**REPORT TIPS**

* Concise and clear rather than discursive style
* Parts:
  + Problem statement and hypothesis
  + Description of your data set and how it was obtained
  + Description of any pre-processing steps you took
  + What you learned from exploring the data, including visualizations
  + How you chose which features to use in your analysis
  + Details of your modeling process, including how you selected your models and validated them
  + Your challenges and successes what didn’t work and why
  + Possible extensions or business applications of your project
  + Conclusions and key findings
  + Appendices: data + code
  + References
* Specify the roles and the tasks performed of every member of the group

1. PROBLEM STATEMENT AND HYPOTHESIS

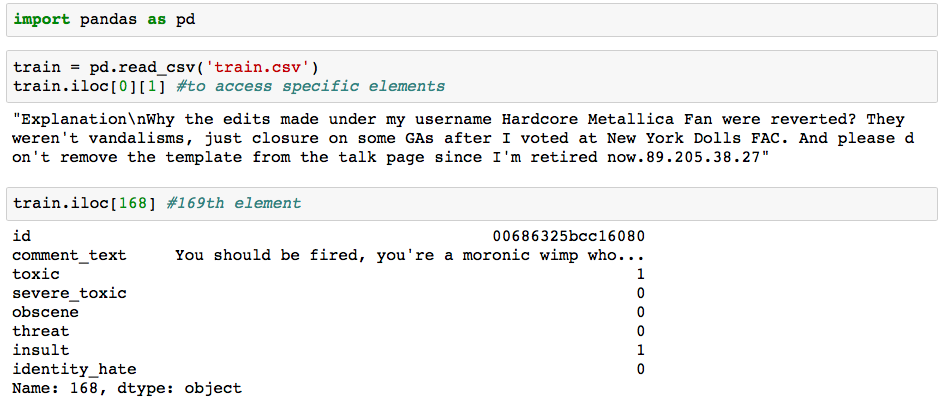
We have decided to take part to a competition launched by the [kaggle](https://www.kaggle.com/) platform, regarding the classification of comments posted online in blogs or discussions according to different levels of offense, in order to address the problem of free expression of personal opinions online and promote civil and effective conversations on the web as well as stop people from leaving online discussions because of lack of respect. The aim of the competition is to study negative online behaviours by looking for a supervised machine learning model able to accurately assign comments to different categories of rudeness or toxicity, such as threats, insults, obscenity.

This is a text classification problem, since it requires to understand textual data, its meaning and its structure and to come up with new insights about that. In this report we will explain the process we have adopted, starting from some exploratory analysis on the dataset, which was taken from Wikipedia’s talk page edits, and the dataset pre-processing to prepare it for the model training, mainly focused on the transformation of text comments into a matrix of numeric vectors representing each of the comments. Once the input for our models is defined, we will show the training and test of the machine learning models we have chosen and their evaluation to understand the performances.

1. DESCRIPTION OF THE DATASET AND HOW IT WAS OBTAINED

The dataset contains 159571 Wikipedia comments labeled by human experts according to six different classes that represent our target variables: “toxic”, “severe\_toxic”, “obscene”, “threat”, “insult”, “identity\_hate”. The model must predict a probability of each type of toxicity for each comment.

Here is a view of the dataset:



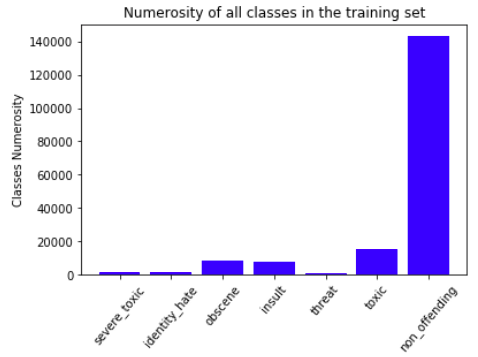
1. EXPLORING THE DATASET

We save the numerosity of each target class in a dictionary data structure and we plot the results. We add a class called “non\_offending” to label all the comments that are not considered offensive, i.e. that have all the six original labels set to 0. This is the resulting dictionary:

{'severe\_toxic': 1595, 'identity\_hate': 1405, 'obscene': 8449, 'insult': 7877, 'threat': 478, 'toxic': 15294, 'non\_offending': 143346}

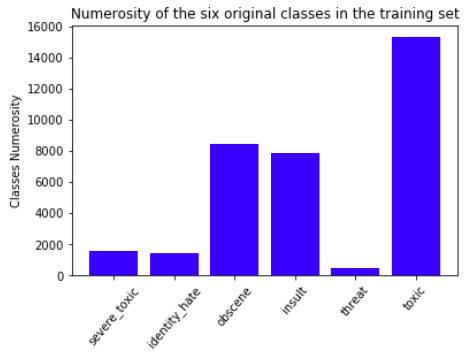
It is important to note that classes can overlap, i.e. a comment can belong to more than one of the offending classes, meaning that the sum of the values in the defined dictionary is greater than the number of the total comments present in the training set.

We plotted numerosity of all the classes, including the new “non-offending” one, in the following bar-plot:



The training set is strongly unbalanced, meaning that the numerosity of the target classes varies a lot: in particular, non-offending comments are much more frequent in the training set than all the other six classes.

We build the same plot with only the six original classes to appreciate the differences between them:



The training set is strongly unbalanced, meaning that the numerosity of the target classes varies a lot: in particular, non-offending comments are much more frequent in the training set than all the other six classes, but also considering only the six categories we see that the toxic class is very common whereas threat is quite rare. This may cause problems during the training of the algorithm because the classifier is more likely to assign new examples to the most numerous classes in the training set, especially if we don't have "enough" examples for each class in the training set.

1. PRE-PROCESSING STEPS

C