

# PROJET DE CLASSIFICATION DES ARTICLES DE PRESSE



×



## Table des matières

Table des matières .....	2
Introduction .....	3
I. Définition du problème .....	3
II. Présentation du dataSet.....	3
III. Motivation du choix des technos utilisées .....	4
A. Pour la classification des thèmes des articles .....	4
B. Pour la classification de la tonalité des articles .....	4
IV. Explications des technos .....	5
A. Pour les classificateurs de thèmes: .....	5
B. Pour les classificateurs de tonalité:.....	5
V. Présentation des résultats.....	7
A. Pour les classificateurs de thèmes .....	7
B. Pour les classificateurs de tonalité.....	9
VI. Transfer Learning avec CamemBERT .....	13
VII. Conclusion et pistes d'améliorations.....	14
A. Classification des sujets des articles .....	14
B. Classification des tonalités .....	14

## Introduction

Odia est la plateforme SaaS qui transforme instantanément vos contenus numériques en contenus audio grâce au “Natural Language”. Proposez tous vos contenus en format audio en même temps que vos contenus numériques. Disposez de votre journal parlé en même temps que votre journal en format numérique, comme si celui-ci était lu par un acteur.

### I. Définition du problème

Le projet consiste à développer plusieurs modèles permettant la classification d'articles en une dizaine de sujets (politique, sport, économie, faits divers...) et la classification des phrases en 4-5 tonalités (ironique, comique, tragique, polémique...)

Pour ce qui concerne la détection du sujet de l'article, la tâche est plus ou moins réalisable. Toutefois, détecter l'ironie, par exemple, est une compétence sociale que de nombreux humains peinent eux-mêmes à maîtriser. En fait, si on revient à ces fameux analystes humains qui ont précédé l'analyse automatique de sentiment, des chercheurs se sont rendu compte que la catégorisation positif/neutre/négatif donnait lieu à différents classements des mêmes résultats en fonction de la personne en charge.

### II. Présentation de la base de données utilisée

La collecte de données est l'étape la plus importante pour résoudre tout problème d'apprentissage automatique. Votre classificateur de texte ne peut être aussi bon que l'ensemble de données à partir duquel il est construit.

Nous avons donc cherché à construire une base de données la plus riche possible en récupérant des articles de différentes manières. Notre base de données est ainsi composée de 2 bases de données trouvées en ligne :

1. Kaggle : <https://www.kaggle.com/arcticgiant/french-financial-news?select=FrenchNews.csv>
2. Webz.io : <https://webz.io/free-datasets/french-news-articles/> ;

Et d'une base de données reconstituée après avoir procédé à un web scrapping du site Le Monde entre le 1/01/2015 et le 03/01/2021 à partir de leurs archives.

Notre base de données comporte la date de publication de l'article, le journal ayant publié l'article, le lien de l'article, le titre de l'article et le contenu de l'article. Pour le site Le Monde, nous avons également récupéré le thème de l'article. Nous avons par la suite fait un découpage par phrase du contenu des articles en gardant en référence l'article d'origine et labellisé la tonalité de chaque phrase.

### III. Motivation du choix des technos utilisées

#### A. Pour la classification des thèmes des articles

Nous avons essayé plusieurs techniques permettant d'atteindre plus ou moins le même résultat et avons choisi la technique la plus précise.

Nous avons d'abord eu l'idée de créer des listes de mots par thème, puis de compter les occurrences de ces mots dans les datasets d'articles que nous avons récupérés. Le thème qui a le plus grand nombre de mots associés présents dans les articles sera le thème prédit. Cela permet d'obtenir rapidement des résultats mais cette approche est un peu fastidieuse, et elle nécessite de créer des ensembles de mots non exhaustif pour chaque thème.

Titre	Contenu	Agency	URL	testURL	mots_économie	len_économie	mots_écologie	len_écologie	mots_politique	len_politique	mots_sport	len_sport	mots_santé	len_santé	mots_technologie	len_technologie	mots_culture	len_culture	catégorie_prédite
side - une Système 10 mètres gler da	C'est une drôle de découverte qu'il faut ce...	Le Point	<a href="https://www.boursorama.com/actualite-economie/">https://www.boursorama.com/actualite-economie/</a>	C'est une drôle de découverte qu'il faut ce...		0	[pre, net, ne]	3		0		0		0		0		0	Écologie
butatut disque le art, l'auto membre du Bar...	L'émotion de cette émission nous emmène au Bar...	France 24	<a href="https://www.boursorama.com/video/actualite/">https://www.boursorama.com/video/actualite/</a>	L'émotion de cette émission nous emmène au Bar...		0		0		0		0		0		0		0	Faits divers
n Grande- tagne, les au d'être satis...	L'ONGREIS (Roulers) - Les ventes au détail en Gr...	Roulers	<a href="https://www.boursorama.com/actualite-economie/">https://www.boursorama.com/actualite-economie/</a>	L'ONGREIS (Roulers) - Les ventes au détail en Gr...	[économique, croissance, taux, inf...	5		0	[rent, (mon européen)]	2		0		0		0	[rent]	1	Économie
monisme celles les avec les moyen...	La sécurité ne fait pas partie des protégées...	Le Point	<a href="https://www.boursorama.com/actualite-economie/">https://www.boursorama.com/actualite-economie/</a>	La sécurité ne fait pas partie des protégées...		0	[protéger]	1	[sécurité, sécurité, politique, ho...	6		0		0		0	[par]	1	Politique
Droit Roulers sacré de crocs, l...	NICE (Roulers) - Le président de l'AS Mars...	Roulers	<a href="https://www.boursorama.com/actualite-economie/">https://www.boursorama.com/actualite-economie/</a>	NICE (Roulers) - Le président de l'AS Mars...		0	[responsable]	1	[Etat, Etat, ki]	3		0		0		0	[par]	1	Politique
monde et Moi à val pour les moyen de...	PARIS (Roulers) - Les agents de la police de s...	Roulers	<a href="https://www.boursorama.com/actualite-economie/">https://www.boursorama.com/actualite-economie/</a>	PARIS (Roulers) - Les agents de la police de s...		0	[responsable]	1	[gouvernement, Etat, sécurité, sécurité, mens...	6		0		0		0	[par]	1	Politique
Laifame, des ven toutes en cendres	6000. C'est le nombre de maisons qui ont été l...	France 24	<a href="https://www.boursorama.com/video/actualite/">https://www.boursorama.com/video/actualite/</a>	6000. C'est le nombre de maisons qui ont été l...		0		0		0		0		0		0	[par]	1	Culture
Les notables proposent monde le gloph...	PARIS (Roulers) - Les députés européens de la ...	Roulers	<a href="https://www.boursorama.com/actualite-economie/">https://www.boursorama.com/actualite-economie/</a>	PARIS (Roulers) - Les députés européens de la ...		0		0	[Etat, Etat, Etat, député, député, par...	7		0	[santé]	1		0		0	Politique
onde des sieurs des disque les sagittat...	D'un côté, les policiers de l'Office central p...	Le Point	<a href="https://www.boursorama.com/actualite-economie/">https://www.boursorama.com/actualite-economie/</a>	D'un côté, les policiers de l'Office central p...		0		0	[gouvernement, mensures, mensures...	7		0		0		0		0	Politique
le pastre le Zender	Alice Zender est venue rendre à l'occasion...	France 24	<a href="https://www.boursorama.com/video/actualite/">https://www.boursorama.com/video/actualite/</a>	Alice Zender est venue rendre à l'occasion...		0		0		0		0		0		0	[roman, roman, roman]	3	Culture
elle - vers monde de monde et d...	Alors que la question de l'indépendance de la ...	France 24	<a href="https://www.boursorama.com/video/actualite/">https://www.boursorama.com/video/actualite/</a>	Alors que la question de l'indépendance de la ...		0		0		0		0		0		0	[par]	1	Culture

Nous nous sommes donc tournés vers l'algorithme de classification **CamemBERT** et en entraînant le modèle sur les données récupérées sur le site Le Monde. Cet algorithme diffusé en 2020 est la version française du modèle RoBERTa sorti en 2019, lui-même inspiré du modèle Bert. C'est un algorithme contextuel capable de détecter les noms propres et la fonction de chaque mot. A la fin de l'apprentissage, chaque mot est représenté par une suite de 768 vecteurs. Donc pour un même mot, ces vecteurs seront différents en fonction de chaque phrase, contexte différent.

#### B. Pour la classification de la tonalité des articles

Nous avons choisi Countvectorizer car il permet d'étiqueter facilement une collection de documents textuels, de générer un glossaire de mots connus et d'encoder de nouveaux documents à l'aide du glossaire, et ce, de manière rapide et plus ou moins précise. Ce qui va surtout jouer, c'est la qualité de la base de données et de la labellisation.

Un point très important de l'algorithme est qu'il a fallu établir des règles sur la labellisation et sur la qualité des phrases 'parsées'. Pour savoir ce à quoi une phrase ironique ou polémique

ressemble, il a fallu que nous passions en revue certaines phrases et que nous nous mettions d'accord sur le sens de chacun des labels.

L'apprentissage du modèle en utilisant countvectorizer est très rapide (quelques secondes), ce qui permet de tester différents datasets et différentes approches (quels labels utiliser) et de nous adapter par la suite, voilà une des raisons pour lesquelles nous avons retenu ce modèle. Nous avons ainsi pu mieux choisir nos labels (le label "neutre" a été supprimé et le label "joyeux" a été ajouté par la suite).

Countvectorizer permet de configurer les paramètres pour obtenir des résultats plus intéressants. Par exemple, nous avons choisi de prendre en compte les unigrams et les bigrams, et de ne retenir que les mots qui sont présents un nombre de fois suffisant. Cela permet d'avoir un modèle plus précis

#### **IV. Explications des technos**

##### **A. Pour les classificateurs de thèmes :**

Nous avons utilisé le modèle pré-entraîné, CamenBERT base avec une architecture de 12 couches, 768 dimensions cachées et 12 têtes d'attention soit 110M de paramètres.

Modèle fine-tuné testé sur le titre et le titre plus le chapo. Les meilleurs résultats ont été obtenus sur le modèle titre-chapo.

Au préalable, lors du pré-processing, des regroupements des catégories ont été effectués afin d'obtenir des classes homogènes et en nombre suffisants.

Voici les catégories utilisées :

- 0 : international
- 1 : culture
- 2 : économie
- 3 : numérique/environnement
- 4 : politique/société

Les différents modèles ont été testés sur les mêmes données suivantes :

- Entraîné sur une base de 4000 articles, 1000 par catégorie.
- Base de données de validation de 1000 articles, 200 par catégorie

##### **B. Pour les classificateurs de tonalité :**


Une classification par analyse des sentiments nous permet d'identifier la polarité d'un texte : le type d'opinion qu'il exprime par exemple ou sa tonalité. Cela peut prendre la forme d'une

évaluation binaire comme ironique/non ironique, ou un ensemble d'options, comme une évaluation par étoiles de 1 à 5.


Nous avons choisi d'appliquer un modèle de machine learning reposant sur la bibliothèque Sklearn. Nous utilisons CountVectorizer comme technologie pour vectoriser les mots de nos textes et les classes que nous leur avons attribuées. La diversité des sources de nos textes dans la base de données que nous avons préparée permet une richesse dans notre vectorizer qui permet en fin de compte de prédire avec précision plutôt bonne la tonalité de la phrase que nous entrons en prédiction. La base de données est entrée dans l'interface prévu à cet effet (en lançant la commande "streamlit run streamlit\_training.py" puis en entrant la base de données sous format Excel). Le modèle est directement produit et sauvegardé ("model/vectorizer.pkl") et prêt à être réutilisé.

### Fichier d'entraînement

Drop a csv file to train on:

 Drag and drop file here  
Limit 200MB per file • XLSX

Browse files

 Labellisation ton - avec Libé sans ton neutre.xlsx 103.4KB

×

	URL	phrase	Fonction	Ironique	Comique
0	https://www.boursorama...	Prostitution étudiante: le ...	titre	0	1
1	https://www.boursorama...	Hey les étudiant(e)s. Rom...	paragraphe	0	1
2	https://www.boursorama...	Des hommes et des femm...	paragraphe	0	0
3	https://www.boursorama...	Un appel à la prostitution	paragraphe	0	0
4	https://www.boursorama...	Chaque jour à Rio, on dén...	paragraphe	0	0

- **Features:** phrase
- **Targets:** ['Ironique', 'Comique', 'Tragique', 'Polémique', 'Joyeux']

Data processed.

Mean cross-validated AUC score: 0.6178766359705646

Models trained and saved.

Après avoir lancé la commande "streamlit run streamlit\_predict.py", on peut directement évaluer une phrase.

```

src > ud > streamlit_prediction.py 3.1 M x  model_honique.pkl U  global_variables.py M
1 import glob
2
3 import numpy as np
4 import pandas as pd
5 import plotly.express as px
6 import plotly.graph_objects as go
7 import streamlit as st
8
9 from src.conf.global_variables import MODELS_PATH, LABELS
10 from src.models.models_interpretation import get_local_weights_df, get_global_weights_df
11 from src.models.models_predict import get_prediction
12 from src.models.models_save_and_load import load_models
13
14 models_filenames = glob.glob(str(MODELS_PATH / 'model_*.pkl'))
15 vectorizer_filename = MODELS_PATH / 'vectorizer.pkl'
16
17 st.title("Catégorisation de la tonalité d'une phrase")
18 classifiers, vectorizer = load_models(models_filenames, vectorizer_filename)
19
20 input_sentence = st.text_input("Entrez une phrase en français, le modèle étant préentraîné sur le dataset des tons liés à des phrases d'articles de presse")
21 st.markdown("La probabilité donnée par chaque régression logistique donne des pourcentages d'appartenance à chaque classe. Il y a possibilité d'afficher les odds pour mieux comprendre comment les poids font évoluer la prédi")
22 st.latex(r"odds=\frac{P(y=1)}{P(y=0)}")
23 preds, test_term_doc = get_prediction(input_sentence, vectorizer, classifiers, LABELS)
24 preds_df = pd.DataFrame({'label': LABELS, 'preds': preds[0] * 100})
25 odds_df = pd.DataFrame({'label': LABELS, 'odds': preds[0] / (1 - preds[0])})
26 is_odds = st.checkbox("Afficher les odds", key='is_odds')
27 if not is_odds:
28     fig = px.bar(preds_df,
29                 x='preds',
30                 y='label',
31                 color='preds',
32                 range_x=[0, 100],
33                 orientation='h',
34                 range_color=[0, 100],
35                 color_continuous_scale='Reds',
36                 title="Probabilité d'appartenance au label")
37 else:
38     fig = px.bar(odds_df,
39                 x='odds',
40                 y='label',
41                 color='odds',
42                 orientation='h',
43                 color_continuous_scale='Reds',
44                 title="Odds d'appartenance au label")

```

Le premier graphique donne le pourcentage de chance pour que la phrase appartienne à chacune des tonalités. Il faut regarder la tonalité qui a le meilleur score (cf. Présentation des résultats). Plus bas, on peut voir quels mots ont joué dans la prédiction et leur poids. La phrase est récupérée et chaque mot est analysé. Il en ressort une probabilité totale pour que la phrase soit classée dans chaque tonalité.

```

def get_prediction(sentence, vectorizer, classifiers, labels):
    x_test = vectorizer.transform(pd.Series(sentence))
    preds = np.zeros((x_test.shape[0], len(labels)))
    if not is_vectorized_sentence_empty(x_test):
        for idx, label in enumerate(labels):
            preds[:, idx] = classifiers[label].predict_proba(x_test)[:, 1]
    return preds, x_test

```

## V. Présentation des résultats

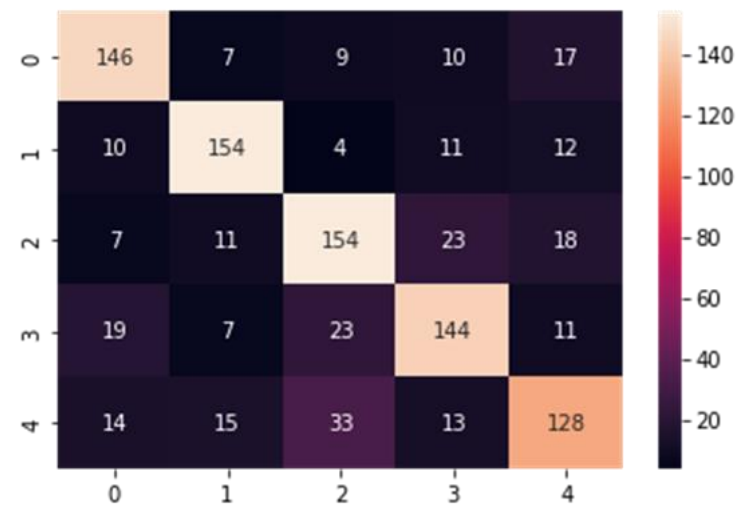
### A. Pour les classificateurs de thèmes

Nous avons d'abord lancé le modèle en prenant en entrée que le titre de l'article. Voici, les résultats sur la phase d'apprentissage du modèle :



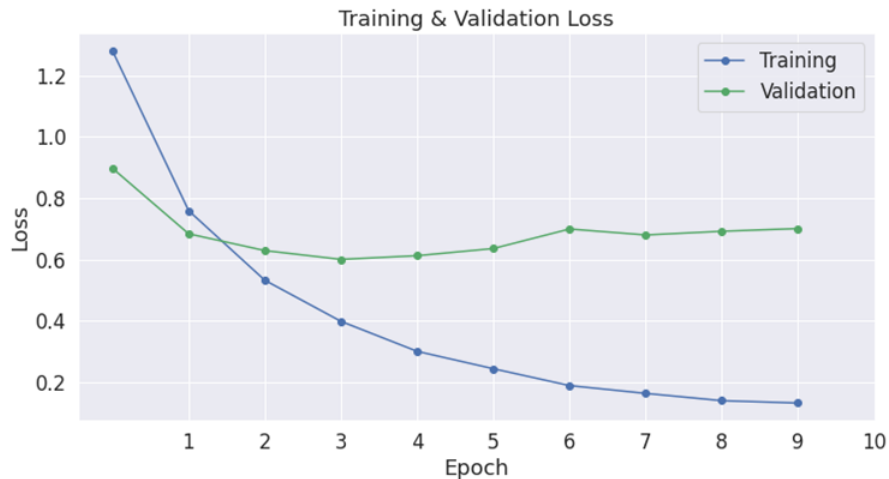
Et voici les résultats sur la phase de validation :

	precision	recall	f1-score	support
0	0.74	0.77	0.76	189
1	0.79	0.81	0.80	191
2	0.69	0.72	0.71	213
3	0.72	0.71	0.71	204
4	0.69	0.63	0.66	203
accuracy			0.73	1000
macro avg	0.73	0.73	0.73	1000
weighted avg	0.73	0.73	0.73	1000



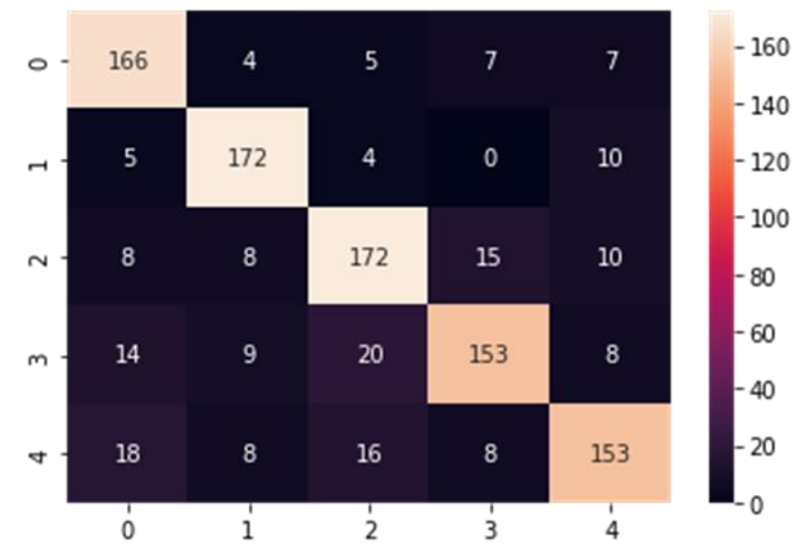
En lançant le modèle sur le titre et le chapo des articles, nous obtenons pour la phase d'apprentissage :





Et pour la phase de validation :

	precision	recall	f1-score	support
0	0.79	0.88	0.83	189
1	0.86	0.90	0.88	191
2	0.79	0.81	0.80	213
3	0.84	0.75	0.79	204
4	0.81	0.75	0.78	203
accuracy			0.82	1000
macro avg	0.82	0.82	0.82	1000
weighted avg	0.82	0.82	0.82	1000



## B. Pour les classificateurs de tonalité

Voici les prédictions que nous obtenons pour différentes phrases :

Entrez une phrase en français, le modèle étant préentraîné sur le dataset des tons liés à des phrases d'articles de presse

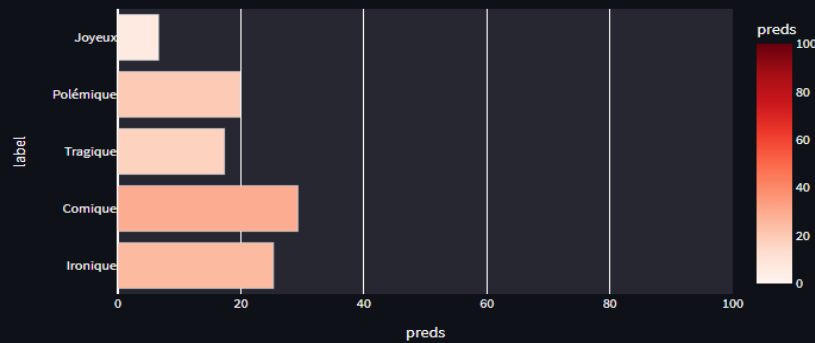
être français, c'est mettre le lait avant les céréales dans son bol

La probabilité donnée par chaque régression logistique donne des pourcentages d'appartenance à chaque classe. Il y a possibilité d'afficher les odds pour mieux comprendre comment les poids font évoluer la prédiction.

$$odds = \frac{P(y = 1)}{P(y = 0)}$$

☐ Afficher les odds

Probabilité d'appartenance au label



Entrez une phrase en français, le modèle étant préentraîné sur le dataset des tons liés à des phrases d'articles de presse

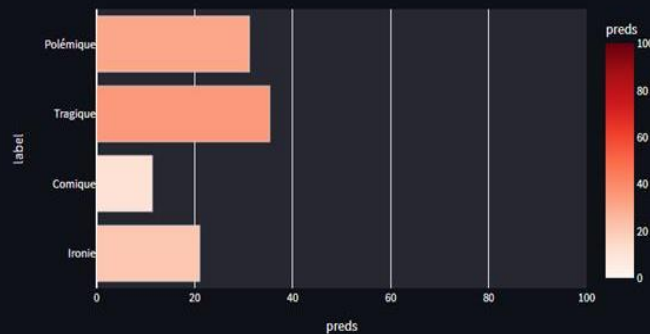
dix personnes tuées par un meurtrier multirécidiviste

La probabilité donnée par chaque régression logistique donne des pourcentages d'appartenance à chaque classe. Il y a possibilité d'afficher les odds pour mieux comprendre comment les poids font évoluer la prédiction.

$$odds = \frac{P(y = 1)}{P(y = 0)}$$

☐ Afficher les odds

Probabilité d'appartenance au label



Entrez une phrase en français, le modèle étant préentraîné sur le dataset des tons liés à des phrases d'articles de presse

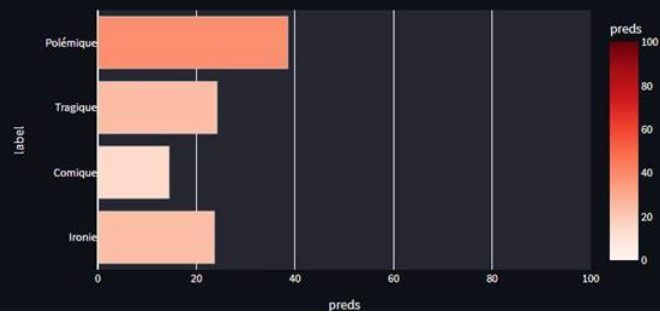
polanski accusé de viol sur mineur

La probabilité donnée par chaque régression logistique donne des pourcentages d'appartenance à chaque classe. Il y a possibilité d'afficher les odds pour mieux comprendre comment les poids font évoluer la prédiction.

$$odds = \frac{P(y = 1)}{P(y = 0)}$$

☐ Afficher les odds

Probabilité d'appartenance au label



Entrez une phrase en français, le modèle étant préentraîné sur le dataset des tons liés à des phrases d'articles de presse

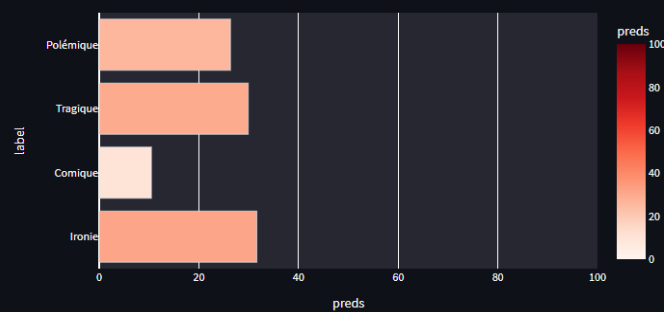
oh comme c'est bizarre!

La probabilité donnée par chaque régression logistique donne des pourcentages d'appartenance à chaque classe. Il y a possibilité d'afficher les odds pour mieux comprendre comment les poids font évoluer la prédiction.

$$odds = \frac{P(y = 1)}{P(y = 0)}$$

☐ Afficher les odds

Probabilité d'appartenance au label



Entrez une phrase en français, le modèle étant préentraîné sur le dataset des tons liés à des phrases d'articles de presse

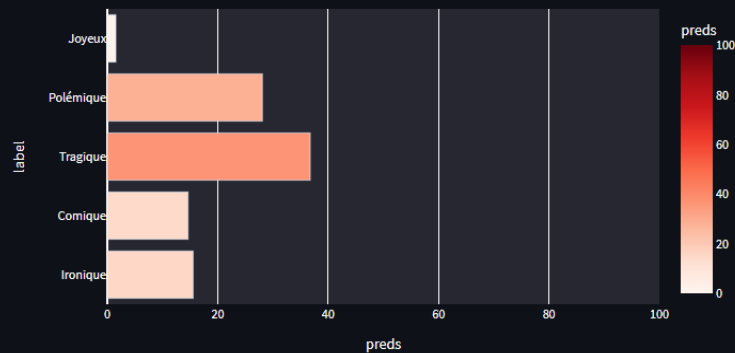
accident grave de voiture qui a provoqué la mort de plusieurs personnes

La probabilité donnée par chaque régression logistique donne des pourcentages d'appartenance à chaque classe. Il y a possibilité d'afficher les odds pour mieux comprendre comment les poids font évoluer la prédiction.

$$odds = \frac{P(y = 1)}{P(y = 0)}$$

☐ Afficher les odds

Probabilité d'appartenance au label



Entrez une phrase en français, le modèle étant préentraîné sur le dataset des tons liés à des phrases d'articles de presse

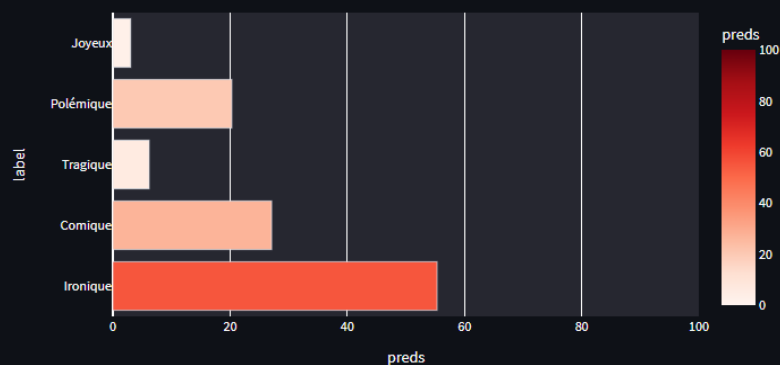
Sur son lit de mort, il se dit qu'il a vraiment bien bossé pour son entreprise

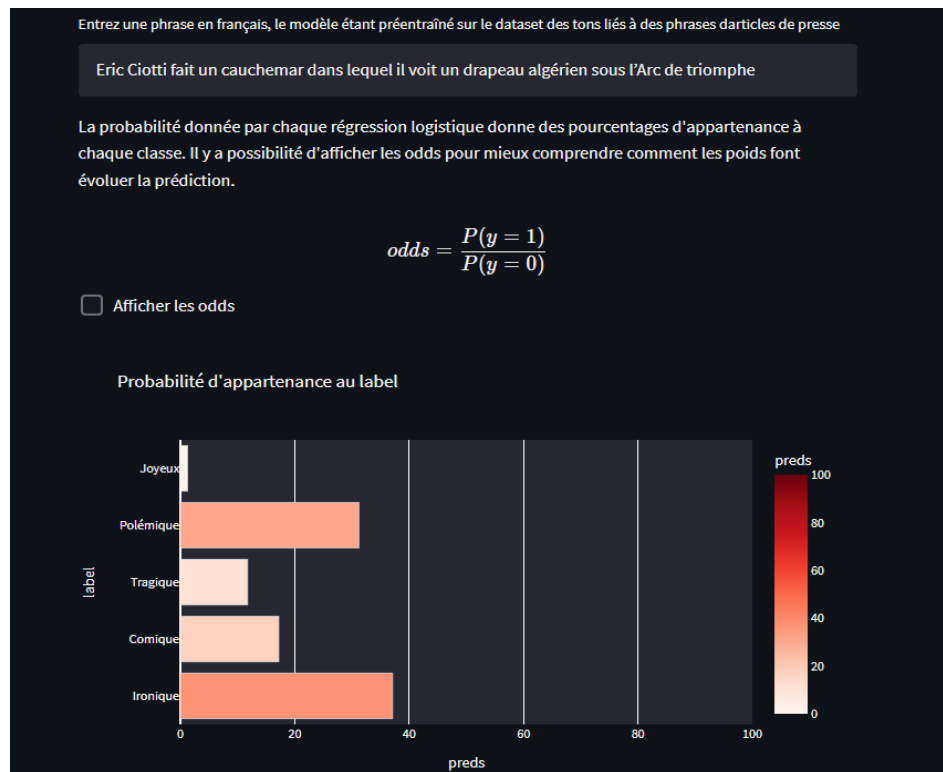
La probabilité donnée par chaque régression logistique donne des pourcentages d'appartenance à chaque classe. Il y a possibilité d'afficher les odds pour mieux comprendre comment les poids font évoluer la prédiction.

$$odds = \frac{P(y = 1)}{P(y = 0)}$$

☐ Afficher les odds

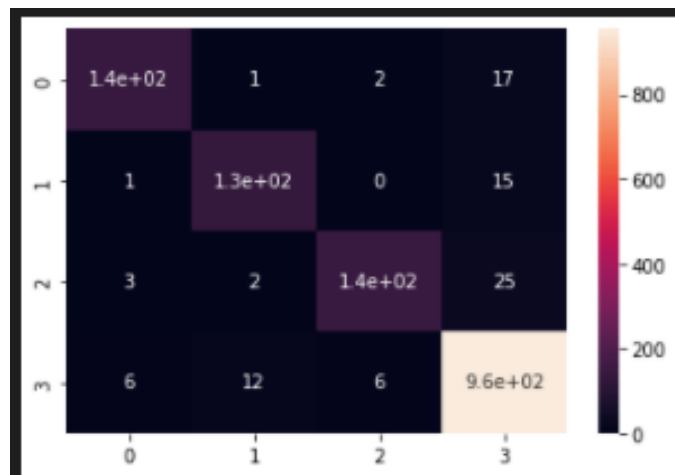
Probabilité d'appartenance au label





## VI. Transfer Learning avec CamemBERT

Nous avons aussi étudié la possibilité du transfert learning en utilisant CamemBERT. Nous allons juste un petit peu le modifier et le finetuner. Nous avons donc importé le modèle camembert, cependant nous avons changé la dernière couche qui habituellement mène vers 2 neurones soit positif soit négatif. Nous avons donc changé la structure de la dernière couche afin d'avoir 5 sorties à la place de 2 et nous avons ré-entraîné le modèle en se basant uniquement sur la phrase. De cette manière, nous obtenons d'excellents résultats.



Cependant ce constat est à nuancer en effet sur une base de données ayant peu de ressemblance avec celui de départ, différence de plume importante entre rédacteurs de différents journaux rend ce résultat nettement moins probant.

## **VII. Conclusion et pistes d'améliorations**

### **A. Classification des sujets des articles**

Les résultats sont satisfaisants en sélectionnant le modèle le plus performant basé sur le titre et le chapo avec des F1\_Score oscillant autour de 0,8, ce qui est encourageant sur un si faible volume de données. C'est pourquoi, il faudrait bénéficier de plus de GPU pour le faire tourner sur plus de données, en sachant que la base de données extrait du monde contient au moins 10 000 articles par catégorie.

Par ailleurs, une GridSearch peut être implémentée pour les paramètres de l'optimiseur Adam et d'autres modèles de bases pré-entraînés peuvent être tester comme le CamenBERT Large.

### **B. Classification des tonalités**

Sur des bases de données où on labellise les phrases "neutres", on remarque que les résultats étaient moins évidents. En effet, en fonction de l'interprétation des personnes, tout et rien ne peut tomber sous l'étiquette "neutre". En retirant le label "neutre", nous obtenons de meilleurs résultats. De ce constat, il est évident qu'il faut avoir des labels plus ou moins bien définis pour que le modèle puisse apprendre correctement.