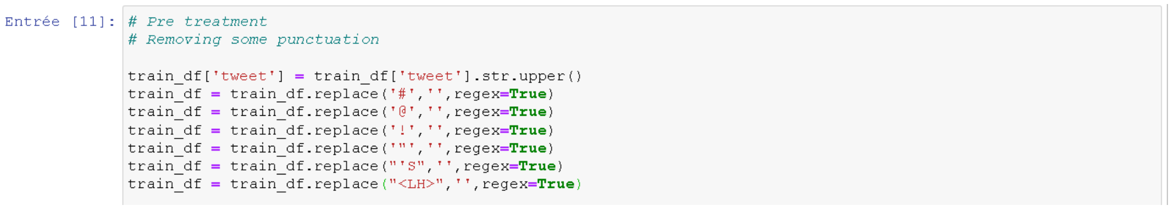
X1085032
VENANT-VALERY Thomas

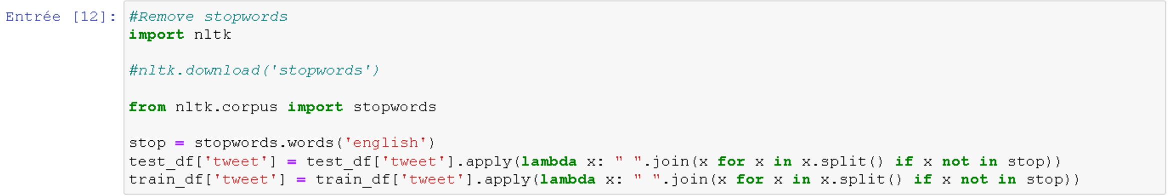
Kaggle competition report

I started by loading data and giving them the format I am going to use (pandas dataframe). Then, I did some preprocessing : upper characters, removing stopwords and special characters, transforming words to their root,…
This step can have a big impact on the result (in a good or bad way). In my experience, it was positive so I kept them. However, I did not try every combination of used/unused transformations but overall it was useful.



```
Entrée [11]: # Pre treatment
             # Removing some punctuation

             train_df['tweet'] = train_df['tweet'].str.upper()
             train_df = train_df.replace('#','',regex=True)
             train_df = train_df.replace('@','',regex=True)
             train_df = train_df.replace('!','',regex=True)
             train_df = train_df.replace('"','',regex=True)
             train_df = train_df.replace("'S",'',regex=True)
             train_df = train_df.replace("<LH>",'',regex=True)
```



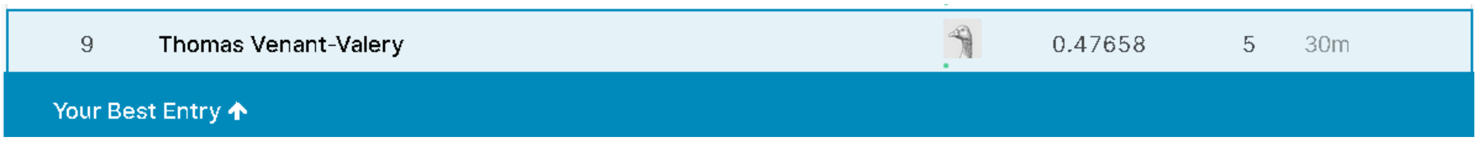
```
Entrée [12]: #Remove stopwords
             import nltk

             #nltk.download('stopwords')

             from nltk.corpus import stopwords

             stop = stopwords.words('english')
             test_df['tweet'] = test_df['tweet'].apply(lambda x: " ".join(x for x in x.split() if x not in stop))
             train_df['tweet'] = train_df['tweet'].apply(lambda x: " ".join(x for x in x.split() if x not in stop))
```

I tried different methods to recognize tweets emotions. The first one was Bernoulli classifier which gave decent results (~46%). I tried with different parameters, but in the end I choose 400000 as the max features value. I tried to increase max features value, but the impact was not enough to be worth.
I also used keras to do some neural network approach. It lasted 7 hours for 1 epoch and gave a result of 47.5%. It took so much time because I have no GPU.



9	Thomas Venant-Valery	0.47658	5	30m

I tried to do the same with 5 epochs and using class weights to impact the model. Nevertheless, I only had a result of 44% success. I suppose that my model was victim of under/overfiting.

I also tried to apply Bert to this competition, however I did not manage to make it work. It would have probably taken a lot of time, but the final result would have been better than my best score.

To conclude, my final result is decent but could have been improved a lot using GPUs and better algorithms such as Bert. It remains a good training for the final project.