U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

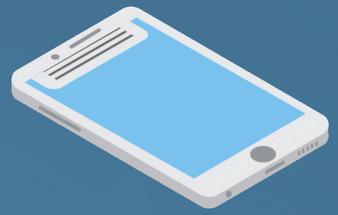
NATURAL LANGUAGE PROCESSING

DETECTION OF EMOTIONS IN SMALL TEXTS

ARTIFICIAL INTELLIGENCE, L.EIC

CHECKPOINT PRESENTATION

Bruno Mendes *up201906166@up.pt*
David Preda *up201904726@up.pt*
Fernando Rego *up201905951@up.pt*



The problem

- Detect underlying emotions in small textual samples
- Seek for the highest precision in reasonable processing time, while avoiding hand coded and not easily maintainable word rules
- Be able to detect possibly dangerous emotions, for use in social media scenarios:
  - anger
  - fear
  - sadness

Related work

Social networks, such as Facebook or Twitter, where the users can share their thoughts and express their feelings, are an example where Natural Language Processing operates to improve users' security and trustworthiness.

Besides the most known examples, we had a look at community solutions to detect emotions in small texts:

- Detecting emotions in texts with Naïve Bayes
- Emotion detection from text

Our approach

- Input data preprocessing
  - Sentence tokenization
  - Word normalization
    - Lowercase words and remove typos
    - Use stemming and lemmatization to compare words with similar meaning
- Train and test dataset split
- Feature extraction
  - Word distribution per emotion based on TF-IDF
- Classification
  - Generative models
    - Naïve Bayes
  - Discriminative models
    - Neural networks
    - Decision trees
- Evaluation
  - Performance measures on the different test sets

Work carried out

- Development environment:
  - Programming language: Python
  - Used libraries: Pandas, NLTK, Sci-kit Learn.
- Work developed:
  - Tokenization and TF-IDF with the use of NLTK;
  - Several classification algorithms:
    - Random Forest;
    - Naive-Bayes (Gaussian and Multinomial);
    - Multi-layer Perceptron;
  - Model creator and tester interface, which allows users to predict the underlying emotion of their own small texts, training the model only the first time a new dataset is loaded.

Data preprocessing

For the data preprocessing, the following methods were implemented:

- Tokenization, using NTLK's TweetTokenizer, which takes into account relevant social-media related features, such as hashtags;
- Removal of stop words;
- Removal of non-relevant words, such as a single-character words;
- Lowercasing;
- Lemmatization;
- Bigram;
- PosTag;

It is worth noting that these methods can and were used both together and independently, in order to achieve better results.

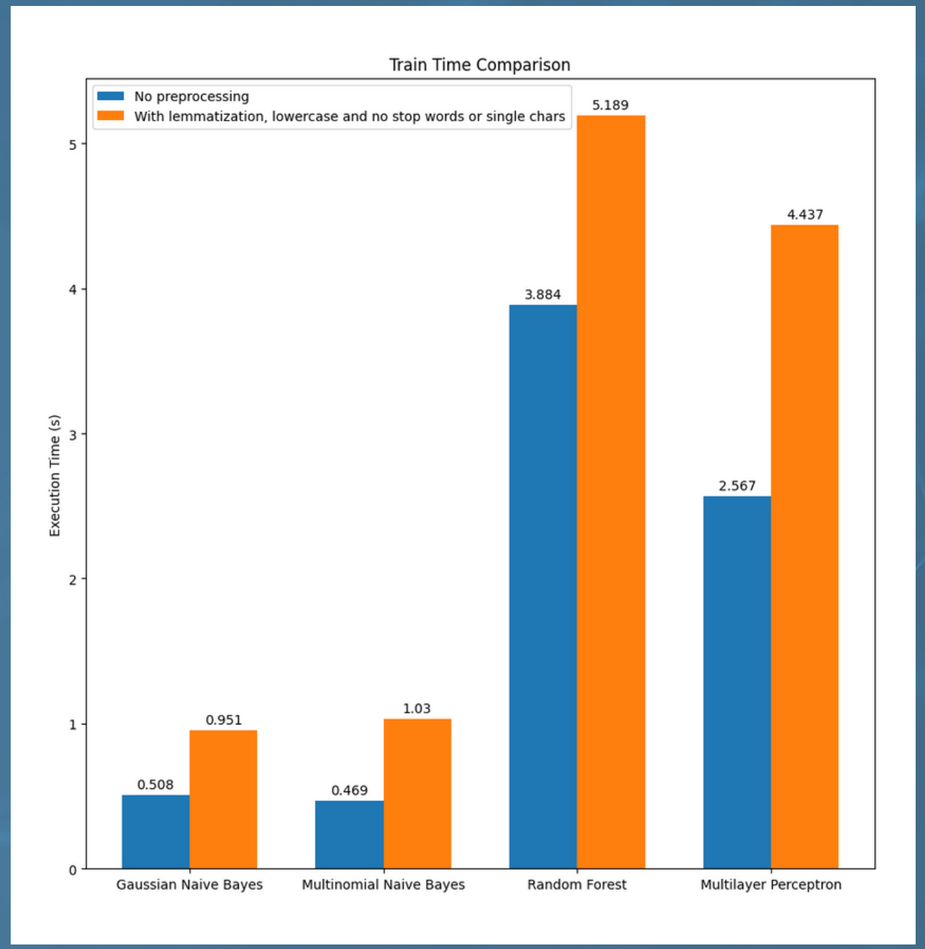
Developed models

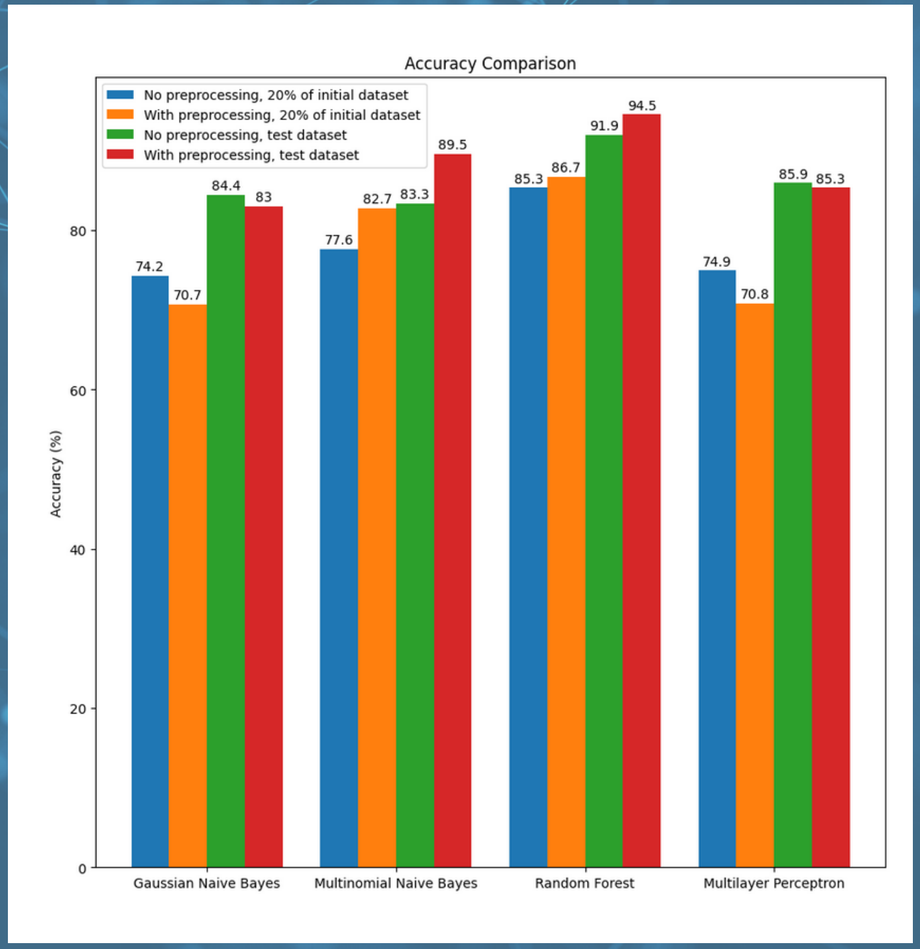
All models use TF-IDF as their basis, to convert tokens into data that the classifiers are able to understand. The BoW technique was also tried out, with consistently worse results.

Therefore, with TF-IDF as the basis, the developed models differ only on their classifier. The following classifiers were used:

- Random Forest;
- Gaussian Naive-Bayes;
- Multinomial Naive-Bayes;
- Multi-layer Perceptron.

Models comparison





The view

Natural Language Processing

Detection of emotions in small texts

Text Input

im very happy

Evaluate

Emotion: joy

Evaluate Dataset

20% train dataset:
accuracy: 0.843; precision: 0.85; recall: 0.843

Test dataset:
accuracy: 0.85; precision: 0.843; recall: 0.85

Duration: 86.829 seconds

Confusion Matrix of Test Dataset

227	10	7	3	14	1
6	186	4	0	10	15
25	9	629	49	28	12
1	2	26	103	5	0
14	5	20	2	518	0
2	12	9	2	6	38

Random Forest

Dataset 2

Remove Stop Words ●

Lowercase ●

Lemmatize ●

Remove Single Chars ●

Bigram ○

PosTag ○