**Team Orange**

Big Scale Analytics Course  
Master of Information Systems - HEC Lausanne  
Prof Michalis Vlachos

**Team Members:**

* Cyrill Oswald
* Tobias Tingström
* Patrick Meireles
* Constant Blanck

**Disaster Tweets, A Kaggle Competition – Text classification**

We are working on the Kaggle competition “Real or not? NLP with Disaster Tweets”. The dataset contains about 10’000 tweets with vocabulary linked to disasters. For each tweet we want to predict if it is talking about a real disaster. We will have a short overview about the machine learning models we created and how we proceeded to improve their results.

**Project structure**

Our project is split in three parts:

1. **Data cleaning** in the notebook CLEANING
2. **Models development** in the notebook MODELS
3. **Voting of the models** in the notebook VOTING

*Note that the notebooks should be run in that order, as dependencies are installed in the data cleaning notebook.*

**Project path**

While reading the notebooks, be aware that we arrived to these results after several iterations. We will explain here the different steps we made. To improve their understanding, the parts of the notebooks will be annotated with the number of the Kaggle submission for which they were added. (e. g. *Submission 1*)

In the **CLEANING** notebook, we import the Kaggle dataset and perform some cleaning on the tweets before saving a new cleaned dataset that will be used in the *MODELS* notebook.

We started by getting rid of special characters, numbers and URL-links and putting the tweets to lowercase. Then we lemmatized and removed stopwords. This has been done for *Submission 1*.

Only later were added the deduplication (*Submission 9*).

For the **MODELS** notebook, our initial attempt was training a Logistic Regression (*Submission 1*). This gave us the best overall results (see table 1). After that, we fine-tuned the hyperparameters of the LR using SVC. For *Submission 3, 4 and 5* we added a random forest model and tested different parameters (see *Table 1* for more details). We then submitted a kNN model having k = 3 (*Submission 6*) and finally, a Multinomial Naive Bayes model (*Submission 7*).

In the **VOTING** notebook we put together the predictions from all the models above to output a single prediction. To do this, we trained a Logistic Regression having the predictions of the different models as variables (*Submission 8*).

You can find the score of the different Kaggle submissions in the table below:

|  |  |  |
| --- | --- | --- |
| Submission | Model | F1 score |
| 1 | Simple logistic regression | **0,81492** |
| 2 | C-Support Vector Classification (SVC) with tuned hyperparameters using GridSearch cross-validation | **0,80470** |
| 3 | Random forest with 100 estimators and max\_depth 10’000 | **0,79754** |
| 4 | Random forest with 1000 estimators and max\_depth 100’000 | **0,80061** |
| 5 | k-Nearest Neighbors (kNN) with optimal number of neighbors (k = 3) | **0,61758** |
| 6 | Multinomial naive Bayes | **0,80265** |
| 7 | Aggregation of 5 models (LR, SVC, RF, kNN, NB) through voting, model’s influence on the vote is weighted through logistic regression | **0,80879** |
| 8 | Logistic regression having reduced duplicate rows to 1 entry, completely eliminate contradicting duplicates (same text but target is not consistent: sometimes 1, sometimes 0) | **0,81595** |

Table 1: Summary of the different submissions with their F1 score.