Text Mining and Natural Language Processing

2022-2023

Alice Menna-502172

Hate Speech

Introduction

Through sentiment analysis, it is possible to analyze a text or a sentence to detect emotions or classify a positive, negative or neutral sentence. In this specific dataset, named 'Dynamically Generated Hate Speech Dataset'**,** the main task is to predict if a sentence deals with hate speech.

Sentiment analysis can be performed using some machine learning techniques as SVM and Logistic Regression. For this reason, in this project, both are used to compute prediction, in order to understand the suitable algorithm for this kind of data.

In addition, another goal of this project is to compare vectorization techniques, Tf-idf and Bag of Words. Using comparison, it is possible to understand if this kind of choice affects model accuracy.

Data

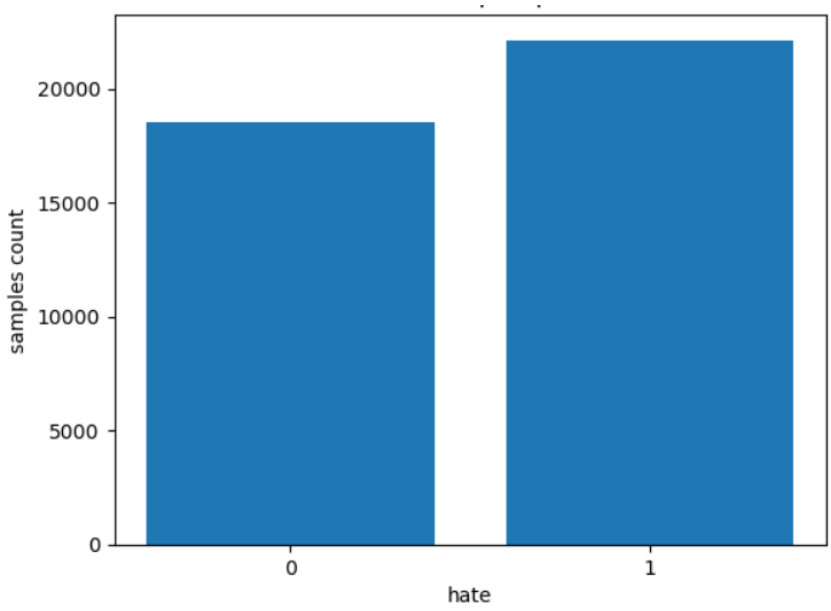
This dataset contains 40.000 sentences about ordinary topics, such as women, gay people, foreigners,

disabled people and so on. All these sentences deal with personal judgements that, in some cases, underline a racist, homophobic, misogynist and, in general, offensive thinking.

The original dataset is composed by 11 columns but, since the aim of this project is to classify the sentences as offensive ('hate') or not offensive ('not hate'), only text and label columns are considered. All the other columns are removed with the *drop* method.

The possible outcomes are slightly imbalanced:

- **'hate': 22124**
- **'not hate': 18499**



Methodology

In order to compute sentiment analysis, the first step is to prepare and clean data. This process is called **preprocessing.** Preprocessing is used to eliminate the unnecessary parts of data.

Preprocessing stages:

- Lowercasing.

  For simplicity, all characters are convert into the same casing format. I apply the method lower() to all *text* column.

  Example

  ```
  i don't work this hard so that those immigrants can take all the benefits
  ```

- Eliminating punctuation marks

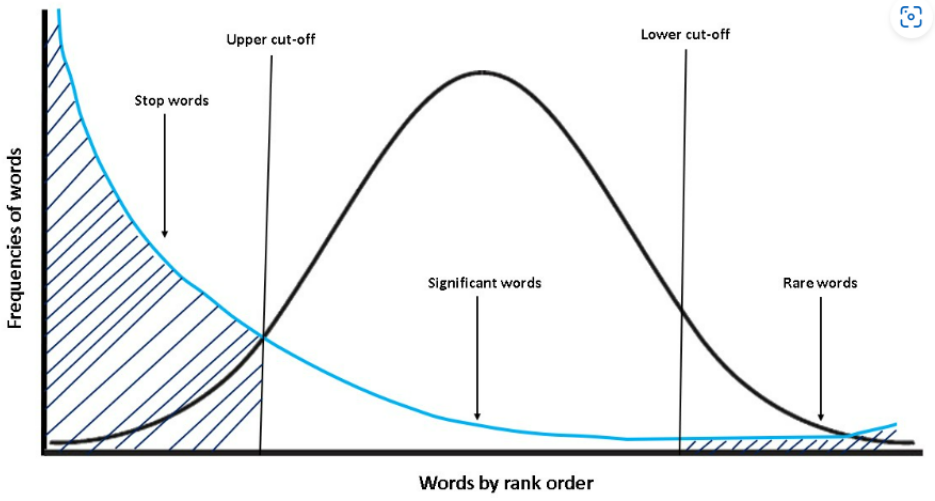
  Removing punctuation marks is a useful tool to consider all text equally. In this way ‘data’ and ‘data!’ are considered the same. I apply the method replace().

  Example

  ```
  i don t work this hard so that those immigrants can take all the benefits
  ```

- Eliminating 'stop words'

  Stop words correspond to all meaningless words that, according to Zipf’s Law, have a high frequency in the text. These words are, for example, articles or pronouns which are useful for construction of the sentence but useless for classification task.



To eliminate stop words I use nltk (Natural Language toolkit) which is a set of libraries useful for text analysis. Nltk, also containing stop words elimination methods in different languages. Since

the sentences are written in English, the English method is chosen.

Example

```
work hard immigrants take benefits
```

- Tokenization and Lemmatization

  To tokenize means to split the sentence in tokens. I decide to apply word tokenization, so I divide the sentence considering words. Tokenization is important because it helps to understand the meaning of words.

  Last step of preprocessing is lemmatization. Lemmatization, which considers the context, converts words in their meaningful base form, defined as Lemma. In tasks as sentiment analysis, it is necessary to have a base word, in fact, grouping together the different inflected forms of a word, different inputs can be analyzed together. Using lemmatization can lead to several advantages, such as:

  1. Improve accuracy, since reducing words to their Lemma, makes it easier to analyze them when they have similar meanings.
  2. Reduce data size.

  In order to compute tokenization and lemmatization, I apply Natural Language toolkit (nltk) and in particular WhitespaceTokenizer()and WordNetLemmatizer() methods.

  Example

```
[work, hard, immigrant, take, benefit]
```

As soon as preprocessing ends, next step is vectorization. Vectorization transforms data in vectors of real numbers. There are several methods to vectorize input, in this project Tf-Idf and Bag of words are used. I decide to use both in order to confront them, since Tf-Idf solves the main limitation of Bag of words. In fact, Bag of words only focuses on the frequency of words and in this way, articles, pronouns (meaningless words) are considered as much as meaningful words like, adjectives, nouns and so on. On the other hand, with Tf-Idf, meaningful words are not overpowered by meaningless ones.

Tf-Idf has two parts, Tf and Idf.

- Tf which is the Term Frequency.

$$TF = \frac{Frequency\ of\ word\ in\ a\ document}{Total\ number\ of\ words\ in\ that\ document}$$

- Idf which is the Inverse Document Frequency, which calculates each word's relevance in the text.

$$IDF = \log(\frac{Total\ number\ of\ documents}{Documents\ containing\ word\ W})$$

To implement Tf-Idf, I apply sklearn library with TfidfVectorizer().

Bag of words involves 3 operations, which are tokenization, vocabulary creation and vector creation. Obtaining tokenize words, a vocabulary is created with words sorted in alphabetical order and, in the end, a sparse matrix is created. To summarize, bag of words deals with the frequency of vocabulary words in the text.

To implement Bag of words I use sklearn library with CountVectorizer().

For the aim of this project, I decide to use machine learning techniques.

The models used to achieve the task are:

- SVM

  Support Vector Machine is a supervised machine learning algorithm used for classification tasks. The goal is to find an optimal hyperplane in a N-dimensional space that can separate data in classes in feature space.

- Logistic Regression

  Logistic Regression is a supervised machine learning algorithm used for classification tasks. It estimates the probability that an input belongs to one class.

To evaluate the performance, I used accuracy since the classes are slightly unbalanced. Accuracy is defined as:

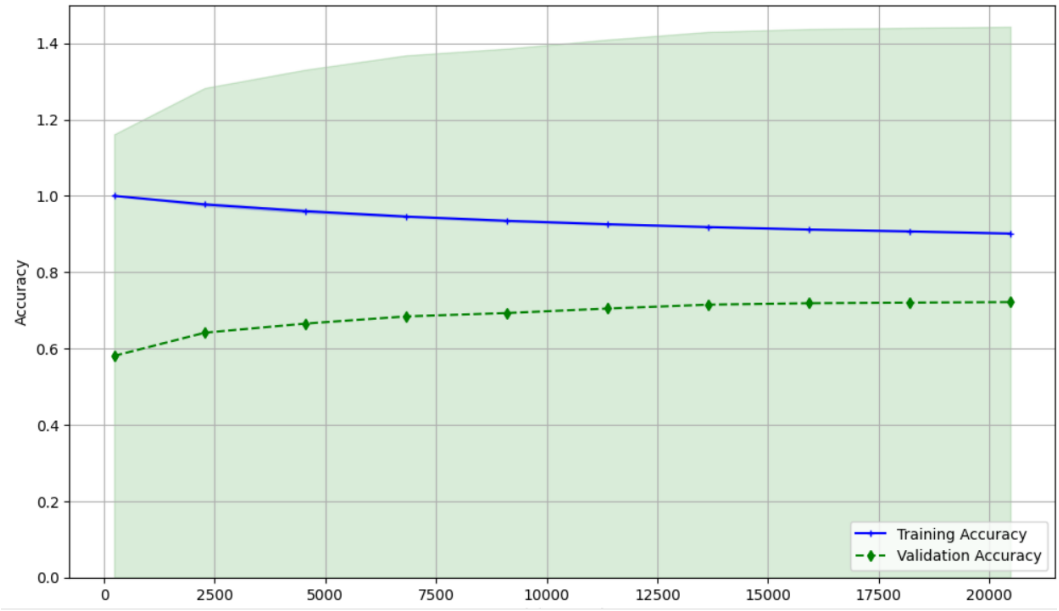
$$\frac{(TP + TN)}{(TP + FP + TN + FN)}$$

Result

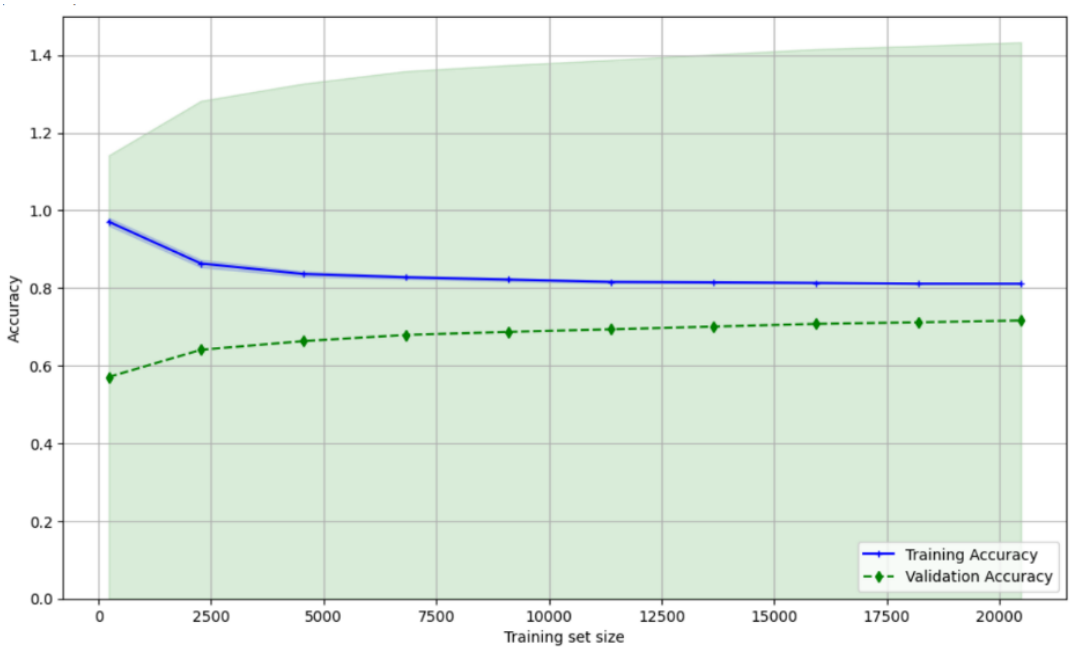
Summary	Tf-idf	Bag of Words	Logistic regression	SVM	Accuracy
Model 1	x			x	0.72
Model 2	x		x		0.72
Model 3		x		x	0.73
Model 4		x	x		0.69

The table shows the obtained results.
The best accuracy is reached training SVM model and using Bag of Words technique.
However, there are not so many differences between the 4 possible combinations. This means that, given this type of data, choosing either tf-idf or Bag of words as vectorize technique does not make a huge difference. This kind of result is shown also for train model algortihms, SVM and Logistic Regression.
To better understand the above results, here is the Learning Curve of all the possible implementations considering accuracy.

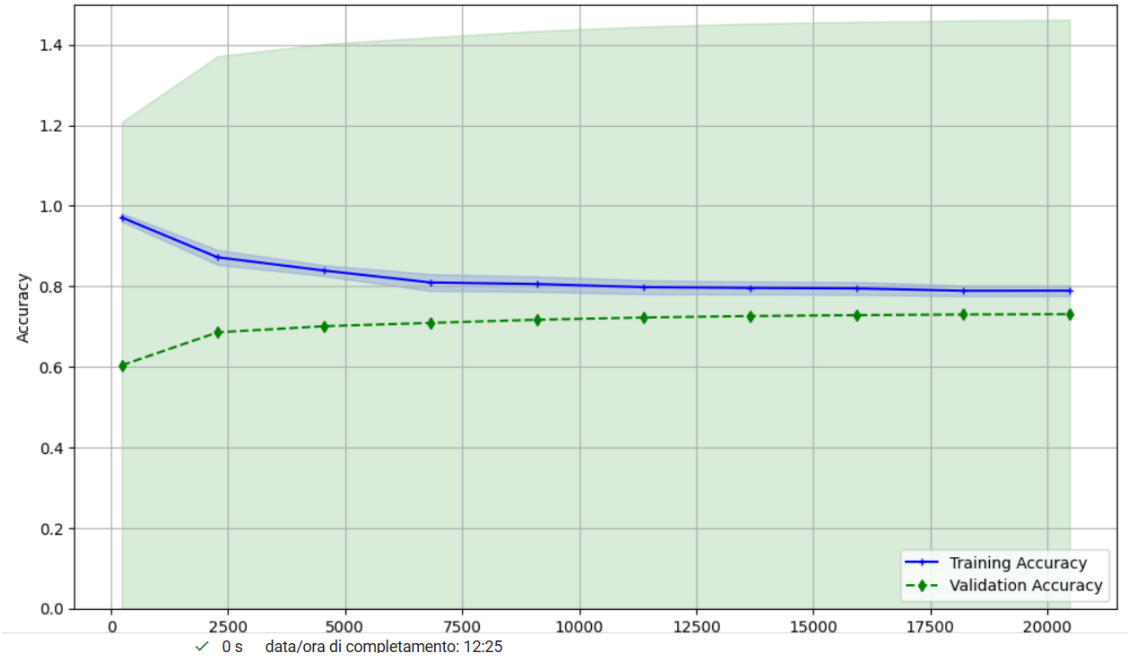
- Tf-idf, SVM



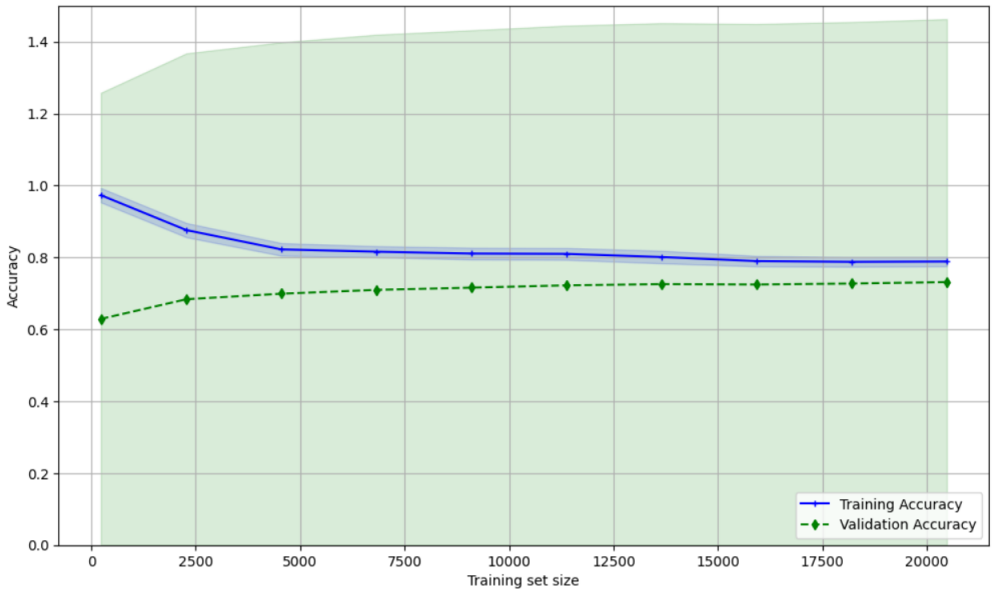
- Tf-idf, Logistic Regression



- Bag of words, SVM



- Bag of words, Logistic Regression



Conclusion

The proposed system employs supervised machine learning architectures such as SVM and Logistic Regression to classify if a sentence reflects offensive language or thoughts. The aim of this project is to confront 2 methods of vectorization, tf-idf and Bag of words, to verify how they influences the accuracy of the model.

The implementation requires a first step of preprocessing where:

- all characters are converted in lower case,
- all stopwords and punctuation marks are removed.
- Text is tokenized and words are lemmatized.

After preprocessing, all words are vectorized with tf-idf and bag of words and then Logistic regression and SVM are applied. Four possible combinations are tested:

- Tf-idf and SVM
- Tf-idf and Logistic Regression
- Bag of words and SVM

- Bag of words and Logistic Regression

The results show that, regardless of the chosen model, the accuracy score is similar for all the combinations. It means that, for this kind of data, choosing either Tf-idf or Bag of words does not affect the result.

References

[1]George B. Aliman, Tanya Faye S. Nivera, Jensine Charmille A. Olazo, Daisy Jane P. Ramos, Chris Danielle B. Sanchez, Timothy M. Amado, Nilo M. Arago, Romeo L. Jorda Jr., Glenn C. Virrey, Ira C. Valenzuela; Sentiment Analysis using Logistic Regression. Journal of Computational Innovations and Engineering Applications JULY 2022: 35-40.

[2] Rohan Shiveshwarkar1, Om Shende2, Soudagar Londhe, Siddhesh Ramane4, Prof. Prajakta A Khadkikar. Review on Sentiment Analysis on Customer Reviews. International Research Journal of Engineering and Technology (IRJET).

[3] Vishal A. Kharde, S.S. Sonawane. Sentiment Analysis of Twitter Data: A Survey of Techniques. International Journal of Computer Applications (0975 – 8887) Volume 139 – No.11, April 2016.

[*4] Imamah*, Husni, Eka Malasari Rachman, Ika Oktavia Suzanti, and Fifin Ayu Mufarroha. Text Mining and Support Vector Machine for Sentiment Analysis of Tourist Reviews in Bangkalan Regency. Journal of Physics: Conference Series.*