of word. |

## 2.3 Training Datasets

The training process defined a label for each tweet which can be Infection, Concern, Vaccine, Campaign, or News. We assigned one of these labels to our tweets based on a list of features we defined as meningitis dictionary. We asserted that the label of a tweet corresponds to the category name and a tweet must have only a feature belonging to that category. For the feature’ s selection, we were inspired by the work done by Lam et al. in their research related to influenza tracking

through Twitter data. [10]. We extended the list of categories by adding the categories News and Campaign. We also extended the list of features presented in the paper published by Lam et al. The following Table 1 shows the list of features we considered in our dictionary.

| Category | List of terms |
|---|---|
| Infection | contracted meningitis, contract meningitis, contracting meningitis, get meningitis, got meningitis, getting meningitis, have meningitis, having meningitis, had meningitis, has meningitis, catch meningitis, caught meningitis, infect meningitis, infected meningitis, recovering meningitis, recovered meningitis |
| Concern | worried, afraid, scared, fear, worry, nervous, dread, dreaded, terrified, panicked, tormented, wondering; |
| Vaccine | meningitis vaccine, meningitis vaccines, meningitis shot, meningitis shots; |
| Campain | raised money, raised funds, raised fund, collected funds, collected fund, collected money, fund raising, support, supported, campaign, raise funds, raise fund, collecting money, donation; |
| News | meningitis outbreak, meningitis alert, meningitis killed, outbreak of; |
| Negation | no sign of, not had, not have, rules him out, rules her out, etc. |
| Self | me, i, myself, we, us; |
| Others Experiencer | you, he, she, they, relative parents (son, aunt, brother etc.), some common English names (John, Zax etc.) |

Table 2.1: Features related to meningitis

### 2.3.1 Infection

The patient may be the person posting the message like in this example [**... i got meningitis**] or someone else illustrates by this [**my friend' s sister had menin-**

**gitis ...**]. Since the patient can explain his or her problem better than someone else, we focus on identifying if the patient is the one who posted the information by checking the subject in the tweet. In this category, we assessed the problem of the negative tweet. Some tweet may be in present or present continuous form but expressing a negation like in this example [**i have no sign of meningitis**]. We identified these tweets by checking the negative expression preceding the keywords 'meningitis' such as no, no sign etc. Since the Infection category seems to depend on some other factors such as the experiencer and the presence or the absence of the negation, we defined a dictionary for each of the elements. The list of keywords we described for the experiencers is essentially pronouns common name in English. We separated this list in two different groups: The terms reflecting the patient itself and then the terms indicating another person.

There exist a bunch of verbs in medical jargon to indicate a disease infection. However, some verbs are more common in our daily conversations. To track the tweet related to meningitis infection, we defined a list of bigrams composed by the keyword meningitis preceding by the verb that reflect the presence of the disease.

### 2.3.2 Concern

The Concern category is the type of messages through which the user expressed his or her concern making a possibility of contracting meningitis. For example [**my teacher scared us by saying that immune weak ppl can get severe infections like meningitis sometimes**]. They can often express compassion towards someone with meningitis. For example, [**worried about my coworker who has meningitis and hoping no one else at work gets it**] or a Concern after a case of meningitis diagnosed in public places such as universities or cities is illustrated by [**we were worried about meningitis there was a case in her school**]. To target this type of tweet, we used a minimum list of verbs relating to concern. We improved the list defined by Alex Lamb et al. [8]

### 2.3.3 Vaccine

The vaccine category refers to tweets expressing the action of having received a vaccine against meningitis. For example, [**i got to get a meningitis shot day for college nervioso**] or waiting to receive it [**... waiting for a shot meningitis for me**].

### 2.3.4 Campaign

This Category of the tweet is about fundraising campaigns to help meningitis patients, such as [**the family of the baby who died from meningitis have**

**raised ... in his memory**] or to support research centers [**me and my fellow trainee journos are raising money for meningitis research**].

### 2.3.5 News

The news is generally the tweet that tends to share information about meningitis outbreak such as [ **“meningitis outbreak on east and west coast only happening to men**].

## 2.4 Training Data Generation Algorithm

The first step is to tag the features defined in our dictionary. Once the expressions are tagged in the tweets, we look for these tags in the second step. If a tag corresponding to a given category is found, then the label of this tweet will be this category. However, it often happens that in a tweet we have multiples tags corresponding to different categories. In this case, we drop this tweet. For the category infection case, we check if the tweet contains a feature indicating that the person is talking about himself. Tweets indicating a case of infection but where the person who posted the tweet is different from the patient are also dropped.

| **Algorithm** Training data generation |
| --- |
| **Input:** dataset D, feature infection, feature vaccine, feature campaign, feature negation, feature concern, feature news, feature self, feature other.<br>**Output:** tweet class value, tweet class.<br><br>**Step1:** receive the tweets.<br>**Step2:** annotate the tweets.<br>**Step3:** for each tweet annotated, check if a feature of a category is found.<br>**Step4:** drop all tweets having multiple different features.<br>**Step4:** drop all tweets having the negation feature.<br>**Step5:** set the class name of the tweet with the name of the class the feature belongs to. |

# 3. Implementation Tools

## 3.1 Jupyter Notebook

This work was conducted using the Python programming language. The integrated development environment used was Jupyter Notebook which an open-source interactive web application for data science and scientific computing. The application can be installed through **Annaconda** available on the official page "https://www.anaconda.com/distribution/ ".

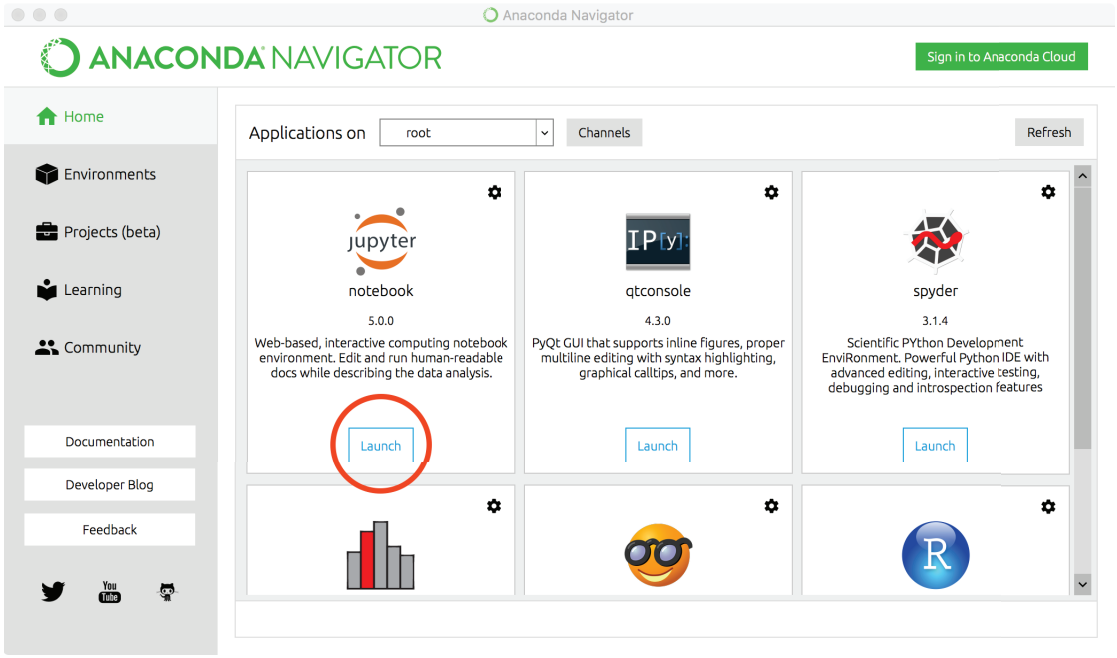


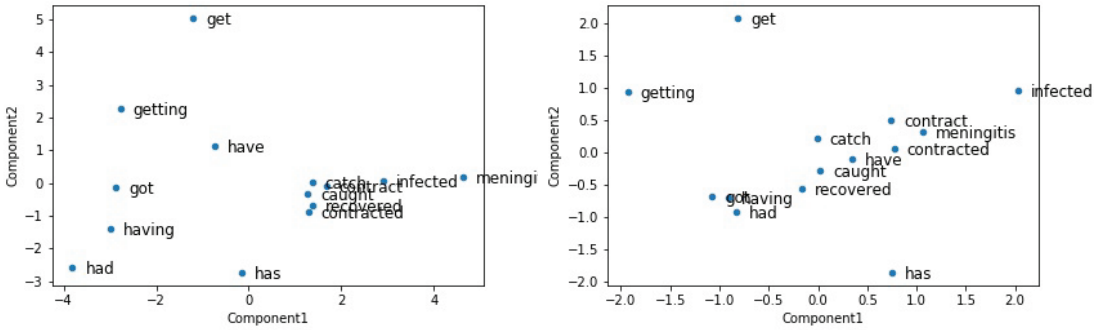

Figure 3.1: Anaconda

## 3.2 scikit-learn

scikit-learn is a python library for performing classification, regression and clustering tasks. [18]scikit-learn support different machine learning algorithms such as Random Forest, Support Vector Machine, Logistic Regression, Multiple Layer Perceptron etc. We used this library to train our four models.

## 3.3 Gensim

Gensim is a library developed in python language integrating several features dedicated to natural language processing. [19]One of the modules that we have used in this work is the word2vec module that we presented in chapter 1. It has the advantage of processing big data faster and more efficiently. It can estimate the similarity between the words and integrates the algorithms CBOW and Skigram. The general idea behind the concept of word2vec being that similar words tend to appear in the same context, we had to test and evaluate which of the two algorithms best represents our tweets in vector space. The two figures below are the results of a test performed using the word *meningitis* and the *features of the infection* category. In this example, we can see that the Skip-gram model has better representation than the CBOW model.




(a) CBOW word representation

(b) Skip-gram word representation

Figure 3.2: Comparison between Skip-gram and CBOW

## 3.4 spaCy

The spaCy library is a python library also dedicated to natural language processing and integrates several modules including the Tagger module. As we write this document, the spaCy library supports 15 languages. We use this module to tag the features we have defined in the dictionary. [20]

# 4. Results and Discussion

In this section, we evaluate in a first step the precision of the algorithm that we have developed for the generation of training data by using a human expert. The results obtained after this step are presented in section 4.1. The second step was to use different approaches to convert the tweets into numerical data and to train the four models shown in chapter 1. We compared the results of the models according to their precision, F1-Score in a table and also from a boxplot in section 4.2.

## 4.1 Training Data Generation Result

The algorithm described in section 2.4 allowed us to get our training data as shown in the figure above. We found a small volume of the quantity of the category Concern and Campaign compared to other categories Infection, News and Vaccine. This unequal distribution can be a source of results biased; therefore during the training of the classifiers, we proceed to a balance adjustment between the categories by decreasing the number of tweets of classes News, Vaccine and Infection.

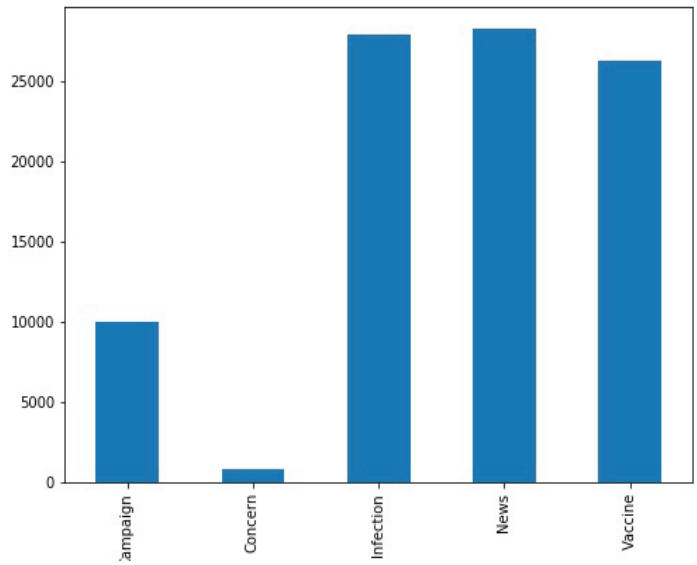



Figure 4.1: Tweet dataset mentioning the keyword meningitis

## 4.2 Classification Results

We estimated the precision score, f1-score and recall to evaluate our models. The **precision** is the ratio tp / (tp + fp) where tp is the number of true positives and fp the number of false positives. The precision is intuitively the ability of the classifier not to label as positive a sample that is negative.
The **recall** score indicate the ability of the classifiers to find all tweets correctly classified. The more the value is close to 1, the more the classifier is good. is the ratio tp / (tp + fn) where tp is the number of true positives and fn the number of false negatives. The recall is intuitively the ability of the classifier to find all the positive samples.
The **f1-score** can be interpreted as a weighted average of the precision and recall, where an F1 score reaches its best value at 1 and the worst score at 0. The relative contribution of precision and recall to the F1 score are equal. The formula for the F1 score is: F1 = 2 * (precision * recall) / (precision + recall).
To clarify all these calculation methods, consider the following confusion matrix generate for the RF model using the TF-IDF vectorization model

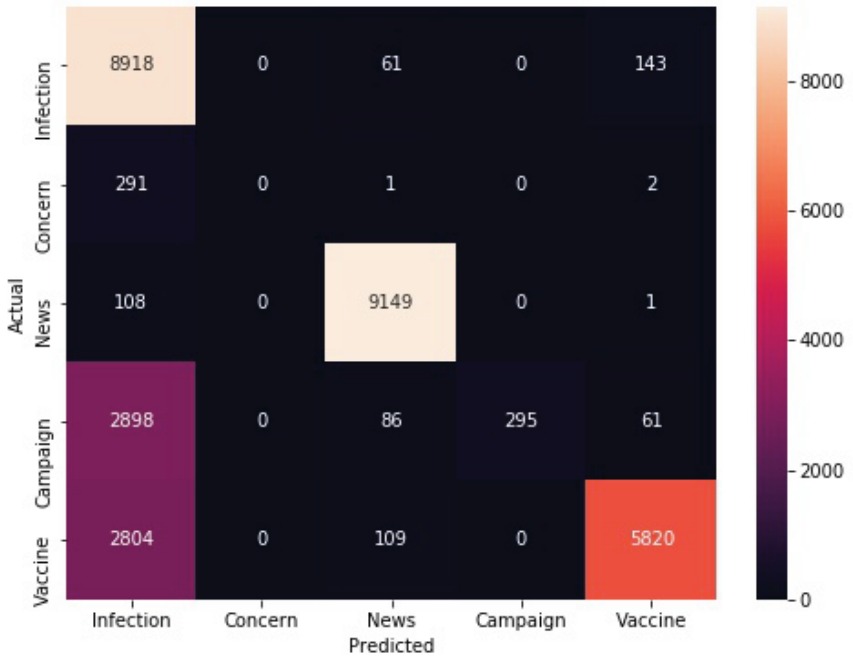



Figure 4.2: Confusion Matrix for the RF model

$$Precision = \frac{8918}{8918 + (291 + 291 + 2898 + 2804)} \approx 0.59$$

$$Recall = \frac{8918}{8918 + (0 + 61 + 0 + 143)} \approx 0.98$$

$$F1 - Score = \frac{2 * 0.59 * 0.98}{(0.59 + 0.98)} \approx 0.74$$

| | | Precision | F1-Score | Recall |
|---|---|---|---|---|
| Infection | SVM | 1.00 | 1.00 | 1.00 |
| | ANN | 0.99 | 0.99 | 1.00 |
| | RF | 0.59 | 0.74 | 0.98 |
| | LR | 0.98 | 0.99 | 0.99 |
| Concern | SVM | 1.00 | 0.99 | 0.98 |
| | ANN | 1.00 | 0.98 | 0.96 |
| | RF | 0.00 | 0.00 | 0.00 |
| | LR | 1.00 | 0.71 | 0.55 |
| Vaccine | SVM | 1.00 | 1.00 | 1.00 |
| | ANN | 1.00 | 1.00 | 1.00 |
| | RF | 0.97 | 0.16 | 0.09 |
| | LR | 1.00 | 1.00 | 1.00 |
| Campaign | SVM | 1.00 | 1.00 | 1.00 |
| | ANN | 1.00 | 1.00 | 1.00 |
| | RF | 1.00 | 0.16 | 0.09 |
| | LR | 1.00 | 1.00 | 0.99 |
| News | SVM | 1.00 | 1.00 | 1.00 |
| | ANN | 0.98 | 0.97 | 0.99 |
| | RF | 0.97 | 0.76 | 0.67 |
| | LR | 0.99 | 0.99 | 0.99 |

Table 4.1: Precision, F1-Score, Recall using TF-IDF

The result of the table above shows better results for the models ANN, SVM and LR except for the model Random Forest. This situation could be related to Random Forest‘’s particularity. Even though the Random Forest models is efficient through his ability to learn data faster, it has weaknesses in predicting data beyond trained data.

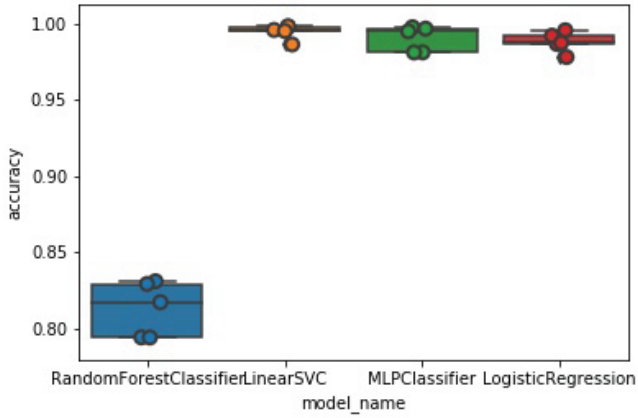



Figure 4.3: Graphical representation of the result using TF-IDF model

For a better visualization of the accuracy distribution, we plotted the accuracy of all the models we used in this work. The following figure is a graphical representation of our result. This plot shows us that the model SVM presents a better accuracy followed by the model ANN and the model LR.

| | | Precision | F1-Score | Recall |
|---|---|---|---|---|
| Infection | SVM | 0.95 | 0.97 | 0.99 |
| | ANN | 0.96 | 0.97 | 0.97 |
| | RF | 0.82 | 0.86 | 0.90 |
| | LR | 0.95 | 0.97 | 0.98 |
| Concern | SVM | 0.00 | 0.00 | 0.0 |
| | ANN | 0.31 | 0.29 | 0.27 |
| | RF | 0.00 | 0.00 | 0.00 |
| | LR | 0.00 | 0.00 | 0.00 |
| Vaccine | SVM | 0.99 | 0.99 | 0.99 |
| | ANN | 0.99 | 0.99 | 0.99 |
| | RF | 0.93 | 0.92 | 0.91 |
| | LR | 0.99 | 0.99 | 0.99 |
| Campaign | SVM | 0.99 | 0.99 | 0.99 |
| | ANN | 0.99 | 0.99 | 0.99 |
| | RF | 0.95 | 0.85 | 0.86 |
| | LR | 0.99 | 0.99 | 0.98 |
| News | SVM | 0.98 | 0.98 | 0.98 |
| | ANN | 0.99 | 0.99 | 0.99 |
| | RF | 0.84 | 0.87 | 0.87 |
| | LR | 0.98 | 0.98 | 0.98 |

Table 4.2: Precision, F1-Score, Recall using Skip-gram embedding model

Like the TF-IDF model, the results of the Skip-gram model also show us good results during the learning phase. However, the box plot shows us that during the test phase, all the models present bad precisions indicated by the points called outliers at the bottom of each boxplot.

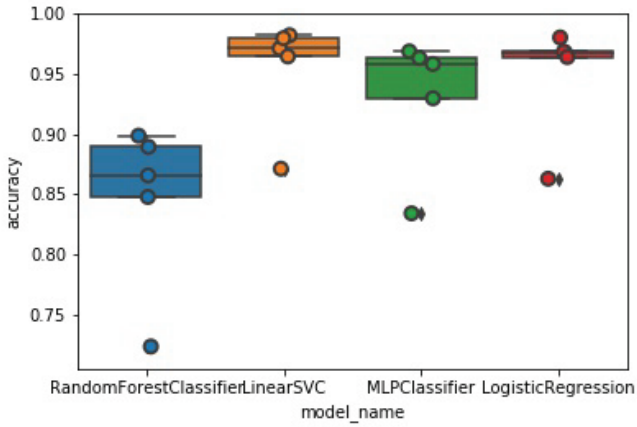



Figure 4.4: Graphical representation of the result using Skip-gram model

# Conclusion

The good accuracy we obtained after training our model can be explained by the quality and the quantity of the dataset. We spent roughly 60% of this work cleaning and normalizing the tweets. This step allows us to get a good quality of the datasets which contribute to increasing the accuracy of the model. Using the approach of Skip-gram we were able to have a rich representation of the different tweets in a vector space that contributes to learning very faster the different models. However, the embedding model using TF-IDF performed good accuracy.

The keywords used in our vocabulary were too small, and they have been used to label the tweet. The variation of the word is tiny, therefore allowed the TF-IDF to emphasize the features. This situation including the fact that the tweets are too short for a better representation using skip-gram explains the reason why we obtain a good accuracy during the classification test. The result relies on the number of features than the words similarities; therefore a tweet can be misclassified if the user uses a synonym of the features presented in Table 1.

Every year meningitis affects many people in Africa. From Senegal to Ethiopia including Burkina Faso are well known as meningitis belt. Our future work will consist of evaluating our model on the dataset from these countries. However, depending on the counties the main languages spoken vary from English to French. Since in this paper we deal with on English, we will consider the French language in future work. We intend to use an ontology implemented by Cédric BÉRÉ et al. as vocabulary to solve the problem multilingual processing problem in one hand and another side to process synonyms and shorts terms expression in the different tweets.[15]

# Bibliography


[1] Mowery J, *Twitter Influenza Surveillance: Quantifying Seasonal Misdiagnosis Patterns and their Impact on Surveillance Estimates*, Online J Public Health Inform, 2016.

[2] Paul M., Dredze M., Broniatowski D., and Generous N., *Worldwide influenza surveillance through twitter*, In AAAI, 2015.

[3] Culotta A, *Towards detecting influenza epidemics by analyzing twitter messages*, KDD Workshop on Social Media Analytics, 2010.

[4] Broniatowski, David A., Michael J. Paul, and Mark Dredze, *National and Local Influenza Surveillance through Twitter: An Analysis of the 2012-2013 Influenza Epidemic* , PLoS ONE, 2013.

[5] United Nations Department of Economic and Social Affairs/Population Division, *World Population Prospects*, 2017, p. 17.

[6] Logan SAE, MacMahon E, *Viral meningitis*, BMJ, 2008.

[7] Savory EC, Cuevas LE, Yassin MA, Hart CA, Molesworth AM, Thomson MC, *Evaluation of the meningitis epidemics risk model in Africa*, Epidemiol Infect 134:1047- 1051, 2006.

[8] Basil Benduri Kaburi, Chrysantus Kubio, Ernest Kenu and Donne Kofi Ameme, et al, *Evaluation of bacterial meningitis surveillance data of the northern region, Ghana, 2010-2015* , Pan Afr Med J, 2017.

[9] Lingani C, Bergeron-Caron C, Stuart JM, et al, *Meningococcal meningitis surveillance in the African meningitis belt, 2004 - 2013*, Clin Infect Dis, 2015.

[10] Lamb, A., Paul, M. J., and Dredze, M., *Separating fact from fear: Tracking flu infections on Twitter*, In North American Chapter of the Association for Computational Linguistics (NAACL), 2013.

[11] Salton, G., Wong A., and Yang C., *A vector space model for automatic indexing*, Communications of the ACM 18, 1975.

[12] Mikolov, Tomas, Chen, Kai, Corrado, Greg, and Dean, Jeffrey, *Efficient estimation of word representations in vector space*, arXiv preprint arXiv:1301.3781, 2013.

[13] Bengio, Yoshua, Schwenk, Holger, Senecal, Jean-Sebastien, Morin, Fréderic, and Gauvain, Jean-Luc, *Neural probabilistic language models, Innovations in Machine Learning*, Springer, 2006.

[14] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, J. Dean, *Distributed Representations of Words and Phrases and their Compositionality*, Proceedings of NIPS 2013, 2013.

[15] Béré W.R.C., Camara G., Malo S., Lo M., Ouaro S., *Towards Meningitis Ontology for the Annotation of Text Corpora*, M. F. Kebe C., Gueye A., Ndiaye A. (eds) Innovation and Interdisciplinary Solutions for Underserved Areas. CNRIA 2017, InterSol 2017. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, vol 204. Springer, Cham, 2018.

[16] World Health Organization, *Meningococcal Meningitis*, World Health Organization, www.who.int/en/news-room/ fact-sheets/detail/meningococcal-meningitis, 9 February 2018.

[17] World Health Organization, *Meningococcal Meningitis*, World Health Organization, www.who.int/gho/epidemic_diseases/meningitis/en/, 3 July 2015.

[18] Lars Buitinck and Gilles Louppe and Mathieu Blondel and Fabian Pedregosa and Andreas Mueller and Olivier Grisel and Vlad Niculae and Peter Prettenhofer and Alexandre Gramfort and Jaques Grobler and Robert Layton and Jake VanderPlas and Arnaud Joly and Brian Holt and Gaël Varoquaux, *API design for machine learning software: experiences from the scikit-learn project*, ECML PKDD Workshop: Languages for Data Mining and Machine Learning, 2013, p108-122.

[19] Radim Řehůřek and Petr Sojka, *Software Framework for Topic Modelling with Large Corpora*, Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks, 2010,p45-50.

[20] Honnibal, Matthew AND Montani, Ines, *spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing*,To appear, 2017.

[21] Rifkin, Ryan M. and Aldebaro Klautau *In defense of one-vs-all classification*, Journal of Machine Learning Research, 2004.