Clustering and Cluster Interpretability Analysis With Decision Trees On the Grand Débat Data - Group 19

Xin ZHANG[1] Yixiao FEI[2] Alaeddine SABBAGH[3] Fedi KOUKI[4]

Abstract— **In this project, answers from participants would be collected and analyzed using data mining approaches. A matrix of TF-IDF features is generated after stemming and the definition of stop words by vectoring the frequencies of words in the responses of an individual. By learning the key word patterns, citizens answering the questions of the "Grand Débat" would be at first clustered into several categories without knowing what are the categories. These categories defined by the K-means algorithm would be then interpreted by looking through the words clouds. After iterating these two steps by adding new stop words as filtering criterion, clear clusters with rigorous interpretability would be defined and these categories would be regarded as the right labels for classification with several classification algorithms. With trained model for the supervised machine learning, the prediction of the categories for the new introduced participants would be possible. In the project, tools of the visualizations of data are realized for interpretation.**

I. INTRODUCTION

In France, President Emmanuel Macron launched the "Great National Debate" in January in response to the nationwide protests over his government's attitude towards purchasing power, taxation and the cost of living. Over two and a half months, the initiative gathered some 1.5 million proposals via 10 000 town hall meetings, 16 000 books where citizens wrote their grievances and an online platform. [11] In addition to taxes, debates included questions on the environment, public services, democracy, health, pensions and the welfare state. The debate got off to a difficult start, when the main organizer stepped down due to an outcry over her salary. But the tone around the debates improved and even Macron's approval ratings have grown, with recent polls showing around 30 percent of voters with a positive view of the president. Meanwhile, attendance at Gilets Jaunes protests has been declining in recent weeks.

Even though the usefulness the the national debate has been doubted since the beginning[1], this event has gotten a great nationwide participation and a considerable amount of data is collected through the web page GrandDebat.fr, and it is believed to have a huge value of research. M. Chevallier[2] believes that through proper methods of investigation, the data can provide valuable references to current political, economical and environmental problems. Some analysis based on the geographical position and the social stratum of the participants has already been made, for example, H. Bennani et al.[3] found that the educational background is the main factor for local percentage of participation online by analyzing those data.

However, three months after the outset of the debate, few articles focused on the text itself of the responses have been published. In this project, we will focus on clustering the responses of every individual in the topic of democracy and citizenship and trying to interpret the result with visualization approaches and decision tree generation. Several representative clusters have been generated with high prediction performances after corresponding classification, and with those classifiers, we can accurately tell the political inclination of an individual from his responses to this topic.

II. PREPROCESSING OF THE DATA

A. Data extraction

The data file chosen to perform the operations is DEMOCRATIE_ET_CITOYENNETE, in this file, we have observed that the ways that the questions were replied vary enormously from person to person. Firstly, few participants have responded all the questions proposed, many of them only responded one or two of the questions. Secondly, even for the same question, there are respondents who wrote merely several words but there are also the ones who have written a comprehensive article to support his ideas. Additionally, even for the samples with most of the questions responded, it can be observed that they usually have a specific preference of questions and their responses are not balanced for each question.

Based on those observations, we believed that a question-based feature representaion will be meaningless, so we have decided to combine all the answers of each sample to extract the text. In this procedure, unanswered questions are ignored and every respondent is used as a sample to process in order to interpret his view of this topic.

B. Stop Words

As stressed by C. Silva et al.[4], the stop word removal is indispensible for text categorization, so we have started the prepreocessing with determining the stop words.

In addition to the basic French stop words provided by the Python package, we have firstly added all the punctuation symbols, but during the process, it is found that there are other unexpected symbols as the data is derived for the web and there is no standardizaiton, so those symbols are also managed to be eliminated.

Moreover, in view of this particular case, several words are bound to be mentioned and do not qualify as a feature to determine the category it belongs to. By the first few

[1] X. Zhang, SD
[2] Y. Fei, SD
[3] A. Sabbagh, SD
[4] F. Kouki, TSIA

experiments of clustering, we have found that words such as “citoyen” and “politique” have considerable weights in all the clusters thus they have affected the representativeness of each group, so they are equally decided to be added in the list of stop words.

C. Tokenization and Stemming

In order to calculate the frequencies of the words, we need to tokenize all the words in a proper way[5]. But to achieve this goals, there are some challenges.

Firstly, French is a fusional language[6], so stemming is required to reduce inflected words to their word stem. So, the Snowball[7] method is used to retrieve the root form of words. Secondly, the elision[8] is a highly common phenomenon in French and various words can become a prefix to affect the extraction of the base word, so in the code of tokenization, those situations are also considered in order to generate purely rooted forms of words.

D. Vectorization

As indicated by W. Zhang et al.[9], TF-IDF possesses better statistical quality than other common vectorization methods like multi-words, so TF-IDF was chosen as the method for vectorization.

By the definition, the tf (text frequency) is defined as

$$\mathrm{t_{i,j}} = \frac{n_{i,j}}{\sum_k n_{k_i j}} \tag{1}$$

where $n_{i,j}$ is the number of appearances of a word n in the responses of a participant j. And the idf (inverse document frequency) is defined as

$$\mathrm{idf_i} = \lg \frac{|D|}{1 + |\{j : t_i \in d_j\}|} \tag{2}$$

where D is total number of respondents and $|\{j : t_i \in d_j\}|$ stands for the number of users who have used this word in their responses. Therefore, the elements in the matrix is obtained from

$$\mathrm{tfidf}_{i,j} = \mathrm{tf_{i,j}} \times \mathrm{idf_i} \tag{3}$$

In using the TF-IDF method, it is supposed that the most important factors for differentiation are the words that appear frequently in certain samples but rarely in others, which is logical in our case as most the respondents can be considered irrelevant and less rational thus certain words can only be used if they have a certain inclination.

As the responses do not have a great length, so after experiments and deliberating, 100 features / words are used for the following training procedure in order to boost the training speed and prevent overfitting.

E. Storage

The answers are transformed to key words’ frequency after stemming by a sparse numpy array because it’s essentially a matrix of zeros, with a handful of nonzero elements per row. The sparse matrix format is more efficient storage wise.

Since the preparation of stemming takes much time, it is convenient to save the sparse matrix so it won’t have to be recalculated. Functions (from the ‘scipy‘ user group) make it easy to save and load the matrix.

III. Clustering and Visualization

In this section, a K-means clustering algorithm is run on the preprocessed data and several approaches has been applied to better understand the clusters obtained. As a result, we have labeled the obtained clusters as indicated in the table below.

TABLE I
Labels of each cluster

Cluster	0	1	2	3
Label	NOTA	Localism	Bureaucracy	Conservatism
Cluster	4	5	6	7
Lable	Nationalism	Education	Law	Activism

The labels are decided by the typical words used in each group after we have read some random chosen answers in each group. The labels represent either their role in this national movement, or the main concern when it comes to citizenship and democracy, and the NOTA here means “None of the above”, which stands for the people who don’t vote.

The representativeness and effectiveness of those clusters will be discussed in the following subsections.

A. K-means Clustering

As the dataset is enormous, a K-means clustering will be an ideal option. In the K-Means algorithm, the goal of optimization is to minimize the WCSS(within-cluster sum of squares) in each group, which is

$$\arg\min_{\mathbf{S}} \sum_{i=1}^{k} \sum_{\mathbf{x} \in S_i} \|\mathbf{x} - \boldsymbol{\mu}_i\|^2 \tag{4}$$

where $\mathbf{S}_i$ represents a cluster, k is the number of clusters, and μ_i is the mean value of all points in $\mathbf{S}_i$.

In this project, the distances are represented by the frequencies of the words chosen by the TF-IDF vectorizer, and the repeated appearance of the words are relatively rare, so it can be seen that chosen words are highly characteristical.

B. Word Cloud and Interpretation

In order to interpret the clusters we obtained in the precedent subsections, Word Cloud is chosen to deliver a straightforward and visually appealing view of the result.

We have utilized the word frequencies extracted from the centroids of each cluster as the base of generation, and the word cloud generated are shown below in Fig. 1.

From the word clouds, it can be seen that every group is determined by several varied decisive words, and other words are barely noticable thus can be kind of ignored.

If we take a deeper look into the clouds, it can be surprisingly observed that by combining the most frequent words, a meaningful opinion can be constructed. For instance, in the first cluster, we can see the words “vote”, “blanc”, “élection”, “maire” and “obligatoire”, which can be easily interpreted as “Dans les élections, par exemple pour le maire, il y a beaucoup de votes blancs, et on a besoin de certainne obligation.” (In elections, for example for a

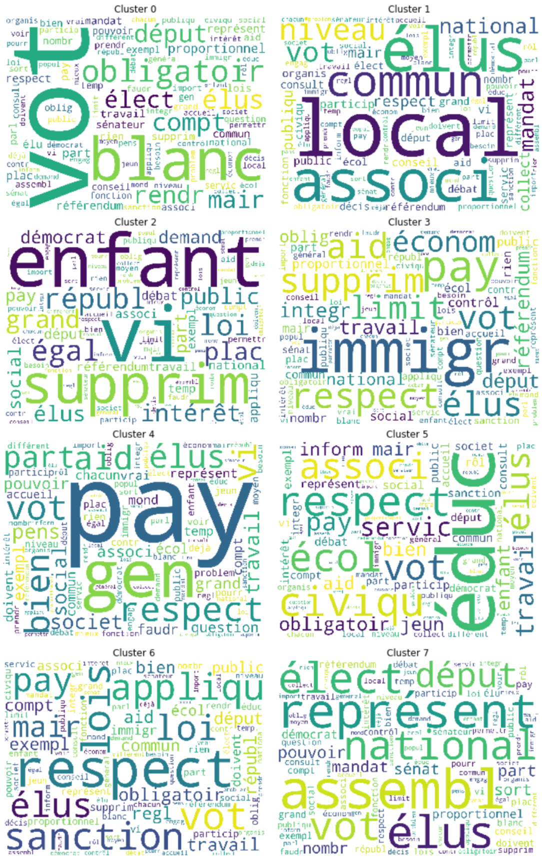

Fig. 1. Word Clouds of the Clusters

mayor, there are too many votes for none of above, which requires obligations). So, it is clearly indicated that the main concern of this participant is the "NOTA(None of the above)" phenomenon happened in elections. Similar insights can also be interpreted from other clusters, which has led us to the summarized labels in table I.

C. Distribution and Cluster Assignments

In order to further illustrate the validity of the clustering results, the histogram of the number of samples is generated in each group by randomly selected samples, as shown in Fig. 2.

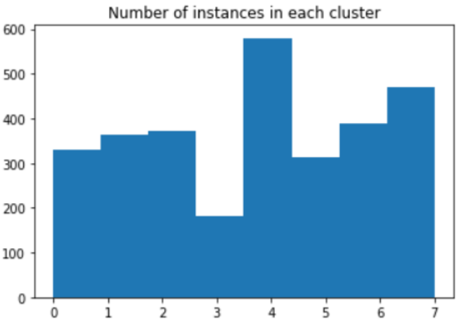

Fig. 2. Distribution of clusters

From this figure, it can be seen that even there is a certain inequality between groups, but this inequality has conversely proven that the clustering result has a high potential to be the best division as it is not simply divided in groups of the same size. Combined with our real-life knowledge, it is obvious that the forth group, which talks about the immigrants and limitations will have lesser samples as political correctness are often considered as common sense nowadays when people talk on the Internet, which will lead to a fewer quantities of members in this group.

D. Dimension Reduction and t-SNE Visualization

To extract the information contained in the high dimensional space down to an array in R^2, for which a scatter plot could visualized easily. One way is the Principal Components Analysis. Another way is t-SNE, or t-distributed Stochastic Neighbor Embedding, which is an extension of SNE.

The technique is a variation of Stochastic Neighbor Embedding[10] that is much easier to optimize, and produces significantly better visualizations by reducing the tendency to crowd points together in the center of the map. t-SNE is better than existing techniques at creating a single map that reveals structure at many different scales. This is particularly important for high-dimensional data that lie on several different, but related, low-dimensional manifolds, such as images of objects from multiple classes seen from multiple viewpoints. For visualizing the structure of huge data sets, we showed how t-SNE can use random walks on neighborhood graphs to allow the implicit structure of all of the data to influence the way in which a subset of the data is displayed. We illustrated the performance of t-SNE on a wide variety of datasets and compare it with many other non-parametric visualization techniques, including Sammon mapping, Isomap, and Locally Linear Embedding. The visualizations produced by t-SNE are significantly better than those produced by the other techniques on almost all of the data sets.

t-SNE aims to learn a d-dimensional map $\mathbf{y}_1, \ldots, \mathbf{y}_N$ (with $\mathbf{y}_i \in \mathbb{R}^d$) that reflects the similarities p_{ij} as well as possible. To this end, it measures similarities q_{ij} between two points in the map $\mathbf{y}_i$ and $\mathbf{y}_j$, using a very similar approach. Specifically, q_{ij} is defined as:

$$q_{ij} = \frac{(1 + \|\mathbf{y}_i - \mathbf{y}_j\|^2)^{-1}}{\sum_{k \neq l}(1 + \|\mathbf{y}_k - \mathbf{y}_l\|^2)^{-1}} \tag{5}$$

Herein a heavy-tailed Student-t distribution (with one-degree of freedom, which is the same as a Cauchy distribution) is used to measure similarities between low-dimensional points in order to allow dissimilar objects to be modeled far apart in the map.

From the illustrated Fig. 3., we can find out that most of the clusters are well divided, especially cluster 0, cluster 2, cluster 3 and cluster 7. It's also interesting to see that the most mixed groups, cluster 2 and cluster 4, which correspond to localism and nationalism, do share extremely similar ideas in real-life, which indicates that we have obtained a potential highly representitive result.

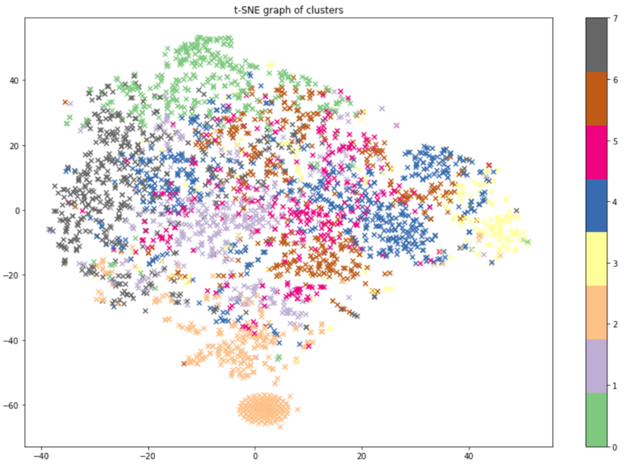

Fig. 3. t-SNE two dimensional graph of the clusters

IV. Decision Trees and Performance

In this section, in order to test the interpretability of the results, several decision tree algorithms are applied by using the clustering result as supervised classes. A high performance prediction model has been trained by ensemble methods like random forest and extra decision trees, and a sample branch of a sample decision tree is illustrated and discussed.

A. Training and Prediction

In this training and prediction process, we have divided the data into learning and testing groups. Then, we have applied the Extra Tree algorithm and Random Trees algorithm and obtained their prediction metrics as shown in table II and table III.

In those two tables, Precision (P) is defined as the number of true positives (T_p) over the number of true positives plus the number of false positives (F_p).

$$P = \frac{T_p}{T_p + F_p} \tag{6}$$

Recall (R) is defined as the number of true positives (T_p) over the number of true positives plus the number of false negatives (F_n).

$$R = \frac{T_p}{T_p + F_n} \tag{7}$$

the (F_1) score is defined as the harmonic mean of precision and recall

$$F1 = 2\frac{P \times R}{P + R} \tag{8}$$

From these tables, it can be observed that through the overall precision is not high enough to be ideal, a satisfying precision has been seen for certain classes.

B. Decision Tree Visualization

To the purpose of getting an inner look of the decision tree, we have managed to illustrate a sample decision tree with the graphviz package, as a 8-classes decision tree is relatively large and wide, we have include the image in the .zip file and only a sample branch is shown below in Fig. 4.

TABLE II
Prediction Metrics of Extra Trees

Cluster	Precision	Recall	f1-Score
0	0.86	0.74	0.80
1	0.79	0.81	0.80
2	0.85	0.95	0.90
3	0.94	0.43	0.59
4	0.65	0.87	0.74
5	0.85	0.68	0.76
6	0.78	0.64	0.70
7	0.72	0.75	0.74
Micro Average	0.77	0.77	0.77
Macro Average	0.80	0.73	0.75
Weighted Averge	0.78	0.77	0.76

TABLE III
Prediction Metrics of Random Forests

Cluster	Precision	Recall	f1-Score
0	0.86	0.73	0.79
1	0.66	0.75	0.70
2	0.86	0.96	0.91
3	0.88	0.80	0.84
4	0.80	0.82	0.80
5	0.82	0.74	0.78
6	0.76	0.69	0.72
7	0.74	0.80	0.77
Micro Average	0.79	0.79	0.79
Macro Average	0.80	0.78	0.79
Weighted Averge	0.79	0.79	0.79

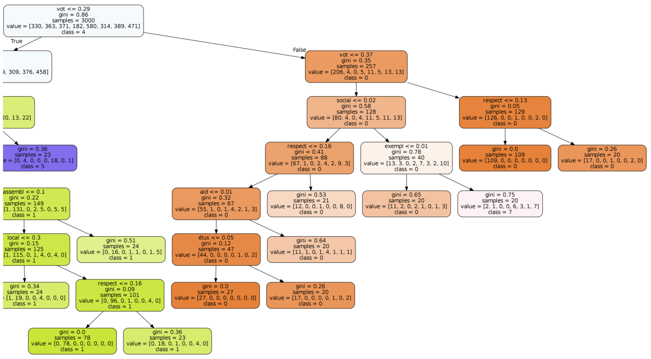
vot <= 0.29 gini = 0.86 samples = 3000 value = [330, 363, 371, 182, 580, 314, 389, 471] class = 4
True
False
9, 309, 376, 458]
vot <= 0.37 gini = 0.35 samples = 257 value = [206, 4, 0, 5, 11, 5, 13, 13] class = 0
20, 13, 22]
social <= 0.02 gini = 0.58 samples = 128 value = [80, 4, 0, 4, 11, 5, 11, 13] class = 0
respect <= 0.13 gini = 0.05 samples = 129 value = [126, 0, 0, 1, 0, 0, 2, 0] class = 0
gini = 0.36 samples = 23 e = [0, 4, 0, 0, 0, 18, 0, 1] class = 5
respect <= 0.16 gini = 0.41 samples = 88 value = [67, 1, 0, 2, 4, 2, 9, 3] class = 0
exempl <= 0.01 gini = 0.78 samples = 40 value = [13, 3, 0, 2, 7, 3, 2, 10] class = 0
gini = 0.0 samples = 109 value = [109, 0, 0, 0, 0, 0, 0, 0] class = 0
gini = 0.26 samples = 20 value = [17, 0, 0, 1, 0, 0, 2, 0] class = 0
assembl <= 0.1 gini = 0.22 samples = 149 [1, 131, 0, 2, 5, 0, 5, 5] class = 1
aid <= 0.01 gini = 0.32 samples = 67 value = [55, 1, 0, 1, 4, 2, 1, 3] class = 0
gini = 0.53 samples = 21 value = [12, 0, 0, 1, 0, 0, 8, 0] class = 0
gini = 0.65 samples = 20 value = [11, 2, 0, 2, 1, 0, 1, 3] class = 0
gini = 0.75 samples = 20 value = [2, 1, 0, 0, 6, 3, 1, 7] class = 7
local <= 0.3 gini = 0.15 samples = 125 [1, 115, 0, 1, 4, 0, 4, 0] class = 1
gini = 0.51 samples = 24 value = [0, 16, 0, 1, 1, 0, 1, 5] class = 1
élus <= 0.05 gini = 0.12 samples = 47 value = [44, 0, 0, 0, 0, 1, 0, 2] class = 0
gini = 0.64 samples = 20 value = [11, 1, 0, 1, 4, 1, 1, 1] class = 0
gini = 0.34 samples = 24 [1, 19, 0, 0, 4, 0, 0, 0] class = 1
respect <= 0.16 gini = 0.09 samples = 101 value = [0, 96, 0, 1, 0, 0, 4, 0] class = 1
gini = 0.0 samples = 27 value = [27, 0, 0, 0, 0, 0, 0, 0] class = 0
gini = 0.26 samples = 20 value = [17, 0, 0, 0, 0, 1, 0, 2] class = 0
gini = 0.0 samples = 78 value = [0, 78, 0, 0, 0, 0, 0, 0] class = 1
gini = 0.36 samples = 23 value = [0, 18, 0, 1, 0, 0, 4, 0] class = 1

Fig. 4. Sample Branch of the Sample Decision Tree

From the tree, we can see that the root node is decided by the word “vote”, and it has naturally lead to the first class, which has proven our initial guess of labels.

V. Conclusions

After interpreting the results, we have found that there are several meaningful main concerns related to the topic of citizenship and democracy during Le Grand Débat, and people that have participated in the discussion can be divided in several groups, as discussed in section II.A..

For further investigations, we think we can try other vectorization methods like LSI, and there is a potential to do several clustering after firstly divided them by social status or geological positions.

References

[1] Legris, M. (2019). À quoi sert un débat en temps de crise?. Etudes, (3), 31-42.

[2] Chevallier, M. (2019). Les non-dits du grand débat. Alternatives Economiques, (2), 10-10.

[3] Bennani, H., Gandré, P., & Monnery, B. (2019). Les déterminants locaux de la participation numérique au Grand débat national: une analyse économétrique (No. 2019-7). University of Paris Nanterre, EconomiX.

[4] Silva, C., & Ribeiro, B. (2003, July). The importance of stop word removal on recall values in text categorization. In Proceedings of the International Joint Conference on Neural Networks, 2003. (Vol. 3, pp. 1661-1666). IEEE.

[5] Uysal, A. K., & Gunal, S. (2014). The impact of preprocessing on text classification. Information Processing & Management, 50(1), 104-112.

[6] Wikipedia contributors. (2019, March 14). Fusional language. In Wikipedia, The Free Encyclopedia. Retrieved April 2019

[7] Porter, M. F. (2001). Snowball: A language for stemming algorithms.

[8] Wikipedia contributors. (2019, February 13). Elision. In Wikipedia, The Free Encyclopedia. Retrieved April 2019

[9] Zhang, W., Yoshida, T., & Tang, X. (2011). A comparative study of TF-IDF, LSI and multi-words for text classification. Expert Systems with Applications, 38(3), 2758-2765.

[10] Maaten, L. V. D., & Hinton, G. (2008). Visualizing data using t-SNE. Journal of machine learning research, 9(Nov), 2579-2605.

[11] RFI, France awaits results of Macron's Great Debate, 08-04-2019