Methodology for the challenge

Sentiment Analysis

Filipe Afonso

28/10/2018

Data

A train file of 31962 tweets with id, description of the tweet and the label taking two values depending on whether the tweet is considered 0 : not hateful, 1 : hateful

id	label	tweet
1	0	@user when a father is dysfunctional and is so selfish he drags his kids into his dysfunction. #run
2	0	@user @user thanks for #lyft credit i can't use cause they don't offer wheelchair vans in pdx. #disapointed #getthanked
3	0	bihday your majesty
4	0	#model i love u take with u all the time in urð±!!! ðððð¦ð¦

A test file of 17197 tweets. We do not have the "label" column because we have to predict it.

id	tweet
31963	#studiolife #aislife #requires #passion #dedication #willpower to find #newmaterialsâ¦
31964	@user #white #supremacists want everyone to see the new â #birdsâ #movie â and hereâs why

Problematic

Analyzing the train file, predict which tweets in the test file are hateful or not.

Strategy 1 : Methodology using standard methods and programming with Python

➢ in accordance with the instructions of the challenge

a. Cleaning of the train and test data files

- string.punctuations, special characters, words with misinterpreted characters, small words (<3characters)

b. Hashtags and “regular” words are separated to allow different weights in the scoring model. Indeed, hashtags are known to be valid words or expressions so that it is good to preserve them. Cleaned train and test files are saved for reuse in the following runs. Data are saved with the following format :

tweet	hate	hashtag
1	0	run
2	0	lyft
2	0	disapointed
2	0	getthanked
4	0	model
5	0	motivation
6	0	allshowandnogo

c. Calculating, for the train file, the number of iterations (support or frequency) of all the words (regular and hashtags) for the ‘hate” labels {0,1}

- pd.crosstab(trainw['word'],trainw['hate']).

d. Deducing empirical conditional probabilities : P(h=1/w)

where h=1 means that hate label=1 and w is :

(1) a "regular" word
(2) or a hashtag
(3) or a sequence of two or three consecutive words (regular or hastag)

item	nb0	nb1	proba
2016election	0	1	100.0
2016in4words	0	12	100.0
ableism	0	4	100.0
abuse	2	2	50.0
administration	0	1	100.0
adultery	0	4	100.0
afghanistan	1	1	50.0

NB : Here I prefer conditional probabilities than Tf*IDF because we particularly want to differentiate tweets with label hate=1

e. Scoring the tweet

We note A the set of "regular" words, B the set of hashtags,

A tweet $t = t_A \cup t_B \in T$ where t_A (resp. t_B) is the set of regular words (resp. hashtags) in the tweet t.

$$score(t) = p_A \times \sum_{a \in t_A} P(h=1|a) + p_B \times \sum_{b \in t_B} P(h=1|b) + p_C \times \sum_{\substack{c_i \in t \\ c_{i+1} \in t}} P(h=1|c_i c_{i+1})$$

with $\sup(a) \geq minSup$ and $\sup(b) \geq minSup$ and $\sup(c) \geq minSup$

$with\ P(h=1|a) \geq minP$ and $P(h=1|b) \geq minP$ and $P(h=1|c_i c_{i+1}) \geq minP$

and where we have 5 given parameters :

- p_A, p_B, p_C are weights according to the importance given to "regular" words, hashtags and sequence of 2 words respectively (optional parameters);
- minSup (minimum support or frequency) and minP (minimum probibility).

The prediction h(t) is deduced with

$$h(t) = 1\ if\ score(t) \geq minScore$$

- minScore : given parameter minmum score

f – Fixing the combination of parameters minSup, minP, minScore

The train file is divided into two files, the first with 2/3 of the data and the second with 1/3 of the data.

The model is calculated, varyng the parameters, with the first file and is tested using the predictions from the second file. The selected list of parameters optimizes the predictions according to the F1 criterion suggested by the challenge :

True Positives (TP) - These are the correctly predicted positive values which means that the value of actual class is yes and the value of predicted class is also yes.
True Negatives (TN) - These are the correctly predicted negative values which means that the value of actual class is no and value of predicted class is also no.
False Positives (FP) – When actual class is no and predicted class is yes.
False Negatives (FN) – When actual class is yes but predicted class in no.
Precision = TP/TP+FP
Recall = TP/TP+FN

F1 Score = 2*(Recall * Precision) / (Recall + Precision)

g- Application of the model, with the selected combination of parameters, to the test file.

Results

A score F1 ≅ 70%

Strategy 2 : Symbolic Data Analysis Methodology using SYR software

➢ using my own algorithms and software

Preamble

I use original methods derived from recent academic research and from R&D at SYMBAD based on the research field of Symbolic Data Analysis (SDA) using SYR software for SDA (see for example (afonso et al. 2018), (Billard & Diday, 2006) and user manuals of the SYR software).

The aim of SDA is to analyze classes (or groups) of individuals, rather than the individuals themselves**, taking into account their internal variation, complexity and diversity.** For example, comparing the success of football players is not the same thing than analyzing the success of their teams. Teams are groups of players, managers, with specific history, etc.

The originality of SDA methodology lies on the two stages of the data analysis in the SYR software:

- The first step involves the fusion and the reduction of the data into classes described by informative aggregated data called symbolic data. The method is able to merge and aggregate data from multiple databases into a single data table of symbolic data. The data to be merged and reduced can be massive or not, heterogeneous (Quantitative + Qualitative + Text + temporal data + etc.) and multi-source (one or more databases + web data + sensor readings + environmental data, etc.).
  For example, data on football players (age, weight, goals, previous teams, ...) are aggregated up to the level of the team considered as a class. **The class will not be described by means or standard deviations** (on age, weight, ...) **but by variables with variation like intervals, bar-charts, histograms, distributions, etc. called symbolic data.**

- The second step is then the analysis of the obtained symbolic data table by specific advanced statistical methods (dissimilarities, clustering, decision trees, factorial analysis, etc.) extended to this kind of reduced and merged data table. It is then possible to find new correlations between data from different databases. This automatic processing of large data also becomes effective with the prior data reduction.

SDA leads at least to the following advantages: studying the data by units given at the needed level of generalization (classes); summarizing by reducing the loss of information; reducing the number of individuals and the missing data reducing; solving confidentiality questions, facilitating interpretation of results and improving decisions, transforming complex data with unstructured data tables and unpaired variables in a structured symbolic data table with paired symbolic variables; being suitable to highly complex industrial data.

Symbolic Data Analysis – Méthodology in 2 steps

step 1 : Creation of the SDA table describing the tweets.

From the initial data file, we create classical data table 1a where each unit is a word from a tweet (one row by word) described by the label hate {0,1} and the type of the word {regular, hashtag}. For each word w, we then add two other variables “support” (sup(w)=card(w)) and “proba_h1” (conditional probability P(h=1/w)). We obtain table 1b.

From this table 1b, we create the symbolic data table 2 describing the unit “tweet” using the SYR software. Each tweet is described by a bar-chart-valued variable of the type of the words

in the tweet (ex. 70% regular, 30% hashtag). “Support” and “proba_h1” are aggregated up to histogram-valued variables where the categories of the histograms are levels of “support” and “proba_h1” respectively. We obtain two histogram-valued variables supk {support ∈ +500, [200,500], [100,200], [50,100], [25,50], [10,25], [5,9], [2,4], 1} and probak {proba_h1 ∈ 100, [90,99], [80_89], [65,79], [50,64], [40,49], [25,39], [10,24], <10. Thus, for each tweet, we have the distribution of its words in the respective levels of supports and conditional probabilities.

Table 1 a : classical data table describing each "tweet x word"

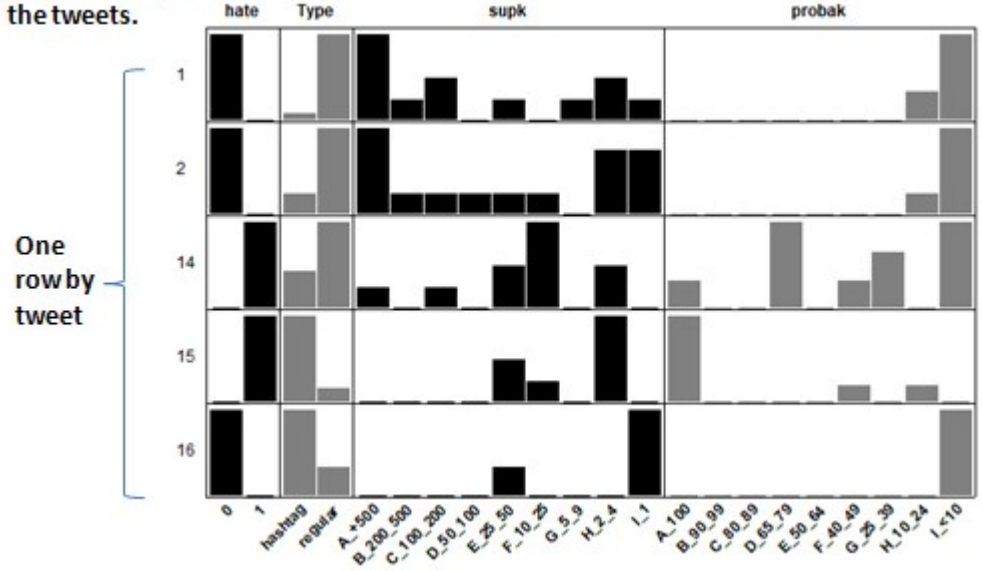
tweet	hate	word	Type
1	0	kids	regular
1	0	father	regular
1	0	dysfunctional	regular
1	0	selfish	regular
2	0	cause	regular
2	0	offer	regular
2	0	credit	regular
9	0	love	regular
9	0	won	regular
9	0	land	regular
9	0	cleveland	hashtag
...			

+

support	proba_h1
143	4,90
130	3,08
1	0,00
7	0,00
72	4,17
26	7,69
14	7,14
1187	1,26
54	14,81
27	22,22
12	25,00
...	

Table 1 b : addition of new variables

Creation of a Symbolic Data Table for the description of the tweets.



One row by tweet

Table 2 : Symbolic data table of the tweets described by bar-chart valued variables

We then obtain a table where each row is associated with a tweet described with variables with variation. Indeed, in each cell, we do not have necessarily a single value but we can have values like bar-charts and histograms.

The tweets are described with comparable descriptions. The table is small compared to the initial data file as the number of categories of the variables is reduced. Moreover, we would have the same results with longer texts rather than tweets.

In the following sections, I use data analysis algorithms extended to these kind of tables.

Step 2 – Analysis of the symbolic data tables with algorithms extended to symbolic data from SYR software

Clustering of the symbolic data table using a kind of k-means extended to symbolic data.

Using the SYR software, we obtain table 3 describing the clusters of tweets in the same way than the tweets (with symbolic data). Each cluster is associated with either the label hate = 0 or the label hate = 1.

The best results according to the within-cluster inertia curve is around 60 clusters.

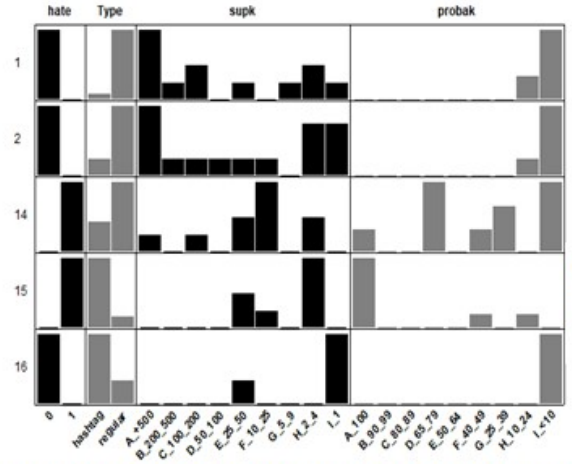


Table 2 : Symbolic data table of the tweets described by bar-chart valued variables

Clustering

extended to symbolic data

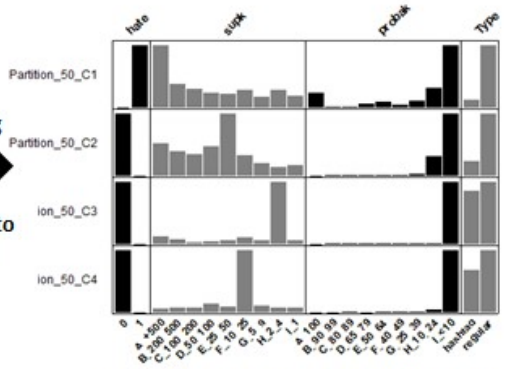
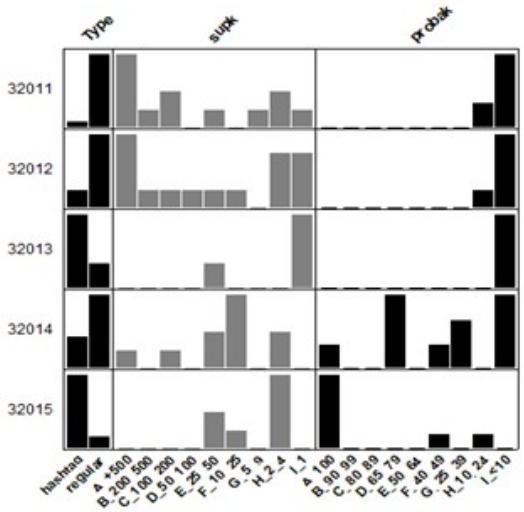


Table 3 : Symbolic data table describing the clusters of tweets obtained with a kind of k-means clustering extended to symbolic data

Allocation of the new tweets (test file) into their classes using the L1 distance extended to symbolic data. Deduction of the label hate {0, 1} and calculation of the F1 score.

The symbolic data table 4 describing the tweets from the test file is created in the same way as for the train file (table 2) using for each word of the test file the "support" and "proba_h1" calculated in the train file. In this first approach, the words in the test file that are missing in the train file are considered with support=1 and proba_h1=0.

Each tweet from table 4 is then allocated to its cluster of tweets (in table 3) using L1 distance extended to bar-chart valued variables. For each tweet, label hate {0 or 1} is then deduced and overall F1 score is calculated.



Allocation

Prediction

using L1 distance extended to bar-chart

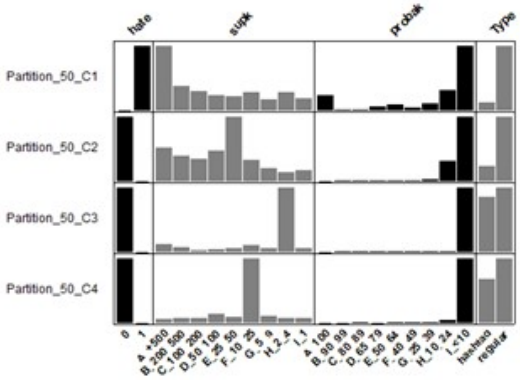


Table 3 : cluster of tweets obtained from the train file

Table 4 : Symbolic data table of the tweets obtained from the test file

Dissimilarity measures for symbolic data

Examples between two histograms :

$$A_i = \left\{ \left[a_{i1,\inf}, a_{i1,\sup} \right[, p^a_{i1}; \ldots; \left[a_{iLj,\inf}, a_{hL} \right[, p^a_{iL,\sup} \right\}$$

$$B_i = \left\{ \left[b_{i1,\inf}, b_{i1,\sup} \right[, p^b_{i1}; \ldots; \left[b_{iLj,\inf}, b_{hL} \right[, p^b_{iL,\sup} \right\}$$

with L = nb of categories constituing the domain of the categorical multi-valued variable

- **The L_1 distance is defined by :**

$$\delta(A_i, B_i) = \sum_{\upsilon=1}^{L} \left| p^a_{i\upsilon} - p^b_{i\upsilon} \right|$$

Results

A score F1 ≅ 70%

Conclusion

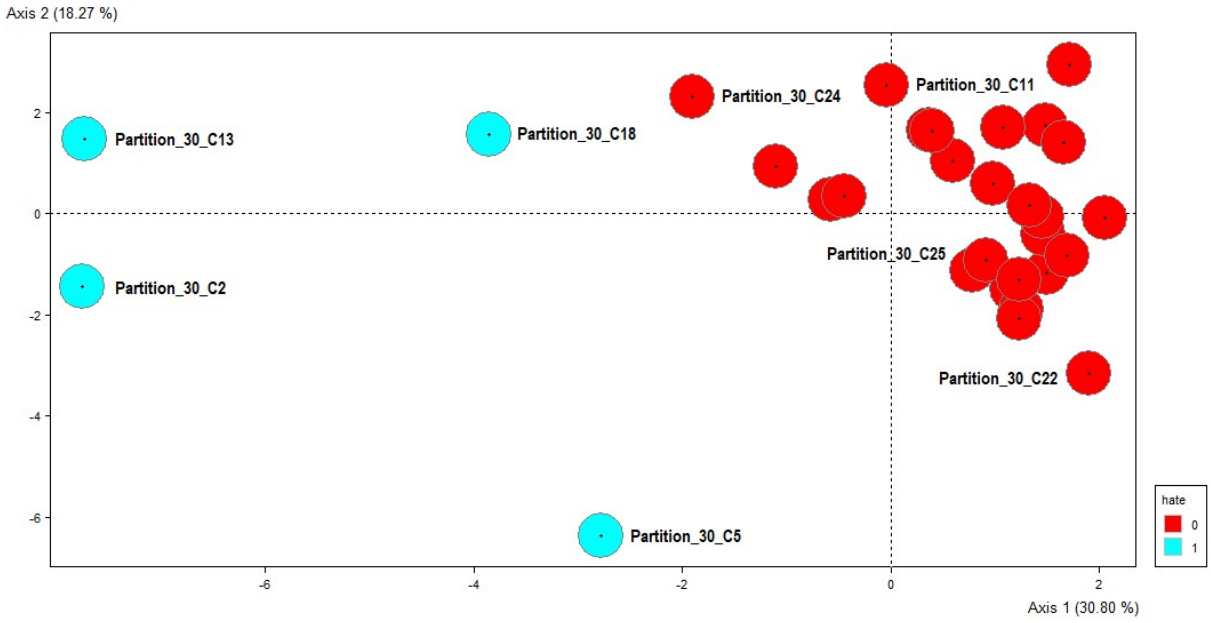
The scores are similar for the two methods. Nevertheless, for the second method we associate a comparable description to each tweet based on statistics calculated on the words. The resulting symbolic data table is small compared to initial data file. The size of this table will not increase if we want to analyze longer texts rather than tweets.

How to improve?

1- Using ontologies. In this first approach, the new words in the test files (missing in the train file) are considered not hateful. Instead, we can associate a probability of "being hateful" and improve this probability using ontologies.
2- Adding other statistics on words. In the "symbolic" example, we describe the words by only two statistics (support, conditional probability of being hateful). We can add TF-IDF, statistics on combination of two or more words, etc...
3- Explanatory analysis of the symbolic data table :

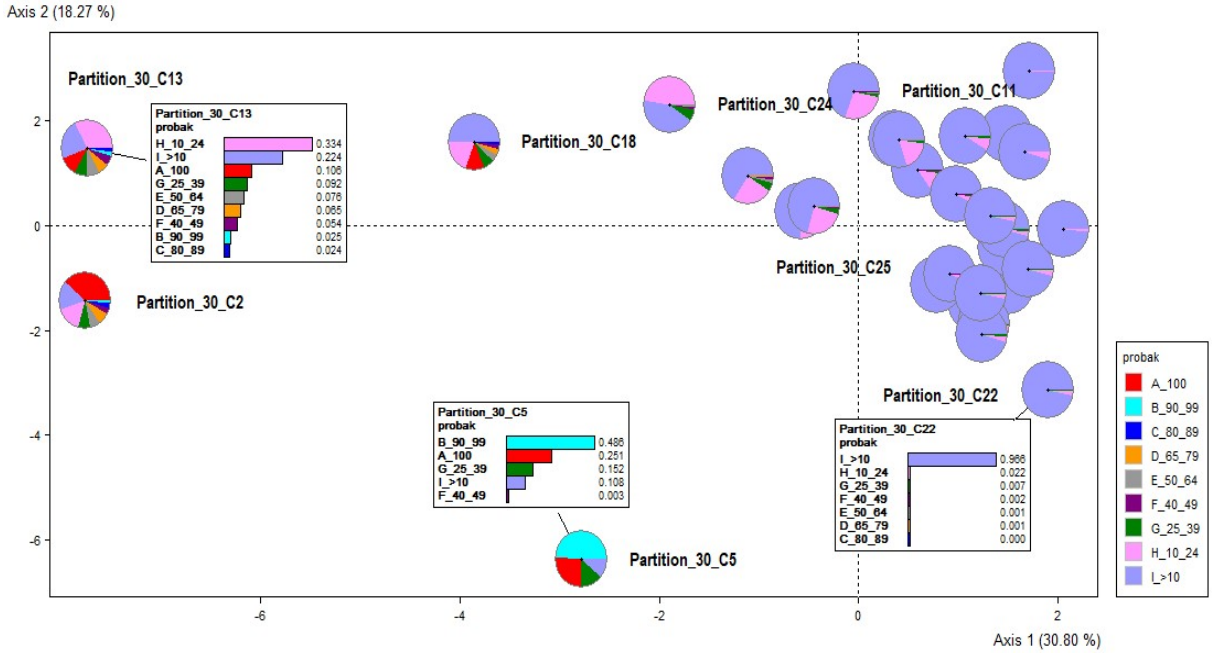
For example, we use Principal Component Analysis (PCA) extended to symbolic data of SYR software.

We apply the « symbolic » PCA to a symbolic data table of 30 clusters of tweets (using only the variables type, supk and probak, see table 3). In the following figure, we vizualise the 30 clusters in the first two principal components. We can display the variable « hate » and see that we have all the not hateful (« hate=0 ») clusters concentrated at the upper right of the graphic. The hateful clusters are discriminated at the left of the first axis.

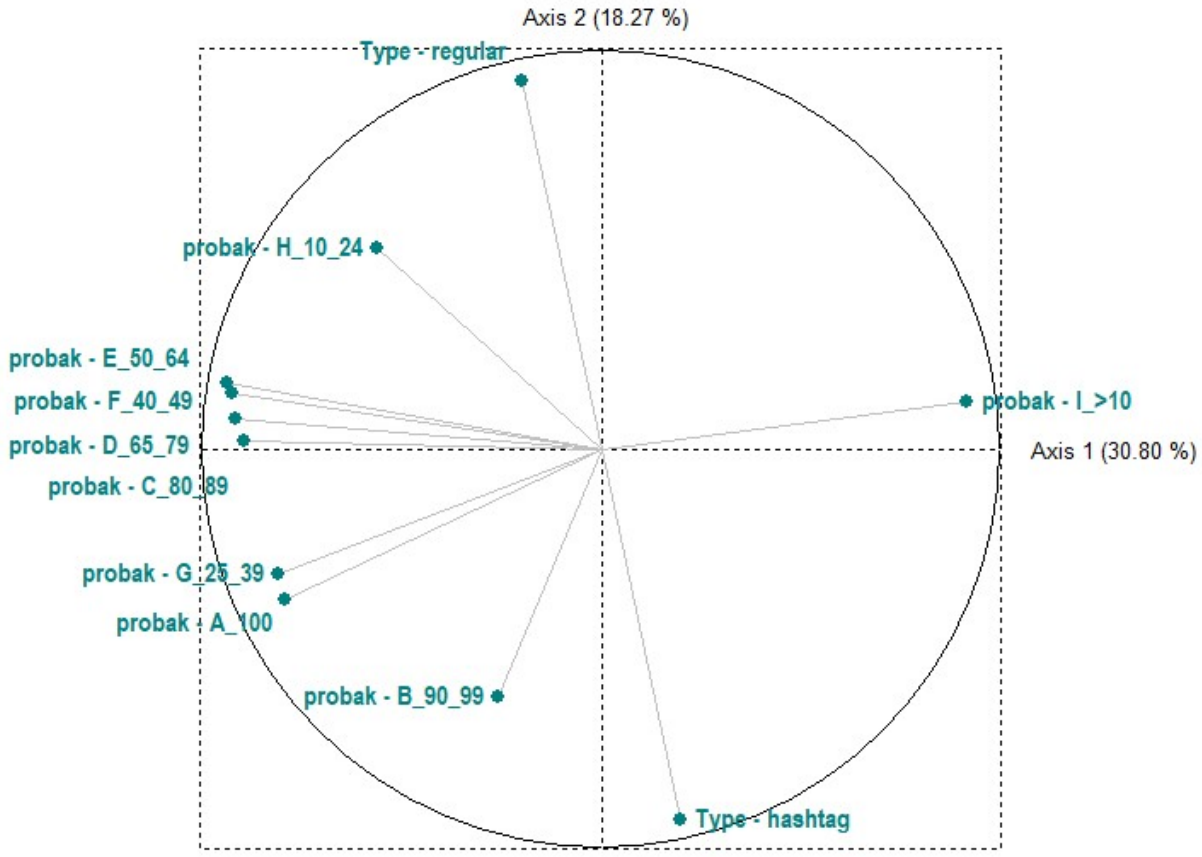


Bar-chart values can also be visualized as pie-charts. A click in a pie-chart displays the frequencies. In the following figure, the variable « probak » is displayed. At the right of the first axis, clusters are mostly composed of words with « probak<10% ». At the left, we see

the hateful clusters with clearly different descriptions.

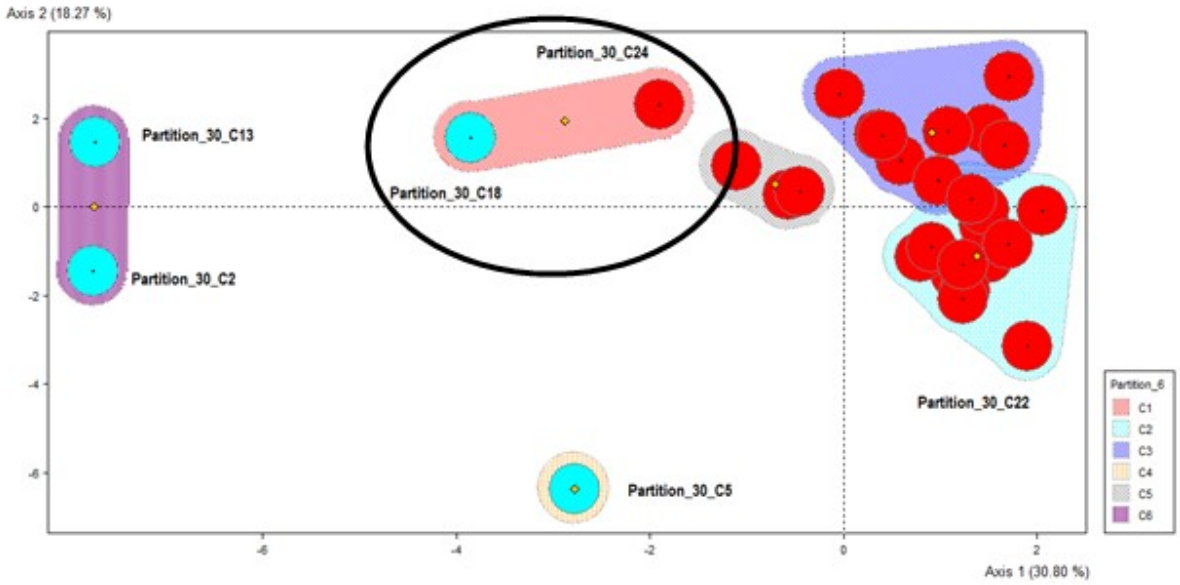


Of course, these results can be synthesized by visualizing the correlation circle.



Finally, we can display a k-means clustering (classical k-means on the coordinates of the clusters in the axis, or symbolic clustering on initial symbolic data table describing the clusters table 3). In the following figure, a clustering into 6 clusters is displayed. We see one

cluster gouping hateful tweets and not hatefull tweets. An additional analysis of this cluster could help to improve the results?



References

Afonso, F., Diday, E., Toque, C. (janv. 2018). Data Science par Analyse des Données Symboliques. Technip. 448 pages, ISBN : 9782710811817.

Afonso, F., Diday, E., (2018). User Manual of the SYR Software. Symbad internal publication.www.symbad.co, 80 pages.

Billard, L. (2011). Statistical Analysis and Data Mining. The ASA Data Science Journal. © Wiley Periodicals, Inc. Cover image for vol. 4 Issue 2. Version of Record online: 8 MAR 2011 | DOI: 10.1002/sam.10115

Billard, L., Diday, E. (2006). Symbolic Data Analysis: conceptual statistics and data Mining. Book. Wiley. ISBN 0-470-09016-2.

Bock, H.-H., Diday, E. (2000). Analysis of Symbolic Data. Exploratory methods for extracting statistical information from complex data. Springer Verlag, Heidelberg, 425 pages, ISBN 3-540-66619-2.

Brito, P., Noirhomme-Fraiture, M., Trejos, J. (2015). Special issue on Symbolic Data Analysis. ADAC. vol. 9, N° 1.

Diday, E. (2016). Thinking by classes in Data Science: the symbolic data analysis paradigm. WIREs Comput Stat 2016, p. 8:172–205. Doi: 10.1002/wics.1384.

GUAN Rong, LECHEVALLIER Yves, SAPORTA Gilbert et WANG Huiwen eds (2013) : Advances in Theory and Applications of High Dimensional and Symbolic Data Analysis

Noirhomme-Fraiture, M., Brito, P. (2012). Far beyond the classical data models: symbolic data analysis. Stat. Anal. Data Mining 2012, 4, p. 157–170.

Noirhomme-Fraiture, M., Diday, E. editors (2008). Symbolic Data Analysis and the SODAS Software. Wiley–Blackwell. ISBN 9780470018835.

Su S-F., Pedrycz, W., Hong T-P., De Carvalho, A.T. (2016). Special Issue on Granular/Symbolic Data Processing. IEEE Transactions on Cybernetics