2nd PROJECT PML

Ghadamiyan Lida, Class 407 AI 15.01.2020

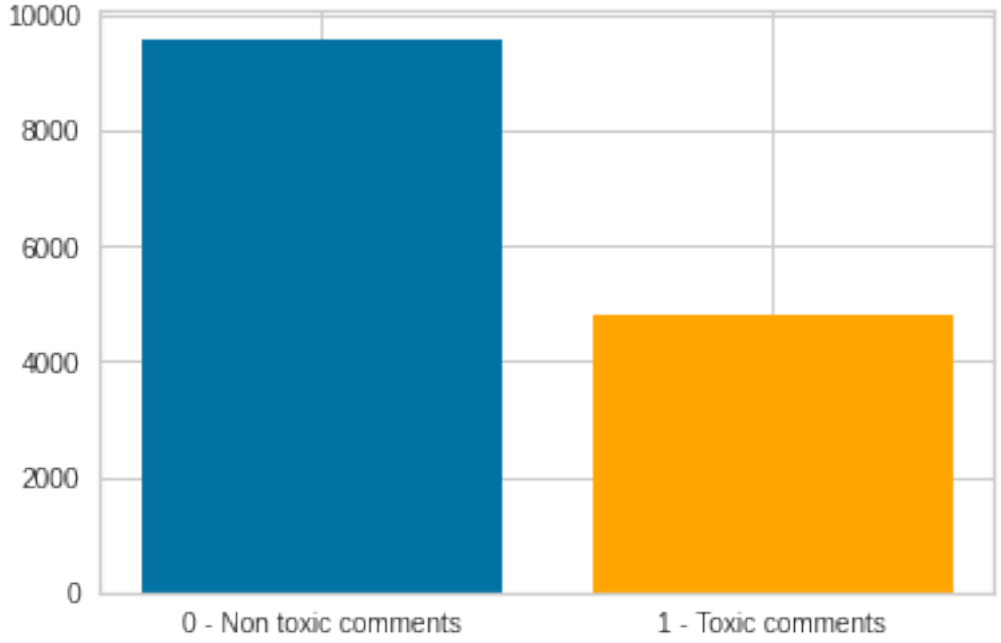
Toxic Russian Comments Classification

The main purpose of this project is to classify text in two categories, depending on their toxicity.

Automating the process of detecting negative comments is very important to keep the internet a safe environment for everyone. Such comments might start different kind of conflicts that can become a reality, or can negatively affect someone. According to Hinduja and Patchin, 25 percent of students experience cyberbullying, and the majority of them were highly affected. Thus, the importance of predicting the toxicity of on-line content is very important.

1 Data analysis and preprocessing

The data set consists in 14412 comments in Russian language, with an average value of 176 characters and their labels. The label 0 is for non-toxic comments and 1 for toxic. The distribution of the labels can be seen in the table below.



The text is in the original form, presenting uppercase letters, punctuation, numbers and '\n' at the end of each line. In order to bring the text to a cleaner point I used the nltk library for tokenizing, lemming and stemming. The punctuation was removed using string.punctuation, which is a predefined

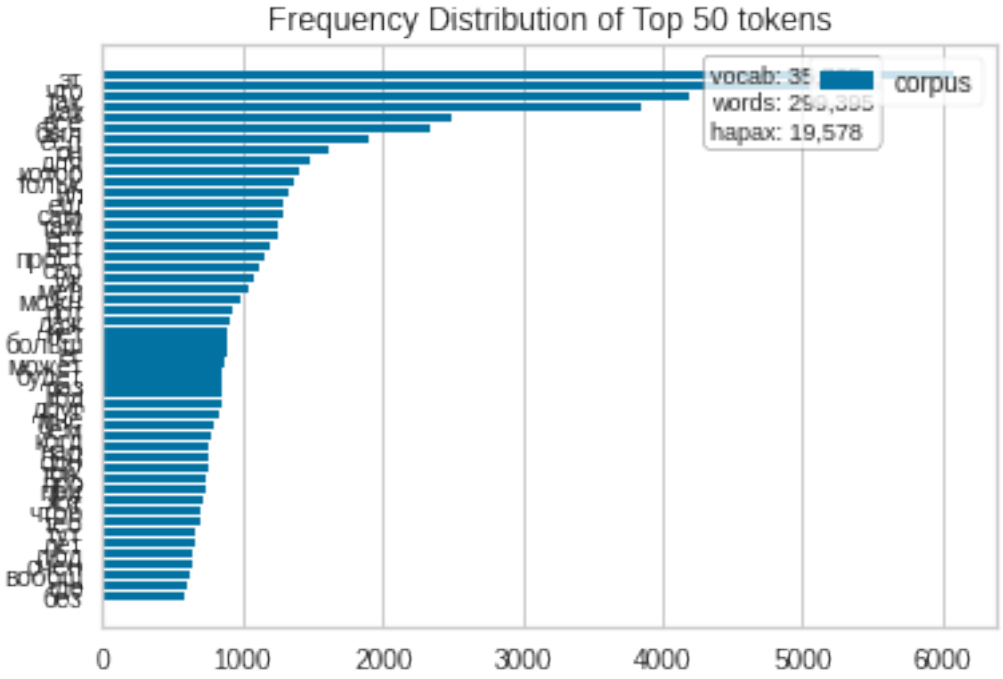
string containing punctuation symbols. After that, I removed the least two characters of every sentence, which were '\n'. As I do not know the Russian language, I could not do much about words removal, so I removed only words shorter than 3 letters, and numeric characters. After these steps, the tokens were put together in sentences, and the sentences were organized in a pandas data frame.

For the text tokenization I used CountVectorizer which returns for each sentence a vector where each word is described by its occurrence in all sentences. Therefore, the result for all the data is a matrix, whit a row for every sentence and a column for every word in the dictionary. The dictionary consists in a set in which every element have a key and a value, namely a word (only form the training data) and its occurrence. After this, I used TfidfTransormer to compute the frequency of each word instead of the occurrence.

In the figure below is shown the data before and after preprocessing.

	comments	preprocessed comments
0	Верблюдов-то за что? Дебилы, бл...\n	[верблюдовт, что, дебил]
1	Хохлы, это отдушина затюканого россиянина, мол...	[хохл, эт, отдушин, затюкан, россиянин, мол, в...
2	Собаке - собачья смерть\n	[собак, собач, смер]
3	Страницу обнови, дебил. Это тоже не оскорблени...	[страниц, обнов, деб, эт, тож, оскорблен, дока...
4	тебя не убедил 6-страничный пдф в том, что Скр...	[теб, убед, 6страничн, пдф, том, что, скрипал,...

The distribution of 50 most common words, after the comments were preprocessed is plotted below.



2 Supervised Method - Multinomial Naive Bayes

For the supervised method I chose Multinomial Naive Bayes. This model is based on Bayes theorem regarding conditional probabilities, and the Naive assumption that the features are independent. Considering a NLP problem, the features are not independent, as the words are not randomly placed in the sentences.

The Bayes formula applied for predicting the class, knowing the features is the following:

$$P(C_j|X) = \frac{P(X|C_J)P(C_J)}{P(X)}$$

where C_j is the class, $j \in 1, .., N$ the possible outcomes and $X = (x_1, .., x_n)$ the feature vector

The classifier obtained by developing the Bayes formula is assigning each feature vector the highest probability of the vector, knowing the class.

$$PredictedLabel = argmax_{j \in 1,..,N} P(C_j) \prod_{1<i<n} P(x_i|C_j)$$

I used MultinomialNB with the alpha = 10^{-1} parameter. The parameter was chosen after running the GridSearchCV with 5 folds.

I used two measures for evaluation, the Accuracy Score and the Fowlkes Mallwows Score, which is the geometric mean between the precision and recall.

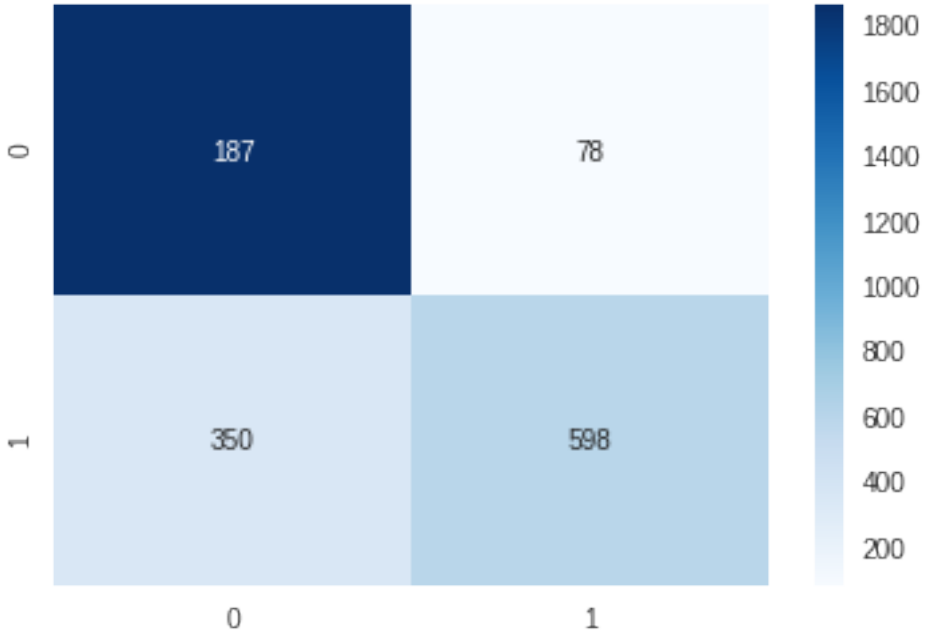
$$Accuracyscore = \frac{TP+TN}{Total} = \frac{Correct}{Total}$$
$$FowlkesMallwowsScore = \frac{TP}{\sqrt{(TP+FP)(TP+FN}}$$

The results for this model were:

	Precision	Recall	F1 score	Accuracy	FMS
NB model	0.86	0.80	0.82	0.85154	0.79098
label 0	0.84	0.96	0.90		
label 1	0.88	0.63	0.74		

As we can observe, the recall for the 0 label is bigger than the precision, and this means that the model assigns more 0 labels (most of them could be wrong). The overall results show that the model behaves well.

The confusion matrix:



It can easily be seen that the model assigns more 0 labels, the majority of them being actually wrong, as the recall suggested.

3 Unsupervised Methods

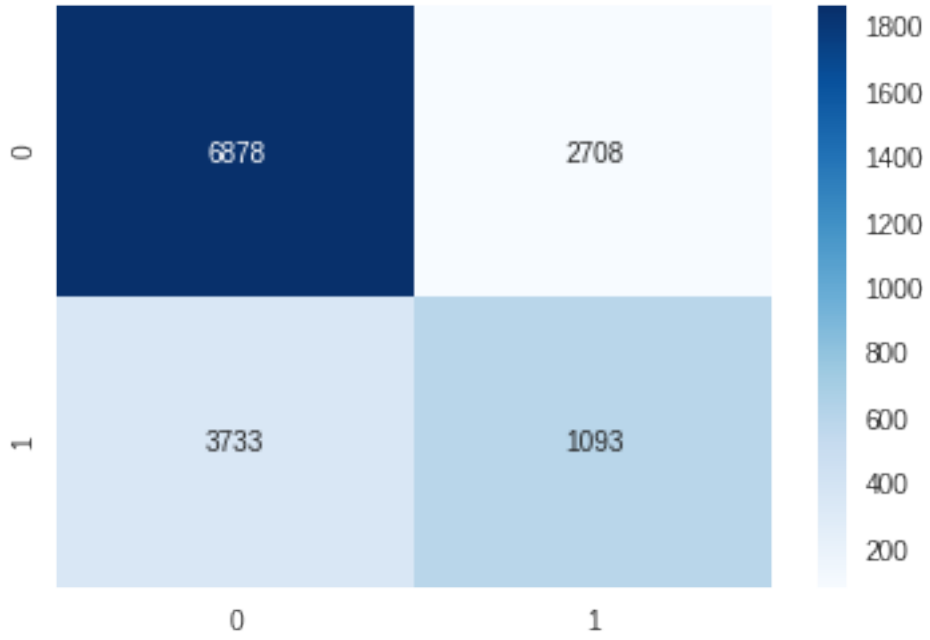
For the unsupervised method I chose K-means and DBSCAN. I used the labels for computing the scores and all the data for training and predicting.

3.1 K-Means

The K-means algorithm works by assigning K initial centroids and calculating the distance between each point to each centroid. It assigns each point the label of the nearest cluster and then compute the mean of the cluster and use it as a new centroid. Then, it repeats this until the cluster does not change anymore.

The results for this methods were:

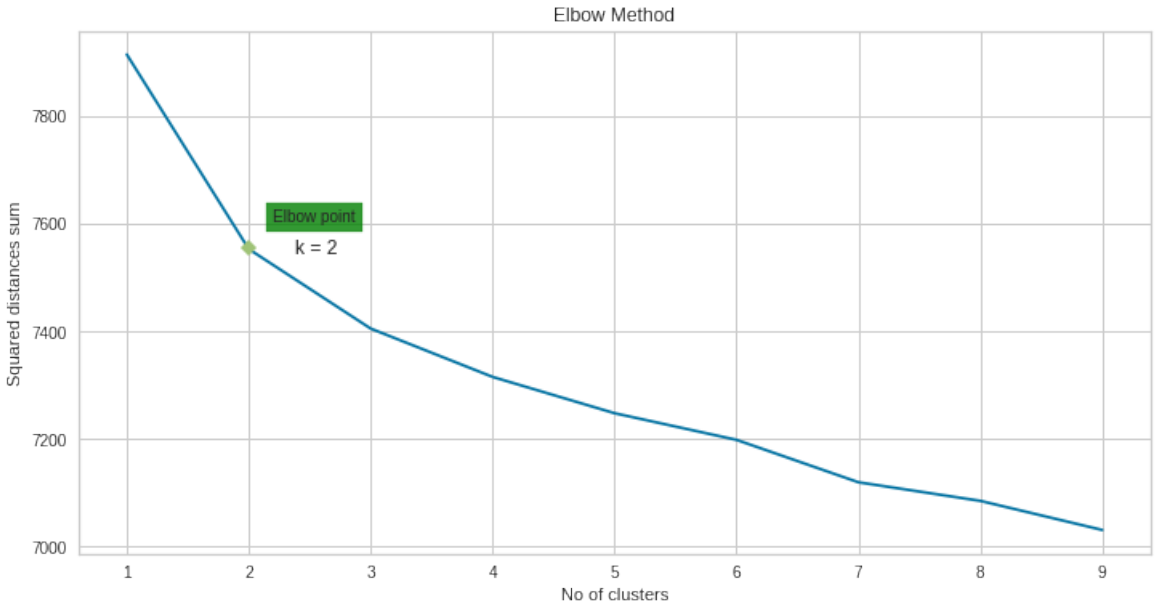
	Precision	Recall	F1 score	Accuracy	FMS
K-Means	0.47	0.47	0.47	0.5530	0.5767
label 0	0.65	0.72	0.68		
label 1	0.29	0.23	0.25		



From the results we can conclude that the most of the labels are predicted good, but, as in the case of Naive Bayes, the most of the predictions were 0.

Elbow Point

The elbow method computes the K-means algorithm for different values and calculates the sum of the squared distances from a randomly selected centroid to all the points. The plot of the sums will be decreasing, as the number of centroids increase. Then, the inflexion point, which is the point in which the curvature changes its sign, will be the optimal k.



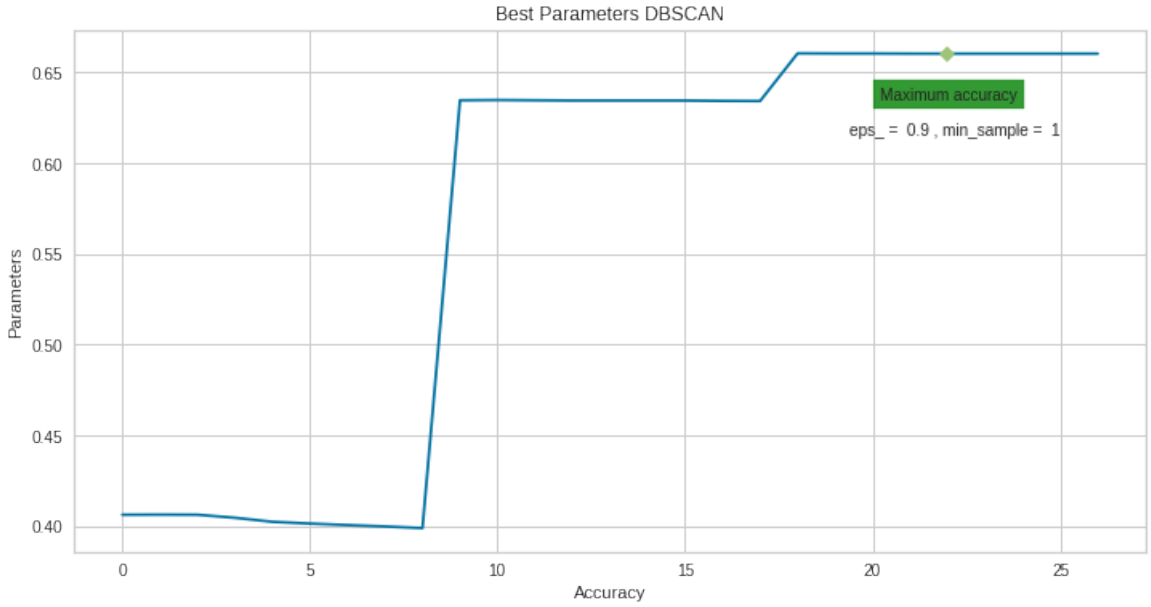
3.2 DBSCAN

This algorithm consists in randomly choosing one point that does not belong to any cluster and then selecting all the points that are in its range (eps). Each of the selected points does the same thing until there is no point nearer than eps to the newly created cluster. After a new cluster is created, the steps are repeated.

For this method I used TruncatedSVD which uses just the first n largest singular values and set the others to zero. The reason because I used this is that TSVD does not center the data, so it works with sparse matrices, unlike PCA.

I chose the eps and min samples parameters by testing the accuracy using some values and selecting the parameters for which I have the most suitable number of clusters (2 or 3).

The accuracy plot and the best score can be seen in the following plot.



For the parameter selection I kept only 3 values for epsilon for the last project, because the table was very large. The numerical results and the parameters with the most increased accuracy, with the minimum number of clusters different than 1 stands in the following table, showing that

$$Selection\ Results: \begin{cases} Best\ score: 0.6602830974188176 \\ Bes\ tParameters: eps = 0.9\ and\ min\_sample = 2 \end{cases}$$

Clusters	Accuracy	Epsilon	Min Samples
4746	0.406	0.6	1
127	0.406	0.6	2
6	0.406	0.6	3
1	0.405	0.6	4
1	0.403	0.6	5
1	0.402	0.6	6
1	0.401	0.6	7
1	0.4	0.6	8
1	0.399	0.6	9
543	0.635	0.8	1
14	0.635	0.8	2
1	0.635	0.8	3
1	0.634	0.8	4
1	0.634	0.8	5
1	0.634	0.8	6
1	0.634	0.8	7
1	0.634	0.8	8
1	0.634	0.8	9
91	0.66	0.9	1
3	0.66	0.9	2
1	0.66	0.9	3
1	0.66	0.9	4
1	0.66	0.9	5
1	0.66	0.9	6
1	0.66	0.9	7
1	0.66	0.9	8
1	0.66	0.9	9

4 Comparison

The comparison between the 3 models is in the table below. It can be seen that the supervised method behaves the best, returning the highest scores, followed by the DBSCAN. In this case, a random assignation of labels would lead to 50 percent accuracy, and all the models passed this "baseline".

Model	Accuracy	FMS
MultinomialNB	0.8515	0.7909
K-Means	0.5530	0.5767
DBSCAN	0.6602	0.7393

5 Conclusion

In conclusion, the Multinomial Naive Bayes model with the alpha parameter set to 10^{-1} model has the best results, as well for Accuracy as for Fowlkes Mallwows Score, followed by DBSCAN. The Naive Bayes and K-means methods return better results for the data preprocessed and tokenized using CountVect and TfidfTransformer, while DBSCAN presented slightly better results using Truncated SVD along with the other techniques.

It was expected for the supervised model to behave the best, but unsupervised methods have their advantages, as, obviously, working without the need of someone to compute the labels first, and no needing for data updates. For example, if we would use the data from several years ago to predict new comments, it may not behave very well, so the training data should always be updated. Thus, it is very important to have good unsupervised methods. In the case of this project, the fact that I did not know the language might be an obstacle, but it was an interesting exercise because I could analyse the text only from the computed statistics, like the histogram presented.

To sum up, I believe that the quality of the results, considering the recall and the precision depends on the action took based on the prediction. For example, if based on these results someone gets banned, we would wish that there would not be many false positive (the text is not violent, but the prediction shows that it is), as it would result in a higher rate of reports that have to be checked again. If the prediction is used, for example, to suggest posts/text on social media, we would only like to select the non toxic ones, with the risk that some of them would have a smaller impact, but it would be the best for the users not to interact with a lot of aggressive content.

Bibliography

For this project I mainly used the laboratories and courses, along with the documentation (mostly to remember the syntax, parameters, model or implementation) page for some of the libraries and functions that I used, like sklearn for MultinomialNB, confusion matrices, scores, etc. The data mentioned in the introduction, regarding the number of the students affected by cyberbullying is from the following paper (page 3, "Cyberbullying by the numbers") " https://cyberbullying.org/Cyberbullying-Identification-Prevention-Response.pdf ".