Cyberbullying detection

AUTHOR

MARTINA SUSTRICO (533252)

TEXT ANALYTICS PROJECT

Data Science & Business Informatics

Academic Year 2021/2022

Contents

Introduction

Cyberbullying has been defined as bullying with the use of digital technologies. It can take place on social media, messaging platforms, gaming platforms and mobile phones. It is repeated behaviour, aimed at scaring, angering or shaming those who are targeted. In cyberbullying, It is challenging to precisely pinpoint the power imbalance, repetition, and destructive purpose that are often associated with traditional bullying because of its digital form and relative anonymity. The goal of this work is to enhance active machine learning cyberbullying detection on **Twitter**, with an emphasis on the capability to identify the characteristic of the victim that the cyberbully is targeting, such as **age, ethnicity, gender, religion, or other**. To do so, the dataset *Cyberbullying Classification* (`kaggle.com/datasets/andrewmvd/cyberbullying-classification`) from Kaggle has been used. It is composed by 47k tweets belonging to 6 balanced classes. The classification of tweets was carried out using three different approaches in order to explore the differences in performance:

1. **Binary classification.** For which two labels were created: "NotCyberbullying" and "Cyberbullying". The class "Cyberbullying" includes all the different original categories.
2. **Multi-class classification.** During which several basic classification models were used.
3. **Classification with BERT.** BERT is a a pre-trained deep learning model, based on transformer architecture.

The entire project was developed in **Python** both with **Jupyter Notebooks** and **Google Colab**. In a first stage, it has been done an analysis in order to understand the data, and all tweets have been cleaned of noise as punctuation, numbers, links, mentions. The pre-processing phase ends with lemmatization and tokenization of the text (Chapter 1). Chapters two, three and four are dedicated to the classification tasks mentioned above. The algorithms used and compared are Naive Bayes, SVM, Logistic Regression, Decision Tree, Random Forest, BERT.

TRIGGER WARNING. These tweets either describe a bullying event or are the offense themselves, therefore if you are sensitive to strong language, do not continue reading.

1 Data Understanding and Cleaning

Initially the dataset contained 47,692 tweets and 2 features: **Tweet Text** and **Cyberbullying Type**. The type is a categorical attribute that give the information about the category of cyberbully related: Age, Gender, Ethnicity, Religion, Other Cyberbullying and Not Cyberbullying. Upon observing some of the tweets in the 'other_cyberbullying' category, I discovered that they were random tweets, some of them without any bullying overtones, so I decided to remove them - instead of putting them in the 'not_cyberbullying' category. Moreover, I removed also 36 dupllicated tweets. After removal, the new dataframe is composed by **39,833 tweets** whose distribution is shown in figure 1.1.

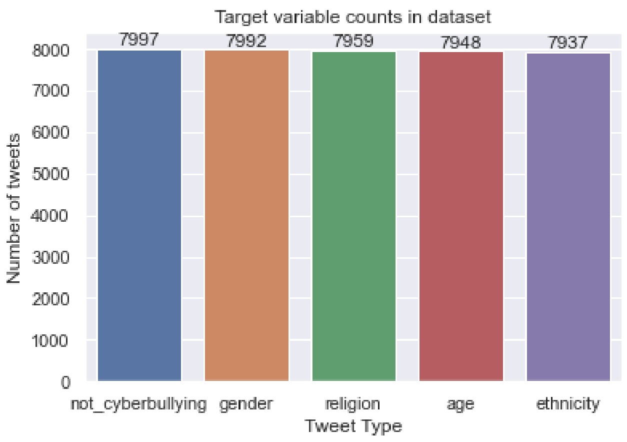


Figure 1.1: Target variable counts

In order to clean the tweets, I performed very basic preprocessing on every tweet, stripping emoticons, links, mentions (@username), the retweet flag "RT" (and other noisy words/characters), punctuation, multiple spaces and word with more than 14 characters. Importantly, I didn't remove stop words or hashtags because I think they are important component of the text and some stop words might be helpful context for the cyberbullying detention task (such as the use of 'not' or the use of male or female pronouns). Moreover, a function has been used in order to decontract verbs and other particular words. Finally, I used **WordNetLemmatizer()** from **nltk** library, to lemmatize the sentences. Lemmatization is the process of converting a word to its base form, it is important to highlight the fact the lemmatization considers the context and convert the words to their meaningful base form. At the end of all the cleaning process, exploiting again the nltk library, the sentences have been tokenized. **Tokenization** is a way of separating a piece of text into smaller units called tokens (can be either words, characters, or subwords). In this analysis each token is a word of the tweet. Removing the duplicates tweets, at the end of the whole preprocessing there are **39,263 tweets**. For better data handling, the target variable was trasformed into a discrete numeric variable, associating a number with each category.

1.1 Word clouds

After cleaned the sentences of all noise, in this section some word clouds are shown, in order to capture the more frequent words for each bullying category. For this task **wordcloud** library has been used. The related results are shown in figures 1.2 and 1.3, in which there is a word-cloud script for each category. Even from a first impression it is possible to notice the difference in words used in a tweet belonging to the class 'non_cyberbullying' and one labeled as bullying.

For 'not_bullying' class the ten most common words (ignoring stop words) are 'mkr' that stands for My Kitchen Rules (an Australian competitive cooking game show), 'bully', 'not', 'do', 'can', 'go', 'just', 'like', 'have', 'school'.

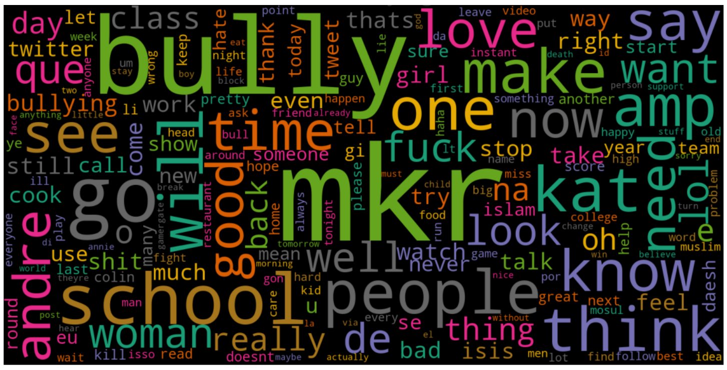
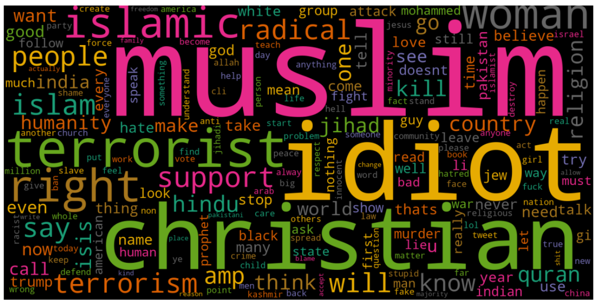
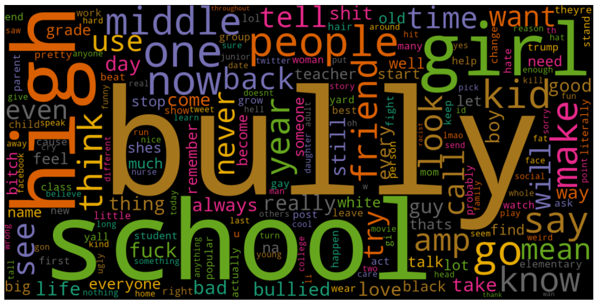


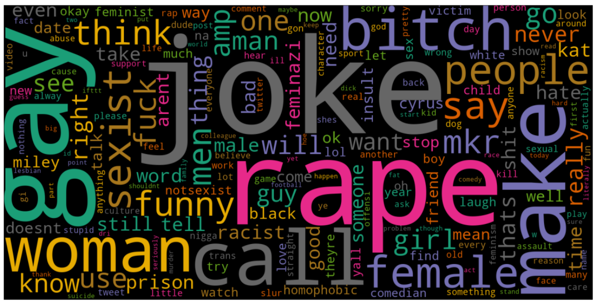
Figure 1.2: Word Cloud - Not Bullying



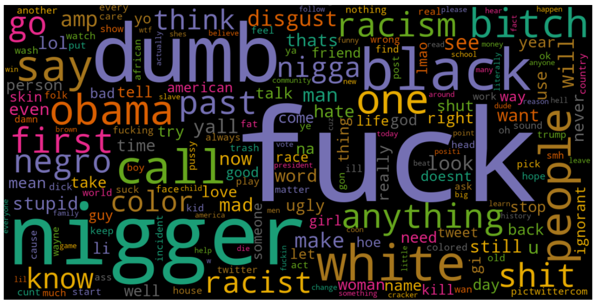
(a) *Religion*



(b) *Age*



(c) *Gender*



(d) *Ethnicity*

Figure 1.3: Word clouds for each cyberbullying class

2 Binary Classification

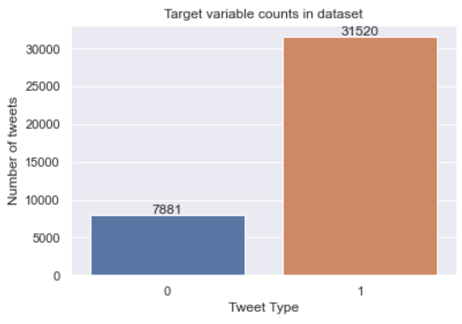


Figure 2.1: Imbalaced dataset

The binary classification task aims to classify whether a tweet refers to cyberbullying situations or not. For this purpose, since the dataset is born as multi-class single-label problem, all the tweets that refer to a cyberbully class have been merged in a single label. From here on, we will refer to cyberbullying class using the tag '1', '0' otherwise. From the figure 2.1, it is possible to notice the huge imbalance that plagues the dataset. In order to reduce that gap, and in light of the distribution of word count (figure 2.2), I decided to remove all the tweets with more than 45 words. The new distribution is **{'0': 6141, '1': 24143}**. Keeping the focus of the tweet count with the word distribution, one could make a point about how the two distributions are different. In fact, non_cyberbullying tweets are composed of fewer words than cyberbullying tweets, this could be due to the fact that cyberbullying tweets are -among other things- stories or accounts of bullying events.

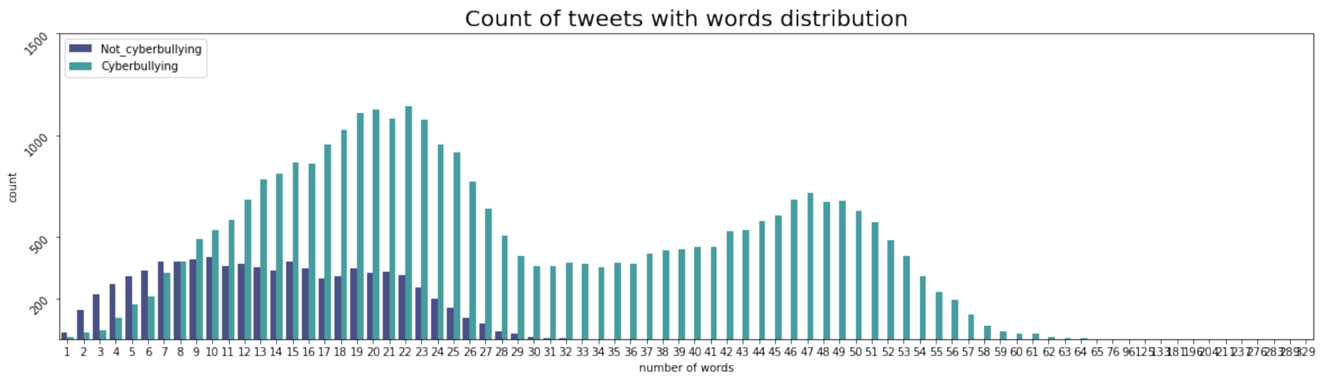


Figure 2.2

2.1 Data preparation

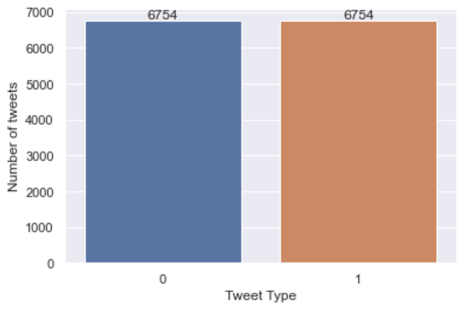


Figure 2.3: Balanced dataset

Before we move into the core of the analysis, it is necessary to prepare the dataset in a proper manner. First of all, the original dataset is splitted in training set (70%) and test set (30%). Then, the training set was **balanced** by oversampling and undersampling techniques, reaching the distribution shown in figure 2.3. Moreover, to improve the following analysis steps, I decide to apply the **SelectKBest** algorithm able to select just the 2000 more relevant features. Then, **TF-IDF** transformation is applied to associate weights to the different words based on their frequency (rarer words will be given more importance).

2.2 Classification

Classification models trained in this task are the following:

- **Naive-Bayes**: *MultinomialNB()*
- **Support Vector Machine**: *LinearSVC()*
- **Logistic Regression**: *LogisticRegression(C = 0.5, solver = "sag")*
- **Decision Tree**: *DecisionTreeClassifier(min_samples_leaf = 20, min_samples_split = 30, criterion = 'entropy')*
- **Random Forest**: *RandomForestClassifier(min_samples_leaf = 20, min_samples_split = 30)*

In this section, in table 2.1 the results obtain within each classifier will be shown and compared.

Classifier	Label	Precision	Recall	F1-Score	Accuracy
Naive Bayes	0	0.74	0.71	0.72	0.89
	1	0.93	0.94	0.93	
SVC	0	0.70	0.89	**0.79**	**0.90**
	1	0.97	0.90	**0.94**	
Logistic Regression	0	0.68	0.92	0.78	0.90
	1	0.98	0.89	0.93	
Decision Tree	0	0.69	0.94	**0.79**	**0.90**
	1	0.98	0.89	**0.94**	
Random Forest	0	0.66	0.93	0.77	0.89
	1	0.98	0.88	0.93	

Table 2.1: Binary classifier performance

As it is possible to notice, performance in terms of accuracy, are similar among all classifiers; but whereas the dataset is unbalanced with fewer observations for class 0, we are more interested in classifying well the records belonging to that class. For this reason, it is best to consider as comparative measure the F1-Score related to class 0. Again, all the classifiers are quite similar, small exception for Linear SVC and Decision Tree that reach 79% of F1-Score for class 0.

2.3 Explainability - Naive Bayes

Explainability, sometimes referred to as 'iterpretability', refers to the idea that a machine learning model and its output may be adequately explained in terms that "make sense" to a human being. For this task I decide to explore the Naive Bayes models' black-box, because from a calibration probabilities analysis [1] turned out that was the most accurate model. For this purpose, the LIME algorithm was used, taking a random instance of the dataset. Figure 2.4 shows the results, where words highlighted in orange support 'Cyberbullying' tweets, those in blue 'Non_Cyberbullying' (the higher the importance of the word, the lower the transparency).

[1]Look at Jupyter Notebook '2.1 Binary Classification' - Calibration probabilities section

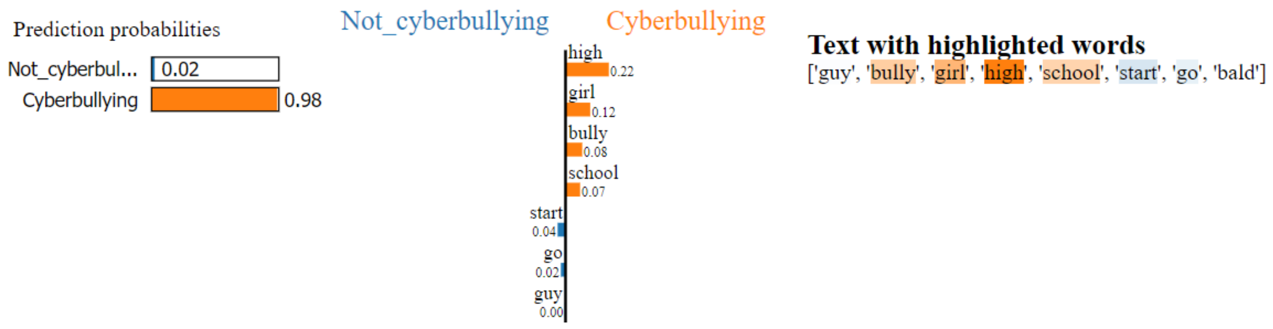


Figure 2.4: LIME binary classification explainability

3 Multi-class Classification

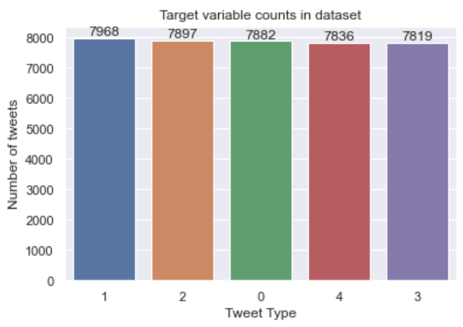


Figure 3.1: Dataset target variable

The multiclass classification task aims to build a model that can recognize which category of cyberbullying a tweet belongs to. In this stage some classical machine learning algorithms were been trained, using the original clean balanced dataset. As the categorical feature has been encoded in a numerical, here the reference encoding: **"not_cyberbullying": 0, "religion": 1, "age": 2, "gender": 3, "ethnicity": 4**; and the releated distribution shown in figure 3.1. Then, to better clean the data, all the tweets with more than 60 words were deleted. Since the classifiers used had worse performance in predicting "not_cyberbullying" tweets (class 0), I decided to oversample with RandomOversampler to make the dataset balanced.

As always, to train a model is necessary to have a training and a test sets. To do so, I divided the original dataset with proportions 67-33%. For the rest of dataset preparation, regarding data transformation (tokenization, feature selection, word weighting), please refer to section 2.1.

3.1 Classification

Of course, I used the same classification models for this task as well, so that I would have a term of comparison.

- **Naive-Bayes**: *MultinomialNB()*
- **Support Vector Machine**: *OneVsRest(LinearSVC())* and *OneVsOne(LinearSVC())*
- **Decision Tree**: *DecisionTreeClassifier(min_samples_leaf = 20, min_samples_split = 30, criterion = 'entropy')*
- **Random Forest**: *RandomForestClassifier(min_samples_leaf = 20, min_samples_split = 30)*

Since the svm model is not designed for multiclass classification problems, it is necessary to use a ploy to adapt the model to these circumstances as well. The idea is to divide the multiclassification problem into binary classification subproblem. To do so it is possible to use the **One-to-One** approach (a binary classifier per each pair of classes), or the **One-to-Rest** approach (a binary classifier per each class). Below there are the summary tables with the performance of all trained classifiers.

	Naive Bayes			OneVsRest SVC		
class	Precision	Recall	F1-Score	Precision	Recall	F1-Score
not bully	0.80	0.58	0.67	0.83	0.85	0.84
religion	0.84	0.96	0.89	0.94	0.96	0.95
age	0.82	0.96	0.89	0.96	0.98	0.97
gender	0.89	0.82	0.85	0.94	0.87	0.90
ethnicity	0.91	0.93	0.92	0.98	0.99	0.98

Table 3.1: Naive Bayes **Accuracy 85%** - 1vsRest SVC **Accuracy 93%**

	Decision Tree			Random Forest		
class	Precision	Recall	F1-Score	Precision	Recall	F1-Score
not bully	0.77	0.90	0.83	0.77	0.87	0.82
religion	0.93	0.93	0.93	0.94	0.95	0.94
age	0.99	0.96	0.97	0.96	0.98	0.97
gender	0.93	0.82	0.87	0.95	0.79	0.86
ethnicity	0.98	0.97	0.98	0.97	0.98	0.98

Table 3.2: Decision Tree **Accuracy 92%** - Random Forest **Accuracy 91%**

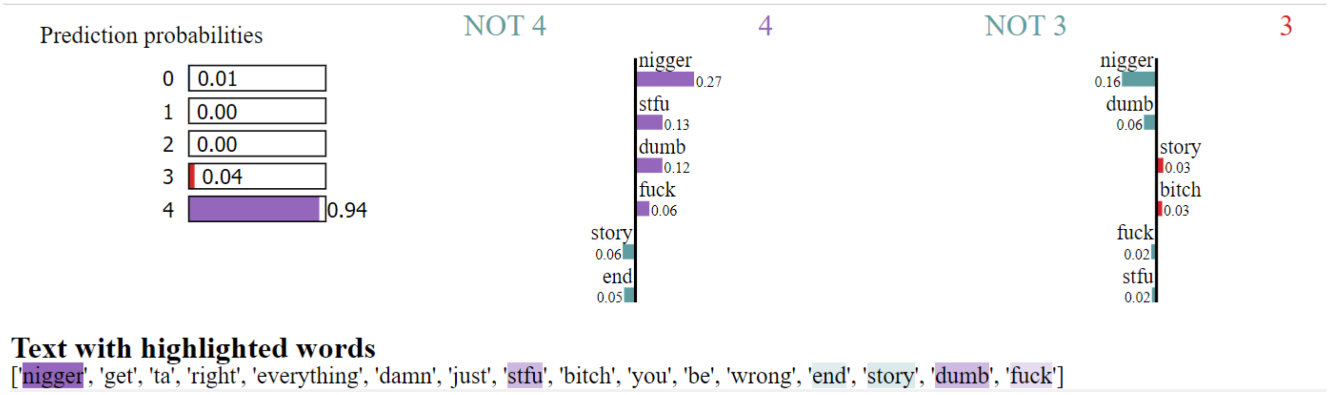
One-Vs-Rest SVC guarantee the best performance both in terms of accuracy and "not bully" class prediction.

3.2 Explainability - Naive Bayes

Also for multiclass classification task I decided to explore the black-box of Naive Bayes model by LIME explainability algorithm, in order to have a comparison between this one and the binary one.

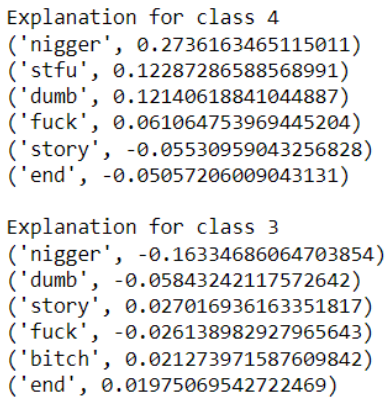
LimeTextExplainer function has been used to extract an instance's prediction probabilities, considering the weight of words associated with each class.

In figure 3.2 there is the explanation of a random instance, correctly classified as 'ethnicity' cyberbullying tweet (class 4) from Naive Bayes classifier. In purple there are the words that support 'ethnicity' class and in red the 'gender' class (class 3) - second class with a higher probability. All the details about the strength of each word of the tweet is shown in figure 3.3.



Text with highlighted words
['nigger', 'get', 'ta', 'right', 'everything', 'damn', 'just', 'stfu', 'bitch', 'you', 'be', 'wrong', 'end', 'story', 'dumb', 'fuck']

Figure 3.2: LIME multiclass explainability



```
Explanation for class 4
('nigger', 0.2736163465115011)
('stfu', 0.12287286588568991)
('dumb', 0.12140618841044887)
('fuck', 0.061064753969445204)
('story', -0.05530959043256828)
('end', -0.05057206009043131)

Explanation for class 3
('nigger', -0.16334686064703854)
('dumb', -0.05843242117572642)
('story', 0.027016936163351817)
('fuck', -0.02613898292796564З)
('bitch', 0.021273971587609842)
('end', 0.01975069542722469)
```

Figure 3.3: LIME explanation for class 4 and 3

4 Classification with BERT

In this section I'll use the HuggingFace's *transformers* library to fine-tune pretrained **BERT Base model** for a tweets classification task. BERT Base contains 12 Transformer Encoders, 12 attention heads and 110M parameters.

BERT architecture is composed by a stack of Encoder. It takes a sequence of words as input and process it up the Encoder stack. Each layer applies **self-attention**, and passes its results through a feed-forward network, and then hands it off to the next encoder. BERT also makes use of **Transformer**, an attention mechanism that learns contextual relations between words in a text (learns how important all of the words in the sentence are by looking at their specific positions in the sentence).

4.1 Pre-Processing

I will lightly treat tweets before tokenizing them, taking out entity mentions (such as @user) and a few unusual characters. Because BERT was trained using the whole sentences, the level of processing is substantially lower than in earlier methods. It is necessary to use the tokenizer (**BertTokenizer**) offered by the library in order to apply the pre-trained BERT. This is due to the fact that the model has a predetermined, set vocabulary and the BERT tokenizer has a specialized method for handling terms that are not in its dictionary. Additionally, there is a need to pad and truncate all phrases to a single consistent length, add special tokens to the beginning and end of each sentence, and use the "attention mask" to explicitly identify the padding tokens.

4.2 Set Up Model

The transformers library has the **BertForSequenceClassification** class which is designed for classification tasks. Then, I created an iterator for the dataset using the **tensor DataLoader** class. This will quicken the training process and aid in memory preservation. To fine-tune Bert Classifier, it was necessary to create an optimizer ($AdamW$), with the following parameters:

- Batch-size: 16
- Learning rate (Adam): 1e-5
- Number of epochs: 2

Finally, the model was trained and the accuracy levels obtained for each class are the following:

Label	Category	Accuracy
0	Not Bully	0.88
1	Religion	0.97
2	Age	0.98
3	Gender	0.90
4	Ethnicity	0.98

Conclusion

The purpose of this project was to compare the different classification's models in order to correctly detect cyberbullying behaviors on twitter. Several algorithms were trained in both a binary and multiclass classification context. In light of the results obtained, it can be concluded that as far as binary classification is concerned, all models ensured accuracy around 90 percent by succeeding in classifying class 1 better than class 0. The same can be said for multiclass classification, with which it is more accurate to predict tweets of some kind of cyberbullying rather than 'not_cyberbullying' tweets. These results could be a consequence of the fact that tweets without cyberbullying, having no particular restrictive context (in terms of words), could deal with all kinds of and very different topics. An example might be 'not_cyberbullying' tweets that contain the word "school," which also appears very often in "age" cyberbullying tweets. In this case, such tweets, could be misclassified as cyberbullying when in fact the word "school" is placed in a different context. In part, this problem, was fixed with the use of the pre-trained BERT model, which showed an increase in the accuracy of all classes, with particular regard to the 'not_cyberbullying' class.