

*Sentiment Analysis on anticovid-19 vaccination*

*Project of the Data Mining course 2020/2021*

*Edited by:*

*Castrignano Matteo*

*Dallatomasina Erica*

Sommario

[Introduction 3](#_Toc74154056)

[Data collection 4](#_Toc74154057)

[Pre-processing 5](#_Toc74154058)

[Supervised learning stage 6](#_Toc74154059)

[Results for n-gram = (1,1) 6](#_Toc74154060)

[Results for n-gram = (1,2) 7](#_Toc74154061)

# Introduction

Our goal was to create an application that could analyze tweets to discover people's opinion on antivid-19 vaccines. It collects tweets via a scraper that will later be labeled in one of three classes (negative, positive, or neutral) depending on the content of the tweet itself. To do this it was necessary to train a classifier with previously labeled tweets and choosing a particular model. The whole thing was made in *Python* using the *SK-learn* library.

# Data collection

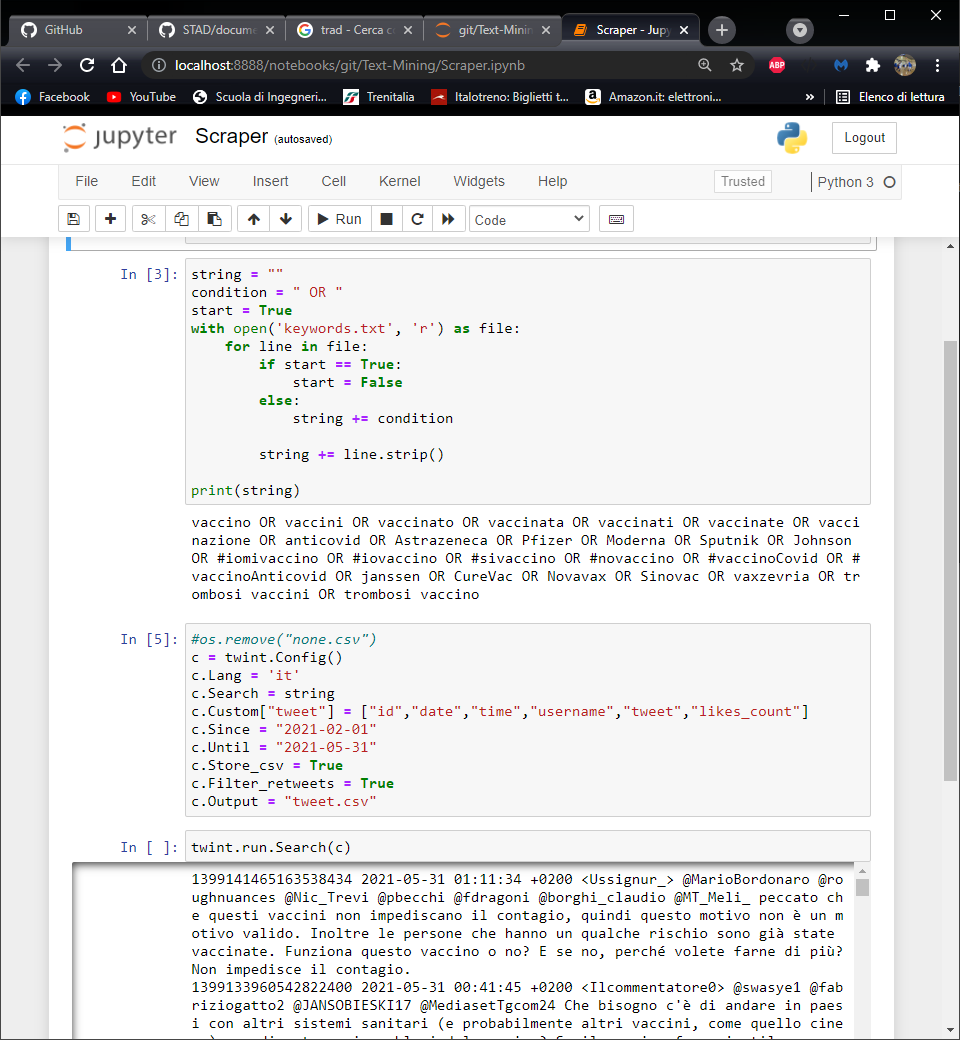
The raw tweets are scraped from Twitter using the *Twint* library. Using the functions offered from this library, we can set some options, but we have set only:

* The date, in order to scrape tweets posted only on a specific period.
* The language, in order to scrape only Italian tweet.
* Some keywords to scrape the tweets.

The scraped tweets are characterized by the following attributes (whose name is self-explanatory):

* id
* date
* time
* username
* tweet
* likes\_count

All information about *Twint* can be retrieve from GitHub repository: <https://github.com/twintproject/twint>



# Pre-processing

In the pre-processing phase raw tweets are elaborated in order to clear them. In particular, we do the following steps:

* Remove all the links, emoji and images contained in the tweets
* Remove all the hashtag symbols (#) from the tweets. For example, if a tweet contains “#word” we transform it in “word”. We decided to maintain the word associated to the hashtag because it is often important to understand the stance of the tweet (for example there can be hashtags like #iomivaccino or #vaccinomorte that are indicative of the opinion about vaccines expressed in the tweet)
* Remove all the mentions to other users in the tweet
* Put the text of the tweets in lower case
* Remove all the punctuation
* Remove all the extra-spaces

***Before***

*"@valy\_s È una cosa normale. L'immunità di gregge è a senso unico. Se vaccini tutti non sconfiggerai mai la malattia. Quando lo capiranno, torneremo a vivere normalmente! Se vuoi fermare la pandemia, studiala e crea una medicina. Il vaccino serve a poco!"*

***After***

*e una cosa normale l immunita di gregge e a senso unico se vaccini tutti non sconfiggerai mai la malattia quando lo capiranno torneremo a vivere normalmente se vuoi fermare la pandemia studiala e crea una medicina il vaccino serve a poco*

* 1. *Tokenization, stop-word filtering and stemming*
  2. *Stem by relevant*
  3. *Classification and evaluation*

# Supervised learning stage

To train the model different experiments were performed, considering different classifiers and different text representations techniques. We decided to test 7 different classifiers:

* Multinomial classifier
* Bernoulli classifier
* Linear support vector classifier
* Random forest classifier
* K-nn classifier
* Adaboost classifier
* Decision tree classifier

In particular, before applying the model to the dataset we have to perform the following steps:

1. Extract tokens from tweets (**tokenization phase**). We use the BOW representation testing different values of n-gram (we consider only unigrams and unigrams and bigrams together)
2. Remove stop-words (**stop-word filtering**) from the set of extracted tokens[[1]](#footnote-1)
3. Eventually perform a **stemming phase**, in order to reduce each token to its “root form” [[2]](#footnote-2)
4. Perform a **f-idf transformation**, where the importance of each stem is computed using the Inverse Document Frequency

The experiments were performed using a 10-fold cross validation procedure. The dataset used for the supervised stage was manually labelled and is made up of 906 instances uniformly distributed into the three classes (so we have 302 instances for each class). Using a 10-fold cross validation procedure, at each iteration the model is constructed and tested using 30 instances as test set, and the remaining instances as training set.

In the following we report the results obtained testing the different classifiers using the tokenization with only unigrams and with unigrams and bigrams. In each one of these experiments we considered as stop-words the set of Italian stop-words offered by the sklearn library and we performed the stemming phase (this is the configuration that in our case gave the better results, for the sake of brevity we will report only the results of those experiments).

## Results for n-gram = (1,1)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *Classifier* | *Class* | *Precision* | | *Recall* | *F-score* | *Accuracy* | *Time (s)* | *Confusion matrix* |
| MultinomialNB() | *0* | *0.59* | *0.62* | | *0.6* | *0.54* | *1.78* | *188 67 47*  *74 138 90*  *59 82 161* |
| *1* | *0.48* | *0.46* | | *0.47* |
| *2* | *0.54* | *0.53* | | *0.54* |
| BernoulliNB() | *0* | *0.57* | | *0.67* | *0.61* | *0.53* | *1.92* | *203 60 39*  *90 130 82*  *66 87 149* |
| *1* | *0.47* | | *0.43* | *0.45* |
| *2* | *0.55* | | *0.49* | *0.52* |
| LinearSVC | *0* | *0.57* | | *0.58* | *0.58* | *0.51* | *2.42* | *174 77 51*  *73 138 91*  *56 96 150* |
| *1* | *0.44* | | *0.46* | *0.45* |
| *2* | *0.51* | | *0.49* | *0.51* |
| RandomForestClassifier() | *0* | *0.56* | *0.58* | | *0.56* | *0.51* | *3.09* | *176 79 47*  *85 135 82*  *56 98 148* |
| *1* | *0.43* | *0.45* | | *0.44* |
| *2* | *0.53* | *0.49* | | *0.51* |
| KNeighborsClassifier()  N = 5 | *0* | *0.48* | *0.66* | | *0.55* | *0.47* | *1.76* | *199 70 33*  *113 125 64*  *105 97 100* |
| *1* | *0.43* | *0.41* | | *0.42* |
| *2* | *0.51* | *0.33* | | *0.4* |
| AdaBoostClassifier | *0* | *0.52* | | *0.49* | *0.5* | *0.45* | *2.56* | *147 99 56*  *78 140 84*  *56 128 118* |
| *1* | *0.38* | | *0.46* | *0.41* |
| *2* | *0.46* | | *0.39* | *0.42* |
| DecisionTreeClassifier() | *0* | *0.5* | | *0.52* | *0.51* | *0.45* | *2.53* | *158 74 70*  *91 118 93*  *68 102 132* |
| *1* | *0.4* | | *0.39* | *0.4* |
| *2* | *0.45* | | *0.44* | *0.44* |

## Results for n-gram = (1,2)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *Classifier* | *Class* | *Precision* | | *Recall* | *F-score* | *Accuracy* | *Time (s)* | *Confusion matrix* |
| MultinomialNB() | *0* | *0.61* | | *0.64* | *0.63* | *0.55* | *3.38* | *193 72 37*  *65 149 88*  *57 88 157* |
| *1* | *0.48* | | *0.49* | *0.49* |
| *2* | *0.56* | | *0.52* | *0.54* |
| BernoulliNB() | *0* | *0.58* | | *0.66* | *0.52* | *0.55* | *3.84* | *198 66 38*  *80 146 76*  *63 89 150* |
| *1* | *0.49* | | *0.48* | *0.48* |
| *2* | *0.57* | | *0.5* | *0.53* |
| LinearSVC | *0* | *0.58* | | *0.59* | *0.58* | *0.52* | *4.63* | *177 77 48*  *72 140 90*  *57 94 151* |
| *1* | *0.45* | | *0.46* | *0.46* |
| *2* | *0.52* | | *0.5* | *0.51* |
| RandomForestClassifier() | *0* | *0.55* | *0.57* | | *0.56* | *0.5* | *4.44* | *173 89 40*  *80 147 75*  *64 103 135* |
| *1* | *0.43* | *0.49* | | *0.46* |
| *2* | *0.54* | *0.44* | | *0.49* |
| KNeighborsClassifier()  N = 5 | *0* | *0.47* | | *0.68* | *0.56* | *0.48* | *3.23* | *206 67 29*  *127 120 55*  *105 84 113* |
| *1* | *0.44* | | *0.4* | *0.42* |
| *2* | *0.57* | | *0.37* | *0.45* |
| DecisionTreeClassifier() | *0* | *0.45* | *0.5* | | *0.48* | *0.42* | *3.41* | *152 85 65*  *94 114 94*  *89 96 117* |
| *1* | *0.39* | *0.38* | | *0.38* |
| *2* | *0.42* | *0.39* | | *0.4* |
| AdaBoostClassifier | *0* | *0.5* | | *0.43* | *0.47* | *0.42* | *4.51* | *131 114 57*  *79 158 65*  *51 166 95* |
| *1* | *0.37* | | *0.52* | *0.43* |
| *2* | *0.44* | | *0.31* | *0.36* |

1. *Concept drift*

1. We made some experiments in which we defined different sets of stop-words, but we saw that the better results was obtained by using the set of Italian stop-words offered by the sklearn library. For the sake of brevity we will report only the results obtained with the experiments performed with the default stop-words set. [↑](#footnote-ref-1)
2. The experiments we did shown that better results are obtained by performing the stemming phase. For the sake of brevity we will report only the results obtained in the experiments that consider this phase. [↑](#footnote-ref-2)