Hate Speech Detection
Sentiment Analysis Project based on white supremacist forum Stormfront

Luca Romano, mat. 980068

University of Milan, Italy, luca.romano5@studenti.unimi.it

Abstract. The project aims to create a model which can detect hate speech given a text. The model is trained using messages from Stormfront, an American white supremacist forum. The main focuses of this work consist of the data cleaning, the pre-processing phase with messages tokenization and lemmatization, a descriptive analysis of the most used words taking into account their part of speech, and eventually a statistical analysis to correctly classify hate messages by training Logistic Regression with three different approaches on the dataset: using the original observations, applying SMOTE and lastly applying random under sampler to balance the number of hate and no-hate labels.

Keywords: Supervised Learning · Lemmatization · Text Classification · Imbalanced Dataset

1 Introduction

Nowadays, social media play a vital role in many aspects of our lives. This situation has brought more and more individuals to sign up and interact with each other. On the one hand, social media indeed allow everyone to express their ideas and has contributed a lot to the support of minorities or protests. On the other hand, they exacerbate polarization and -since users are behind a screen- they make individuals feel safer than they would in the real society removing the risk of consequences for violent and ruthless messages. It is not unusual to find insults or other kinds of violent messages just for prevailing in a verbal fight or to get consensus among other people.

The purpose of this project is to create a model able to classify messages detecting whether they contain hate speech or not. To perform this task it was used a dataset containing messages from Stormfront, a white supremacists forum funded by and exponent of the Ku Klux Clan with more than 300k users as of 2018. The data are extracted from a Github repository from Vicomtech which contains a set of text extracted from several subforums and split into sentences. Among the *.txt* files containing the messages, the repository contains also a *.csv* file with the labels manually associated to each text as well as the id of the subforum from which the sentences are extracted. For the purpose of this work, only the labels are used.

2 Research Question and Methodology

This work aims to visually highlight the most common terms used in hate messages as well as to create, train and evaluate a supervised learning model able to classify a message between hate and non-hate. A detailed description of the steps performed is contained in the following two subsections.

2.1 Data Cleaning and Pre-Processing

To achieve the proposed goals, some data cleaning is performed on raw messages. This operation is needed especially due to the type of text we are working with: being extracted from a forum, messages can often contain words with mixed numerical and alphabetical characters, links, mentions, and other kinds of issues. While all these elements provide almost no information to our model, they can cause problems in the encoding terms and the overall analysis, so they need to be removed. To further improve the quality of the data, capital letters are substituted with lower ones. Finally, punctuation is dropped as well as extra spaces and single characters.

Then, the library SpaCy is used firstly to tokenize and lemmatize the texts, and then to determine the part of speech of each word to build lists of the nouns, verbs, and adjectives most used in non-hate and hate-classified messages and perform a more exhaustive descriptive analysis.

To perform the classification tasks on the cleaned messages, it is needed to convert text to numbers and build the corresponding Bag of Words. The TFIDF Vectorizer from sklearn is used for this purpose. It consists of a CountVectorizer which counts the frequencies of each word in the collection, followed by a TfidfTransformer which aims at removing all the words that appear too often in the collection of documents and therefore are not meaningful.

2.2 Classification

Dealing with a binary classification problem, logistic regression is performed initially on the original dataset using a naive Bayes classifier as a baseline. Given the unbalanced distribution, as shown in the next section, between hate and non-hate messages, a more refined classification is performed by training a logistic regression model on the data after applying two balancing techniques from the library *imbalanced learn*: SMOTE and Random Under Sampler. The first one synthesizes new examples from the minority class, and the second one undersamples the majority class by randomly picking samples without replacement.

3 Experimental Results

3.1 Descriptive Analysis

After the pre-processing phase, the label frequencies are plotted to get an overview of their distribution.

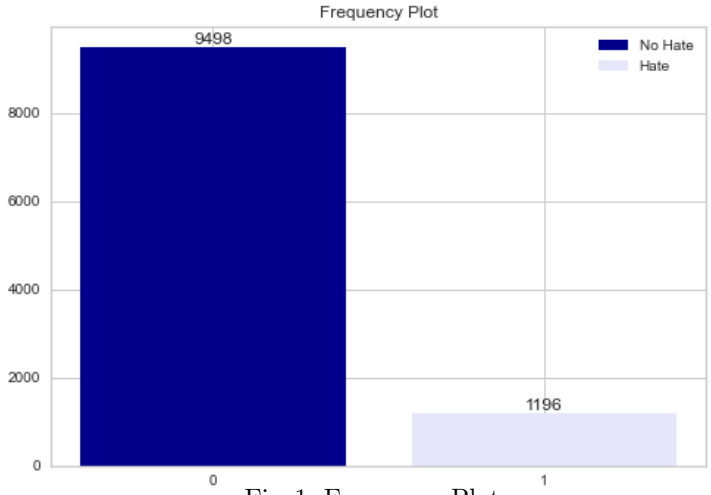


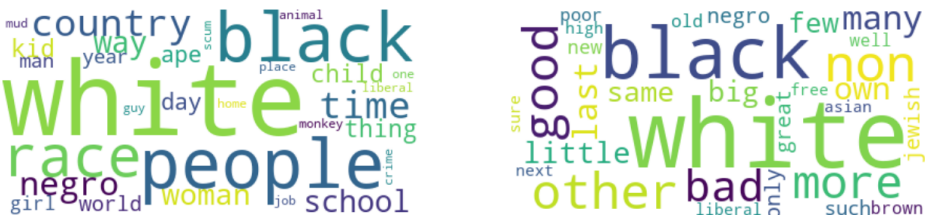
Fig. 1: Frequency Plot

As anticipated in the previous section, the plot shows that the data set is heavily unbalanced. In particular, 9498 labels refer to No-hate messages with just 1196 for the hate ones.

Fig. 2: Most used Words in hate messages

Figure 2 shows the most used words composing a vocabulary of hate messages. Since stopwords were not removed in this work, many of these words are not related to hate: even though there are some meaningful words, like "ape", "jew", "muslim" or "negro", for which it is easy to infer some hate meaning, there are many terms, especially verbs like "will", "think", "make" and many others, which constitutes noise.

A more refined version is obtained using SpaCy's part of speech detection function. This approach allows to filter out the POS containing meaningless words -like verbs in this case- and to focus on the most visually informative ones.



(a) Nouns (b) Adjectives

Fig. 3: Top 30 nouns and Adjectives

As reported in Figure 3, both adjectives and nouns subfigures allow us to observe more significant words that can be easily associated to hate. In particular, in the nouns we can spot, aside from words like "white" or "people" which, as shown in Figure 2, are just commonly used, the expressions "black", "race", "negro" and also "monkey" or "animal" for which is intuitive to associate a hate message. Moving to the adjectives, we have almost the same pattern with some irrelevant terms like "good" or "least" or "few" but also "Jewish" or "Asian" or again, as in nouns, the words "black", "negro".

3.2 Classification Results

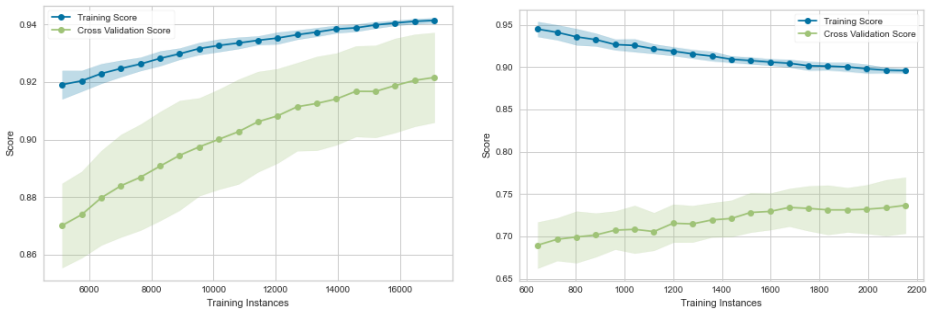
Following the classification approach described in Section 2.2, the first results are related to the straightforward application of logistic regression. Its outcomes are then compared with the naive classifier as a baseline model. As shown in Table 1, using the original data set, the linear regression does a good job in classifying correctly the non-hate sentences and performs just slightly better in accuracy terms but way worse in recall and f1-score with respect to the baseline classifier whose outcomes are already bad when looking at the classification of hate messages. Given these results, it is clear that, due to the highly unbalanced dataset, the model is not able to classify correctly hate texts: even though the precision of 0.68 is way higher than the 0.14 of the naive Bayes estimator, the recall is very poor and of the 478 hate-classified messages, only 21 where correctly classified as a hate post.

Table 1: Classification Results

Model	0-1 ratio test	Precision	Recall	Accuracy	F1-Score
Naive Bayes (original dataset)	0.11	0.14	0.08	0.84	0.10
Logistic (original dataset)	0.11	0.68	0.04	0.89	0.08
Logistic - SMOTE	0.99	0.89	0.92	0.90	0.90
Logistic - RUS	0.99	0.76	0.77	0.76	0.77

To better train the model, two different techniques are applied to the dataset before performing the classification: SMOTE and RandomUnderSampler. In both cases, the outcomes are sensibly higher. Starting with SMOTE, the number of hate-labeled messages is raised from 1196 to 9848, matching the number of non-hate posts in the original dataset. As reported in the table, this balancing act provides the best classification outcome in terms of all the considered indicators with 0.9 in both accuracy and F1-Score. However, looking and the cross-validation plot in Figure 4(a), this model seems to be suffering primarily from error due to variance (the CV scores for the test data are much more variable than for training data), meaning that the model is overfitting and thus it may not be generalized to other data.

Moving to the RandomUnderSampler model, many of the observations of the majority class observations are removed to match the equal size of the hate-labeled messages with 1196 data points. Looking at the indicators, the performance of this model leads to lower accuracy (0.76) and F1-Score (0.77) when compared to the SMOTE model but the plot of the cross-validation score and training score in Figure 4 (b) shows a lower variance error when compared to the respective training score and, since the two curves have not yet converged, the trends suggest that model could benefit from adding more observations.



(a) SMOTE (b) Random Under Sampler

Fig. 4: Learning Curves, 10-fold Cross Validation

4 Concluding remarks

Given the highly unbalanced nature of the dataset, the results obtained in the previous section are satisfactory. Both the models trained with SMOTE and Random Under Sampler provided much better outcomes than the basic application of the Logistic Regression that, unexpectedly, performs worse than the Naive Bayes Classifier for what concerns the recall and thus the F1-Score.

As stated in the Classification Results, the SMOTE model tends to suffer from overfitting: due to how it works, it will not learn more terms related to hate messages but will only learn better the ones it has synthesized. Therefore, its outstanding performances in all indicators are not very meaningful. On the contrary, the Random Under Sampler, even if it is performing worse than the SMOTE one in the presented metrics, has still a way better score than the vanilla Logistic Regression or the Baseline, also showing some potential when looking at its learning curve as it could benefit from adding more observations. Indeed, one possible step in the future could be to expand the Stormfront dataset with messages from other forums or social networks and re-train the model also on these new messages. This new configuration would allow the model to expand the bag of words that is used to detect hate speech in messages and provide a better classification for many more messages with different terminology.

References

1. Vicomtech, Hate speech dataset from a white supremacist forum containing the Stormfront dataset https://github.com/Vicomtech/hate-speech-dataset
2. Jason Brownleee, SMOTE Overview https://machinelearningmastery.com/smote-oversampling-for-imbalanced-classification/
3. Nakul Lakhotia, Hate Speech Detection in Social Media in Python, Github repository: https://github.com/NakulLakhotia/Hate-Speech-Detection-in-Social-Media-using-Python