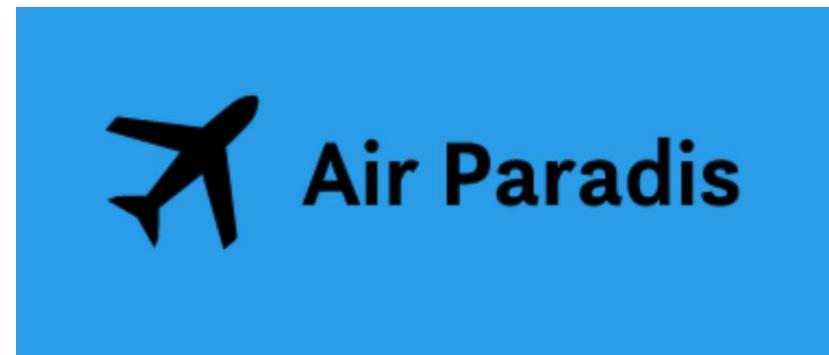


# Déetecter les Bad Buzz grâce au Deep Learning





# Sommaire

PARTIE 1 – INTRODUCTION

PARTIE 2 – EXPLORATION et PREPROCESSING

PARTIE 3 – MODÈLE SIMPLE

PARTIE 4 – MODÈLE AVANCÉ

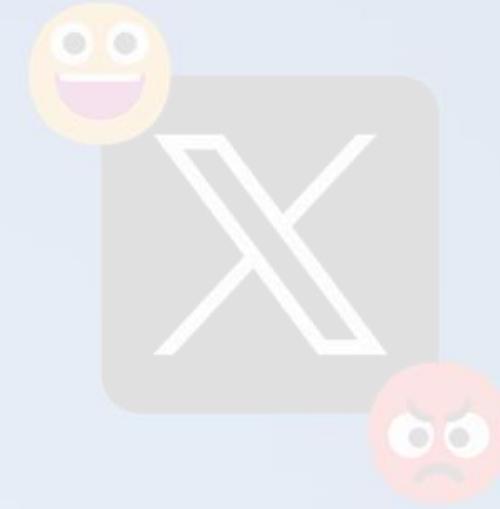
PARTIE 5 – TESTS AUTRES SOLUTIONS et BILAN

PARTIE 6 – DÉPLOIEMENT API et UI

CONCLUSION

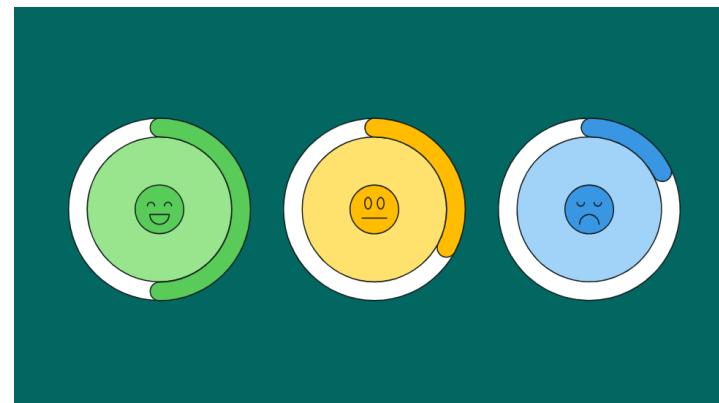


# PARTIE 1 – INTRODUCTION



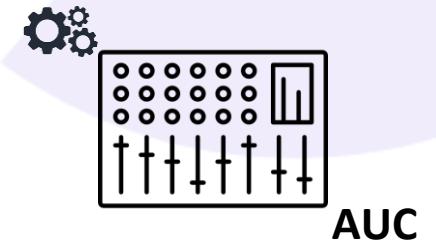
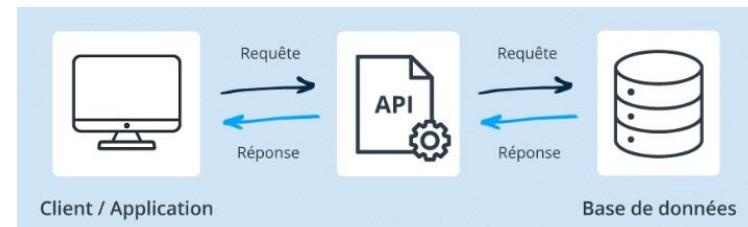
# Le projet de l'entreprise

- Produit IA pour anticiper les bad buzz sur les réseaux sociaux
  - → enjeu court terme de communication en cas de crise
  - → enjeu long terme pour comprendre comment la marque est perçue en ligne
  - → prendre des décisions basées sur les données pour améliorer l'image de la marque



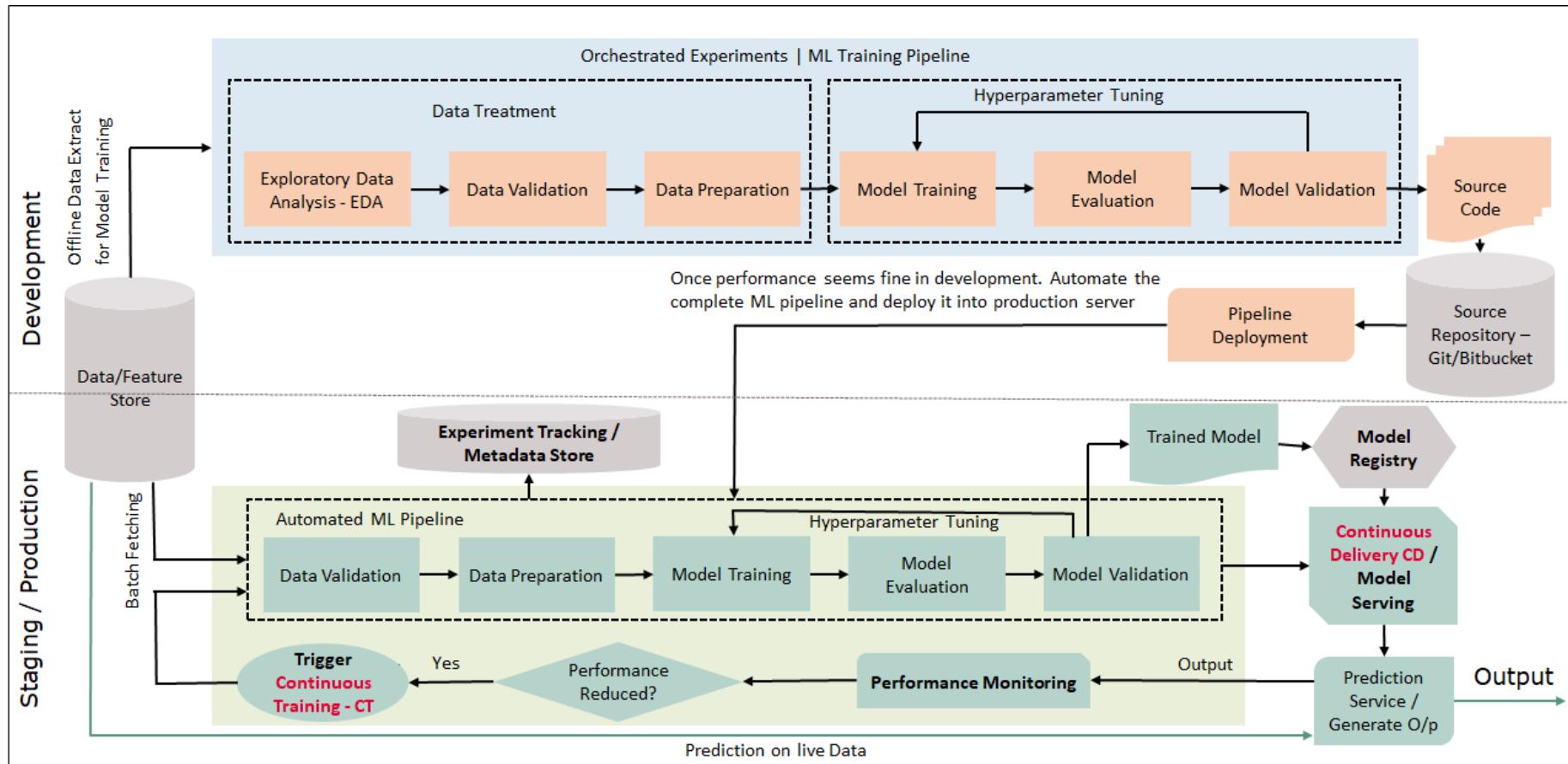
# Notre réponse

- Tests de plusieurs approches de modélisation :
  - Modèle sur mesure simple, régression logistique
  - Modèle sur mesure avancé, LSTM bidirectionnel
- Optimisation des modèles :
  - Méthodes de nettoyage
  - Méthodes de normalisation
  - Méthodes de plongement de mots (word embedding)
  - Hyperparamètres du modèle lui-même
- Création d'une API
- Création d'une UI



# Notre réponse

- MLOps :



# Notre réponse

- MLOps pour ce projet :
  - **mlflow** pour :
    - tracker les **hyperparamètres** de nos différents modèles
    - tracker les **métriques**
    - **enregistrer les modèles** que nous souhaitons conserver
    - **charger le modèle** gagnant et faire des prédictions
    - tester le service de *serving*
  -  **GitHub** pour l'**intégration continue** du code : <https://github.com/ClementPatin/OC-P7>
  -  **docker** pour créer des **conteneurs** exécutables pour notre **API** et notre **UI**
  -  **Azure WebApp Services** pour le **déploiement**
  -  **GitHub Actions** pour le **déploiement continu**



# PARTIE 2 – EXPLORATION et PREPROCESSING



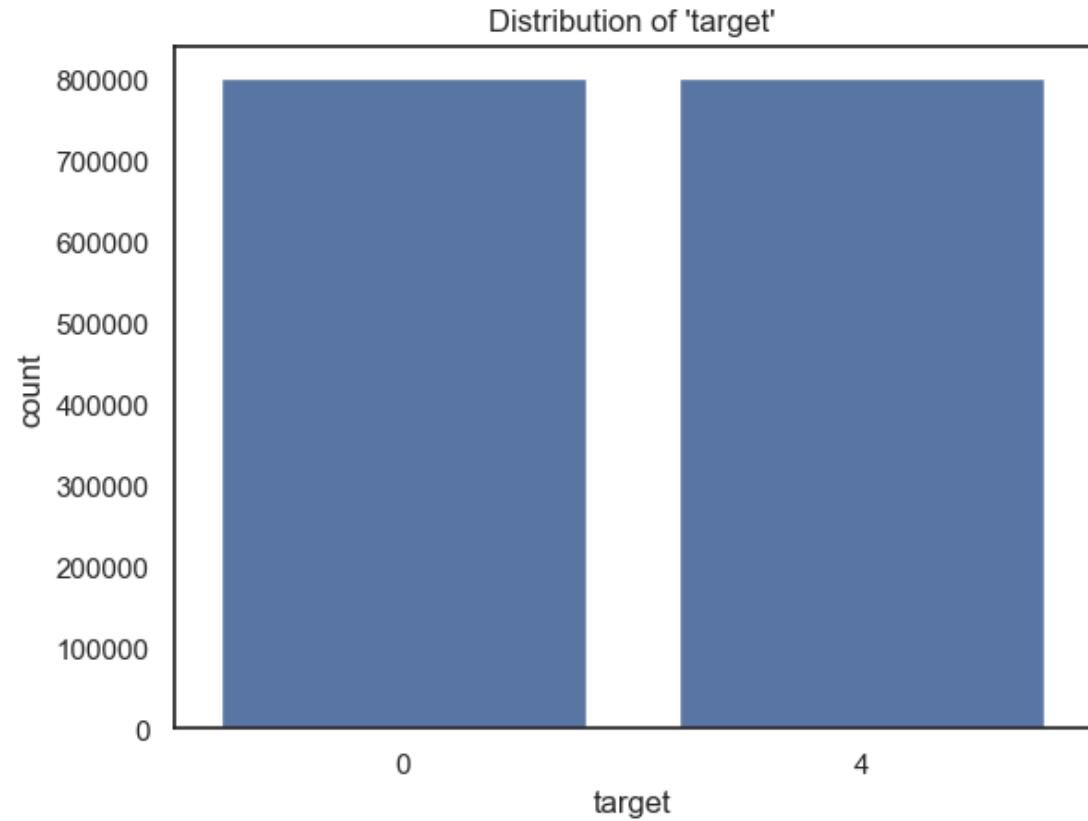
# EDA & nettoyage

## Les Doublons

	target	text
271472	0	David must be hospitalized for five days end...
271505	0	→ David must be hospitalized for five days end...
1132076	4	bathroom is clean..... now on to more enjoya...
1132074	4	bathroom is clean..... now on to more enjoya...
540159	0	#IMISSCATH #IMISSCATH #IMISSCATH #IMISSCATH #...
...	...	...
458170	0	zofran is not making this work day any easier,...
464810	0	~ get SUICIDE !!
463926	0	~ get SUICIDE !!
1170868	4	~~~XOXO~~~!!!XOXO!!! I love you!!! &#...
238478	0	~~~XOXO~~~!!!XOXO!!! I love you!!! &#...

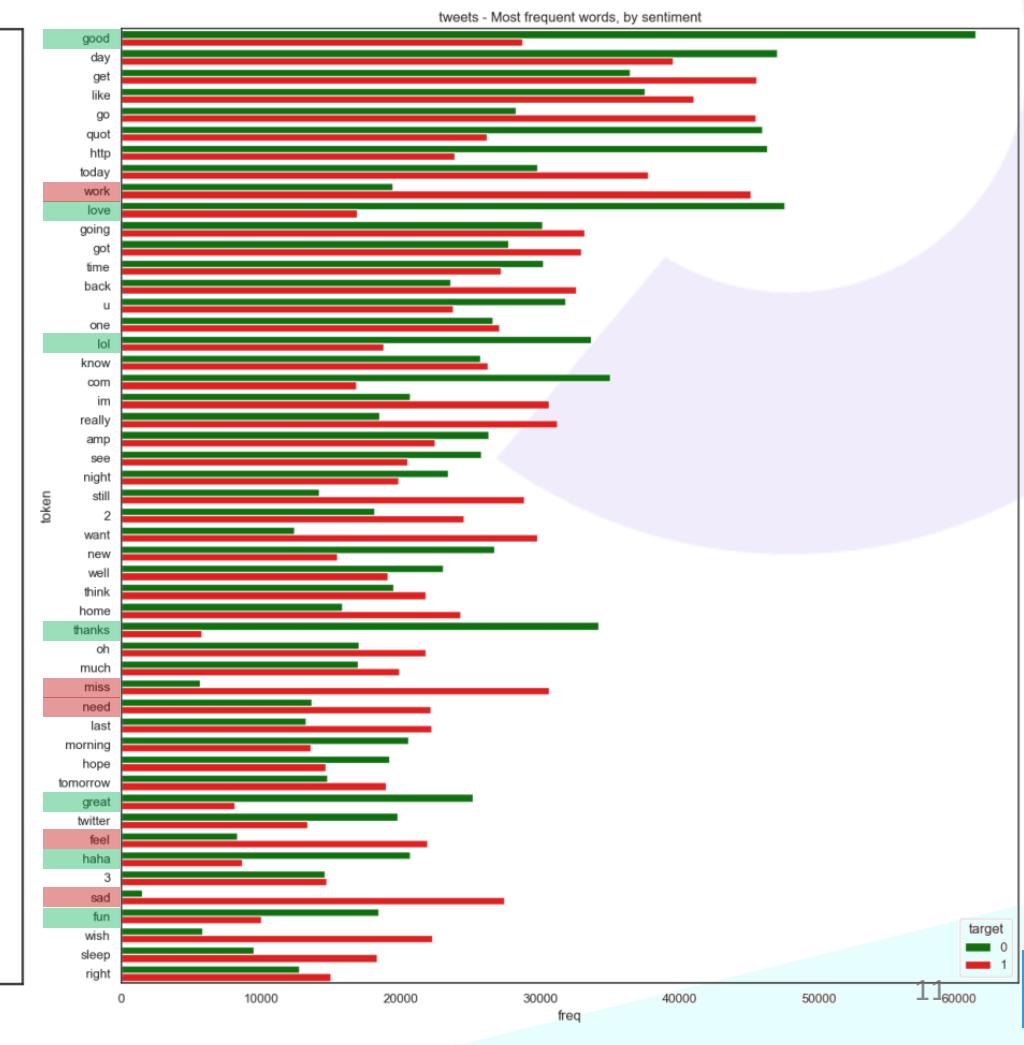
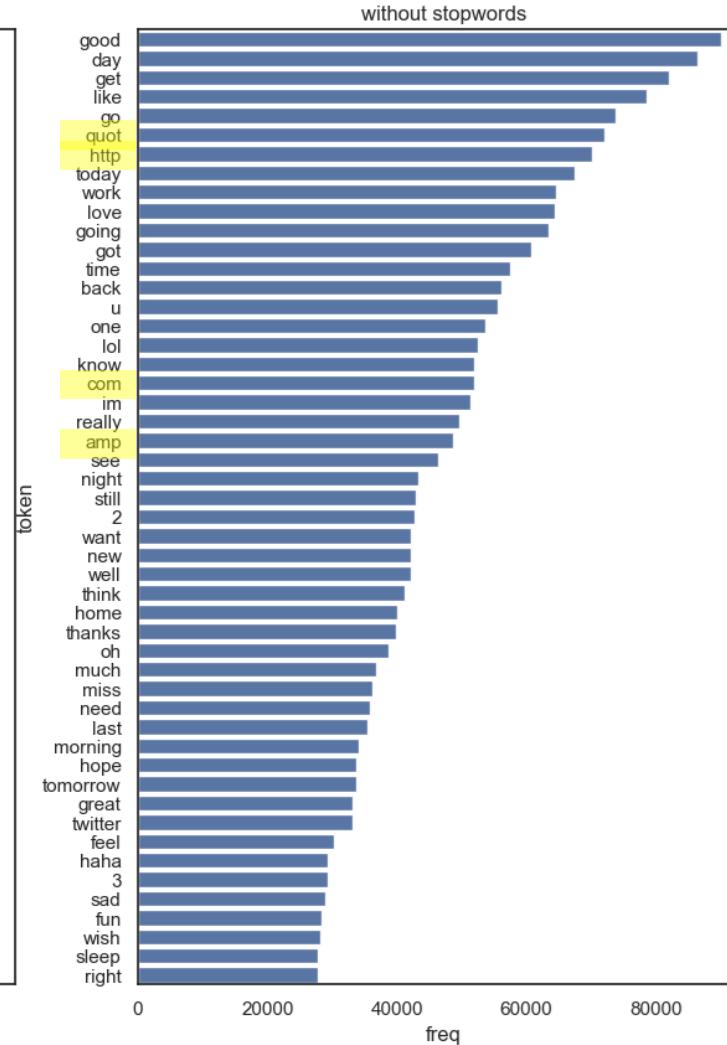
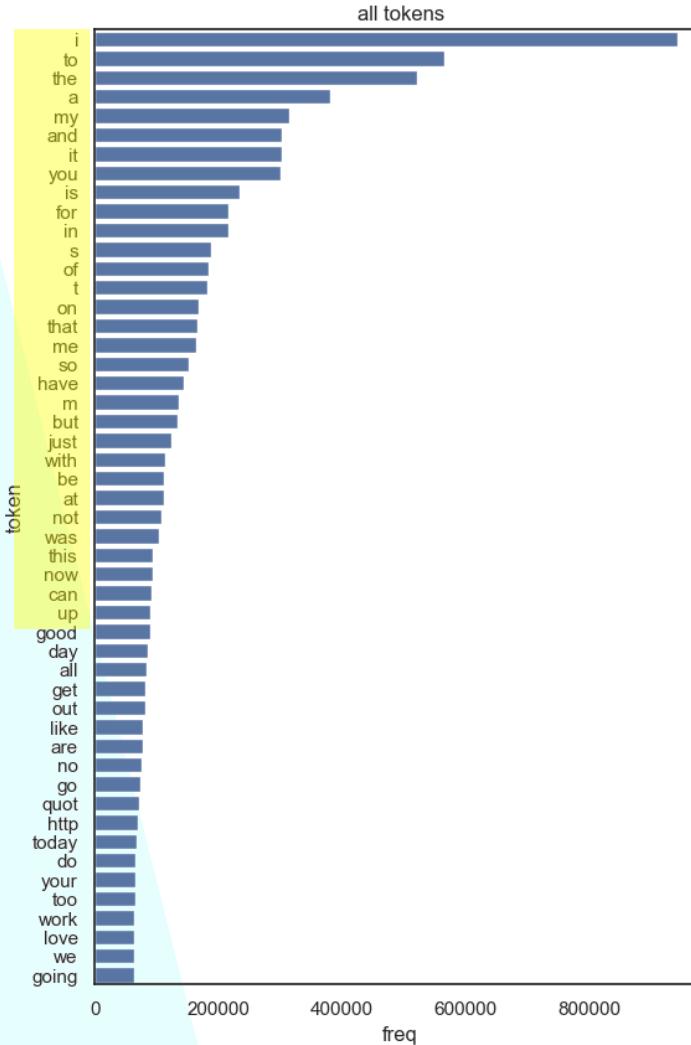
26968 rows × 2 columns

## La target



# EDA – les mots les plus fréquents

tweets - Most frequent words



# EDA & nettoyage

## ASCII non « imprimables »

target		text	NEW_len_text
1405400	0	shagÃ¼er	9
1459484	0	HallÃ¶le	9
638043	1	workÃ±ng	9
1211495	0	I â¤ you	11
1304341	0	Rock RoÃ§a	11
...	...	...	...
650945	1	human shield à®¤à®©à~?à®± à®¤à~¤à®±à~?à®¤à~¤à®...	348
380539	1	@neospace à¹¤à,fà,à¹¤à,à,¤à,¤à,±à¹¤à,¤à,¤à,f...	359
1582941	0	5 days till new top gear ??¤??¤?¤?¤?¤?¤?¤?...	359
1484882	0	@iannnnn à¹¤à,¤à,µà¹¤à,¢à,§à,à,µà,?à,¤à,±à,?à...	369
1295281	0	@catfish_ohm à¹¤à,¤à¹¤à,²à¹¤à,«à¹¤à,?à,²à,£à¹¤...	374

## Les urls

http://www.tongits.net, http://twitpic.com/7fdy  
http://bit.ly/VwV6H http://tweet.sgI  
http://twitpic.com/5nw9a http://twitpic.com/bv57d http://bit.ly/g8SD0  
http://twitpic.com/4ncls http://twitpic.com/tev20 http://twitpic.com/6fwg3  
http://twitpic.com/3knj http://twitpic.com/5f798 http://twitpic.com/6g790  
http://twitpic.com/12Korq http://twitpic.com/5cmep http://twitpic.com/6g55n  
http://twitpic.com/4cyku http://twitition.com/f5iae http://twitpic.com/6gq1om  
http://twitpic.com/etd1 http://twitpic.com/4tzo7 http://twendz.com/  
http://twitpic.com/6uohs http://twitpic.com/3pf7v http://twitpic.com/6f966  
http://twitpic.com/7e07t http://twitpic.com/67uc7 http://twitpic.com/66hb  
http://twitpic.com/67uc7 http://tinyurl1.com/c44c6r http://twitpic.com/65t98  
http://twitpic.com/30l72 http://twitpic.com/7kf4b http://bit.ly/EUNLs http://bit.ly/152cv  
http://twitpic.com/7kf4b http://twitpic.com/6f16k http://bit.ly/votetom http://mattproject.webs.com/index.htm  
http://twitpic.com/6sich http://tinyurl1.com/ry9wap http://twitpic.com/6n1p  
http://twitpic.com/6sich http://twitpic.com/50oze http://twitpic.com/4gnlp http://twitpic.com/6t6  
http://twitpic.com/6sich http://twitpic.com/6dvj4 http://twitpic.com/2y606  
http://twitphoto.com http://twitpic.com/6dvj4 http://twitpic.com/6010 http://twitpic.com/67a0  
http://twitphoto.com http://twitpic.com/6dvj4 http://twitpic.com/6dvj4 http://twitpic.com/67a0  
http://twitter.com/public\_timeline http://twitpic.com/6sjsm http://twitpic.com/6g3tt  
http://promotion.itagg.com http://twitpic.com/5dca http://twitpic.com/6g3tt  
http://bit.ly/anQju http://bit.ly/PmvRY http://short\_to/1cp2  
http://bit.ly/be9nb http://twitvalue.com/6mek http://twitpic.com/624qo http://twitpic.com/67hac  
http://twitpic.com/67wx http://twitpic.com/5656 http://bit.ly/5NvNvY  
http://twitpic.com/67wx http://twitpic.com/6463 http://bit.ly/MWx6  
http://twitpic.com/67wx http://twitpic.com/6446 http://bit.ly/6rrtf

# EDA & nettoyage

# Les adresses mail

## Les caractères répétés



# EDA & nettoyage

# Les contractions

The image is a dense word cloud centered around the word "gonna". Other prominent words include "I", "you", "ain't", "it", "don't", "they're", "haven't", "wanna", "til", "ur", and "wASN'T". The words are rendered in a variety of sizes and colors, including shades of blue, yellow, green, and red, set against a plain white background.

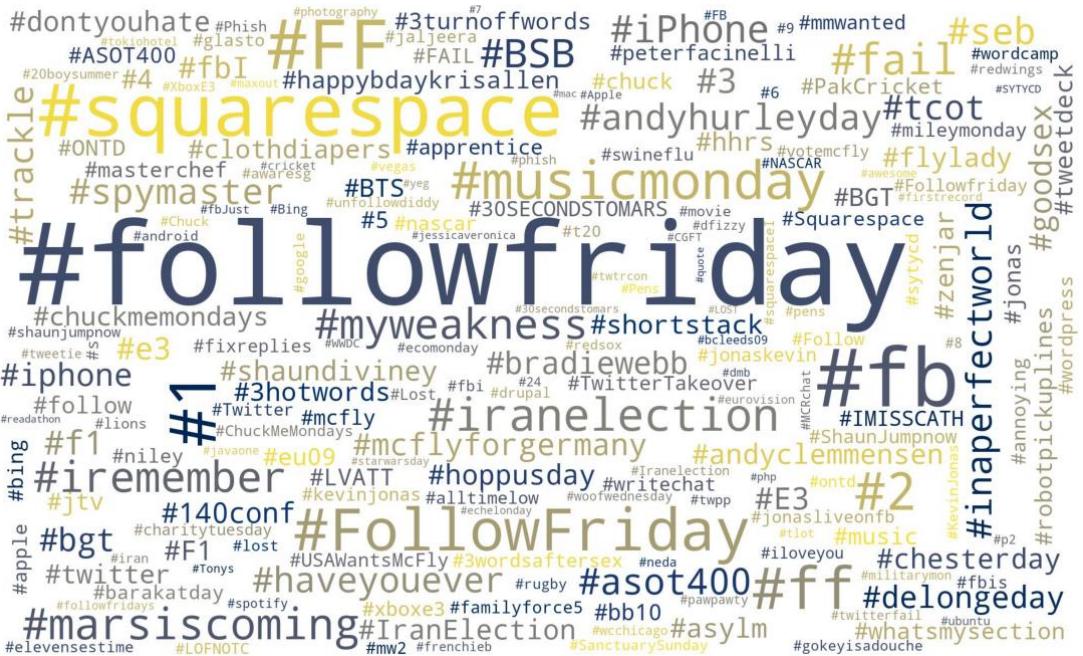
# Les emoticons

This chart illustrates a wide range of emoji faces, each accompanied by a phonetic transcription and a brief description of its meaning.

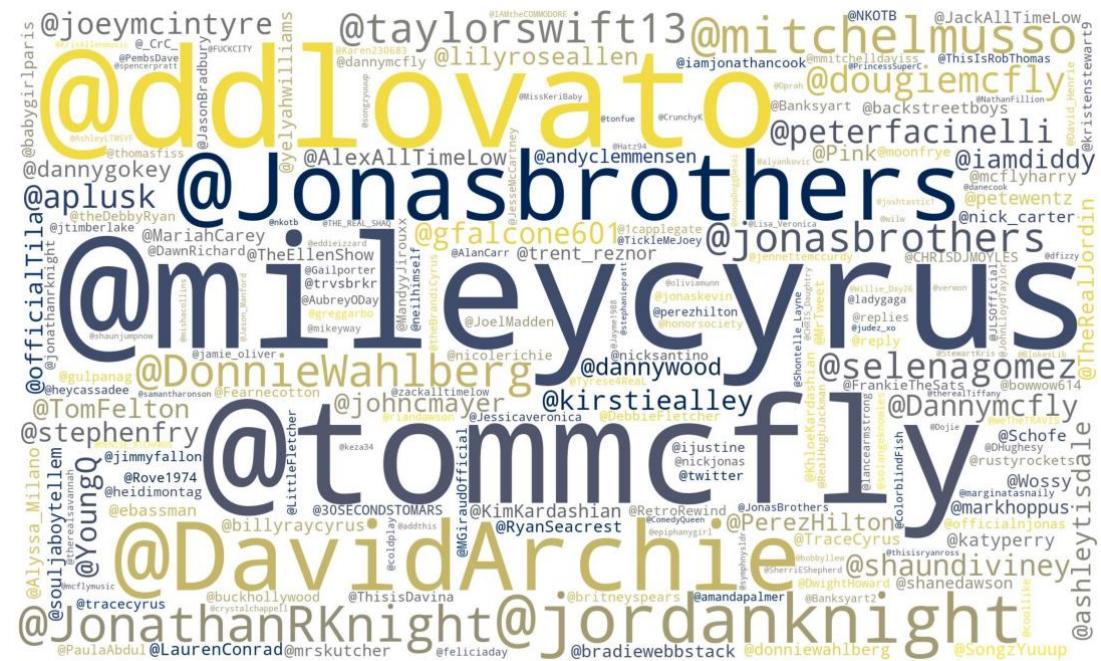
- XD - Laughing**: ] - Happy
- D: - Sadness**: | - Straight
- :[ - Frown**:o,0 - Surprised
- :o - Surprise**:o - Frown
- :; - Wink**:c - Wink
- : / - Skeptical**:\* - Kiss
- ; ) - Wink**:o) - Happy
- o,o - Surprised**:o,o - Great
- o: - Sealed**:D - Wink
- ;; - Sad**:D - Great
- =) - Happy**:b - Tongue
- : - Great**:b - Tongue
- ;) - Happy**:b - Tongue
- :\$ - Embarrassed**:b - Tongue
- :'( - Crying**:S - Skeptical
- :T.T - Sad**:S - Break
- DX - Great**:/ - Skeptical
- :x - Sealed**:& - Sealed
- =D - Laughing**:3 - Angel
- >3 - Laughing**:3 - Angel

# EDA & nettoyage

# Les hashtags



## Les mentions



# Nettoyage personnalisé

## Modèle simple

- Les étapes précédentes
- Stopwords
- Ponctuations
- Espaces
- Nombres
- POS
- Normalisation via stem ou lem

## Modèle avancé

- Les étapes précédentes
- ~~Stopwords~~
- Ponctuations
- Espaces
- ~~Nombres~~
- ~~POS~~
- Normalisation via stem ou lem

# Séparation des données

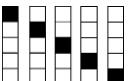
## Modèle simple

- Split train / test

- Predictions
- Interprétabilité

- Hyperparamètres
- Capacité à généraliser
- Choix modèle



- Validation croisée 

## Modèle avancé

- Split train / test

- Predictions
- Interprétabilité

- Hyperparamètres
- Capacité à généraliser
- Choix modèle

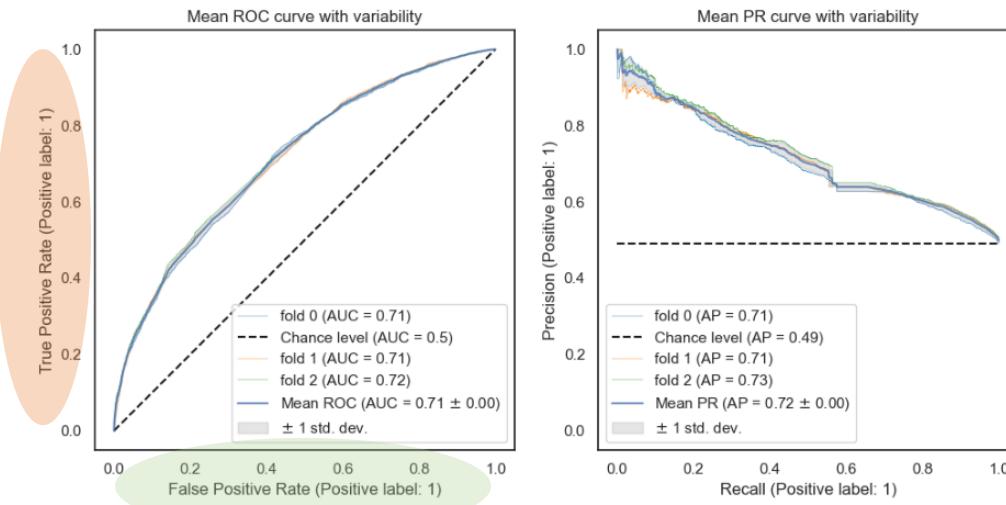


- Split train / validation

# Métrique d'évaluation

- Dataset équilibré
- Minimiser les FN → Maximiser le Recall
- Minimiser les FP → Minimiser le FPR
- Choix : ROC AUC

		Predicted Class		
		Positive	Negative	
Actual Class	Positive	True Positive (TP)	False Negative (FN) Type II Error	Sensitivity $\frac{TP}{(TP + FN)}$
	Negative	False Positive (FP) Type I Error	True Negative (TN)	Specificity $\frac{TN}{(TN + FP)}$
	Precision	$\frac{TP}{(TP + FP)}$	Negative Predictive Value	$\frac{TN}{(TN + FN)}$
				Accuracy
				$\frac{TP + TN}{(TN + FP + FN)}$



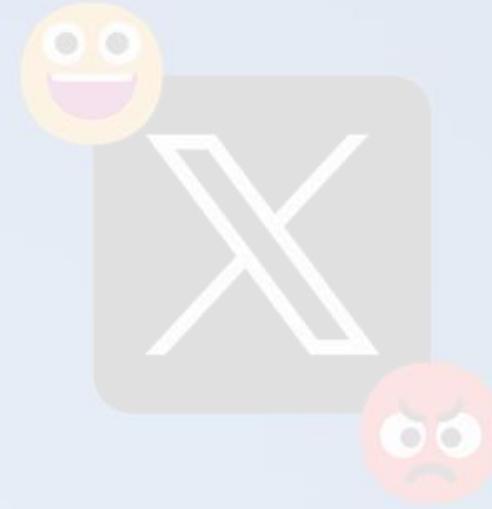
ROC AUC (+accuracy + training time)



mlflow™



# PARTIE 3 – MODÈLE SIMPLE



# Modèle simple - pipeline

- Pipeline d'apprentissage dans une expérimentation **mlflow** :

- Créer :

Model :

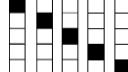
generalCleaner

simpleCleaner

TfidfVectorizer

LogisticRegression

params → **mlflow**

- Validation croisée : 

AUC, Acc, plots → **mlflow**

- Ré-entraîner sur toutes les données :

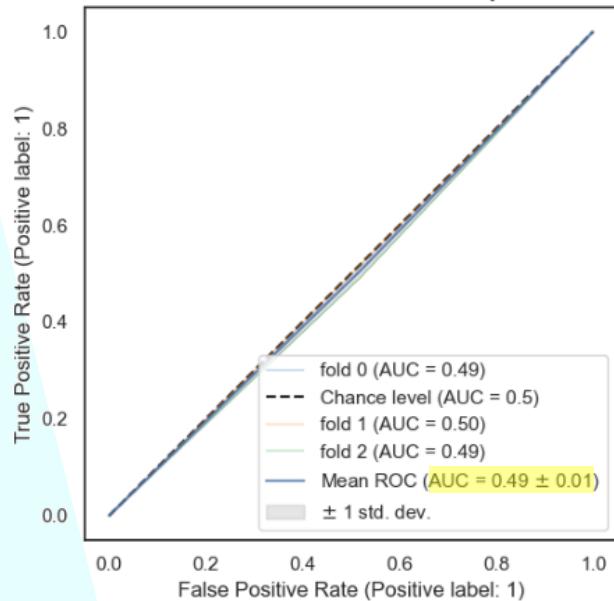
Model, training\_time → **mlflow**



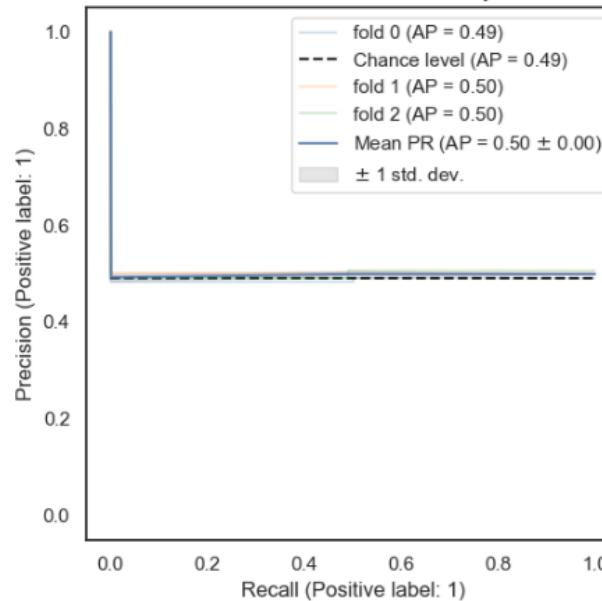
# Modèle simple – premiers essais

tf\_idf\_dummy  
ROC and Precision-Recall curves  
from Cross Validation  
(using the mean of folds curves)

Mean ROC curve with variability

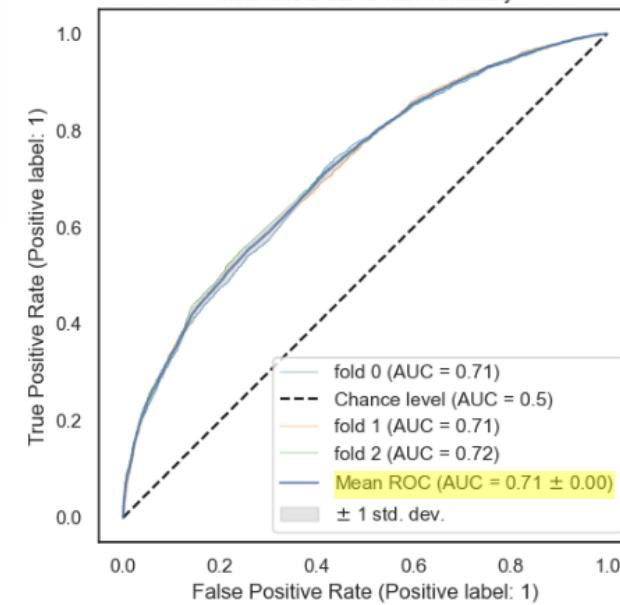


Mean PR curve with variability

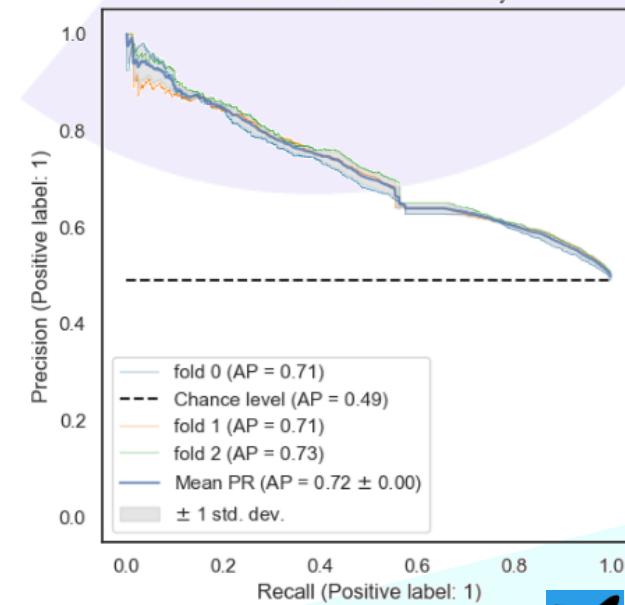


tf\_idf\_lr  
ROC and Precision-Recall curves  
from Cross Validation  
(using the mean of folds curves)

Mean ROC curve with variability



Mean PR curve with variability



# Modèle simple – recherche hyperparamètres

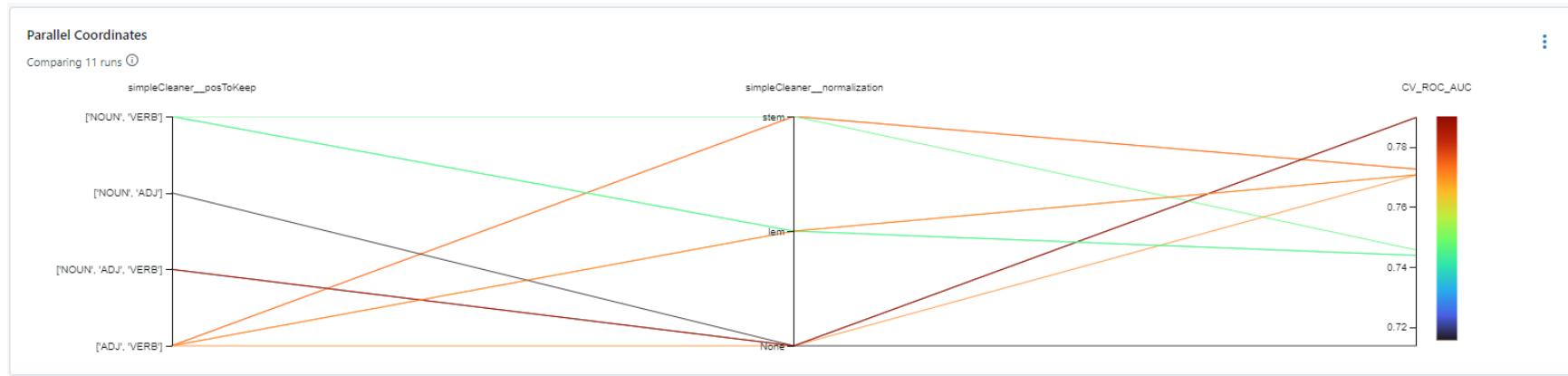
- Utiliser un run `mlflow` PARENT :
  - Créer une fonction objectif  OPTUNA :
    - utilise le pipeline précédent
    - au sein d'un run `mlflow` ENFANT
  - Lancer une étude  OPTUNA

<input type="checkbox"/>		● simple_optuna_logReg_PARENT	 25 days ago	0.71311111...	0.79003970...	0.34929347...	1.18710233...	I2	lbfgs	-	-	best_simple...
<input type="checkbox"/>		● simple_optuna_logReg_29	 25 days ago	0.64605555...	0.68285891...	-	6.38170164...	None	newton-cg	-	-	candidate
<input type="checkbox"/>		● simple_optuna_logReg_28	 25 days ago	0.69472222...	0.76726859...	-	15.3271813...	I2	lbfgs	-	-	candidate
<input type="checkbox"/>		● simple_optuna_logReg_27	 25 days ago	0.71311111...	0.79003970...	-	1.18710233...	I2	lbfgs	-	-	candidate
...												
<input type="checkbox"/>		● simple_optuna_cleaner_PARENT	 25 days ago	0.71305555...	0.78976476...	16.7696905...	-	-	-	None	[NOUN, 'A...	best_simple...
<input type="checkbox"/>		● simple_optuna_cleaner_9	 25 days ago	0.69722222...	0.77254813...	-	-	-	-	stem	[ADJ, 'VERB']	candidate
<input type="checkbox"/>		● simple_optuna_cleaner_8	 25 days ago	0.69722222...	0.77254813...	-	-	-	-	stem	[ADJ, 'VERB']	candidate
<input type="checkbox"/>		● simple_optuna_cleaner_7	 25 days ago	0.68161111...	0.74562650...	-	-	-	-	stem	[NOUN, 'V...	candidate

# Modèle simple – recherche hyperparamètres

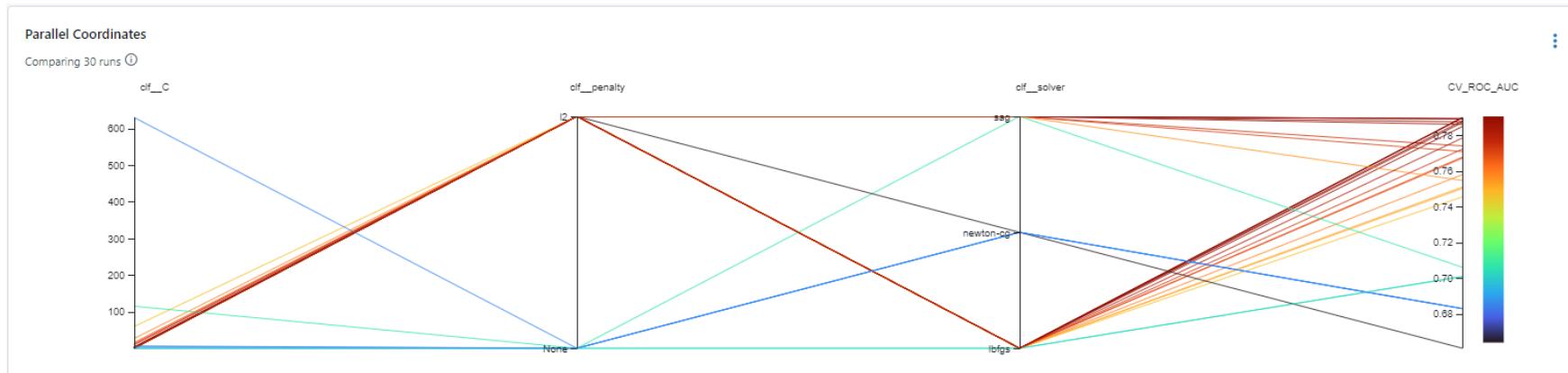
- Grille preprocessing / cleaner :

mlflow™



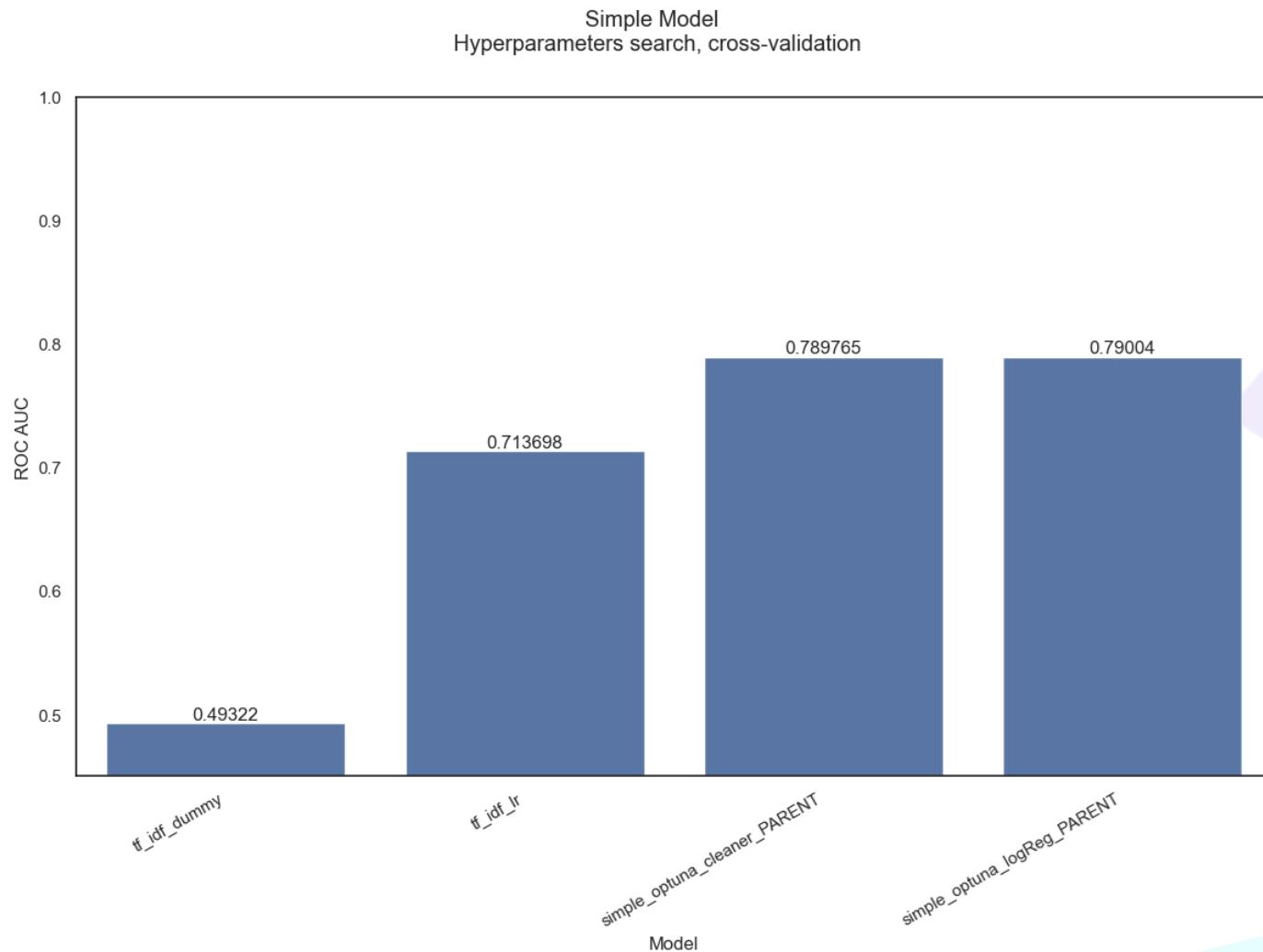
- Grille régression logistique :

mlflow™



# Modèle simple – recherche hyperparamètres

- Bilan :



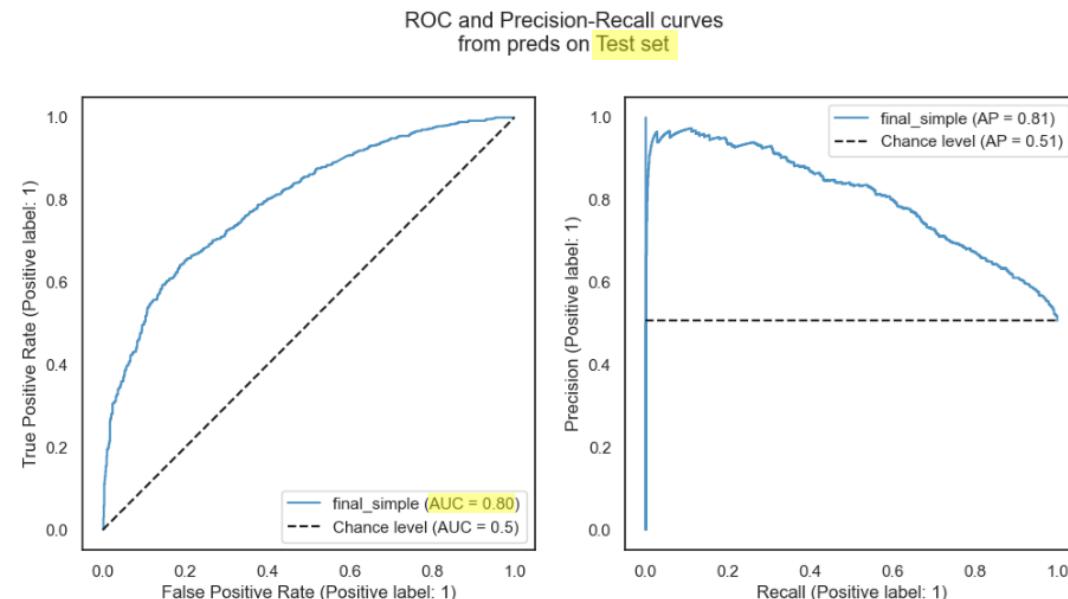
# Modèle simple – modèle final

- Modèle final avec hyperparamètres optimisés :

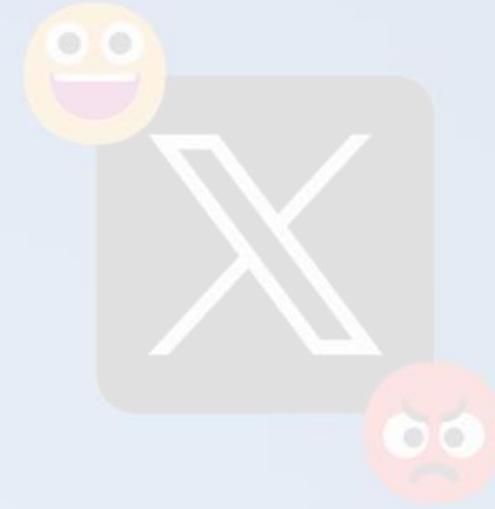
```
# create a pipeline with our simpleCleaner, a Tfidf and a Logistic Regression
completLrPipe = Pipeline([
    ...
    ("generalCleaner", mf.generalCleaner(url = "url", email = "email", hashtag = "hashtag", mention = "mention")),
    ("simpleCleaner", mf.simpleModelCleaner()),
    ("vect", TfidfVectorizer()),
    ("clf", LogisticRegression())
])

```

{'simpleCleaner\_normalization': 'None',  
 'simpleCleaner\_posToKeep': "['NOUN', 'ADJ', 'VERB']",  
 'clf\_C': 1.1871023326363952,  
 'clf\_solver': 'lbfgs',  
 'clf\_penalty': 'l2'}



# PARTIE 4 – MODÈLE AVANCÉ



# Modèle avancé

## Text vectorization Keras :

- standardize :
  - generalClean
  - advancedModelClean
- Taille vocabulaire
- Taille des séquences

## Sequential model Keras :

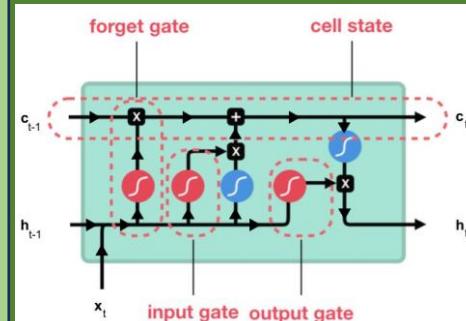
### Embedding :

- Avec ou sans poids pré-entraînés

GloVe

fastText

### Bidirectionnal LSTM :



### Dense :

- 16 unités
- Activation : *relu*

### Dense :

- 1 unité
- Activation : *sigmoïde*



# Modèle avancé - pipeline

- Pipeline d'apprentissage dans une expérimentation **mlflow** :

- Créer :

**Text\_vectorization**  Keras :

- standardize :
  - generalClean
  - advancedModelClean
- Taille vocabulaire
- Taille des séquences

params → **mlflow**

- Créer la matrice d'embedding

- Créer :

**Sequential model**  Keras :

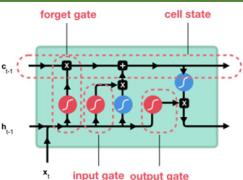
**Embedding** :

- Avec ou sans poids pré-entraînés

**GloVe**

**fastText**

**Bidirectionnal LSTM** :



**Dense** :

- 16 unités
- Activation : *reLu*

**Dense** :

- 1 unité
- Activation : *sigmoïde*

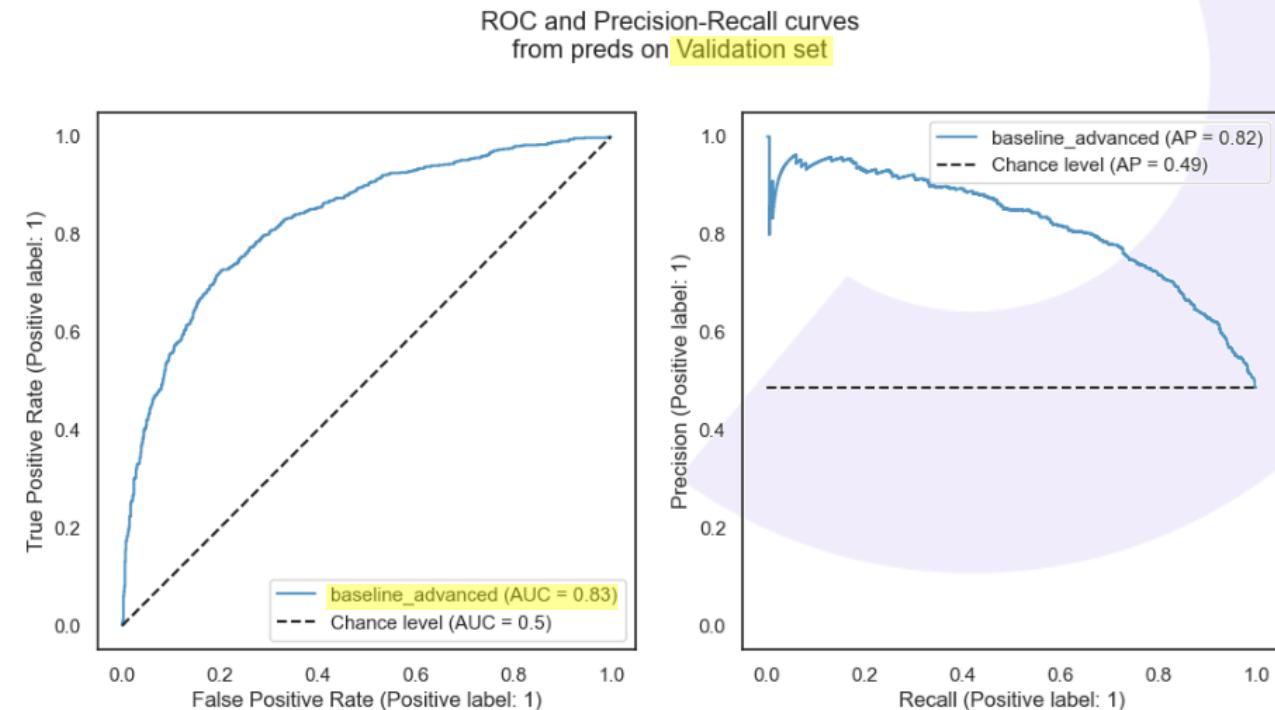
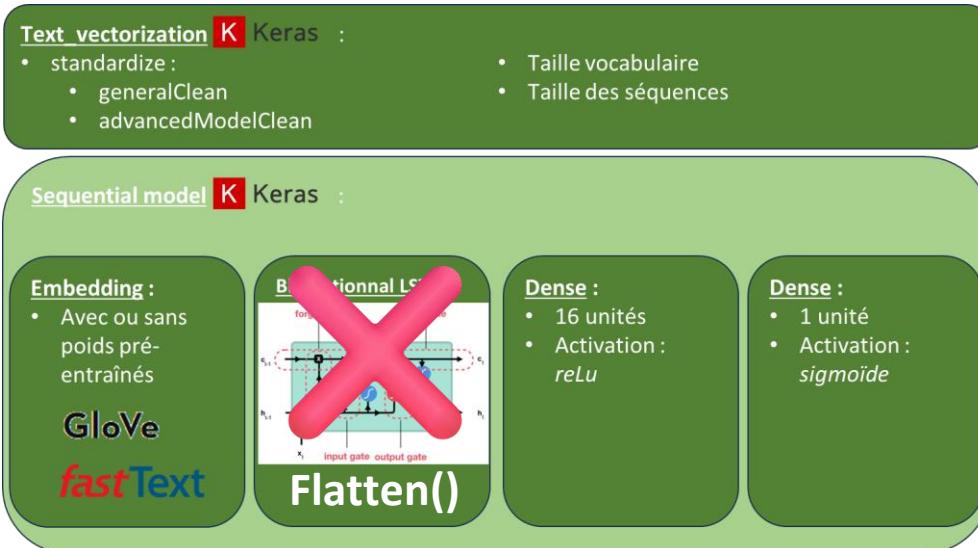
params → **mlflow**

- Entraîner le modèle

AUC, Acc, plots, Model, training\_time → **mlflow**

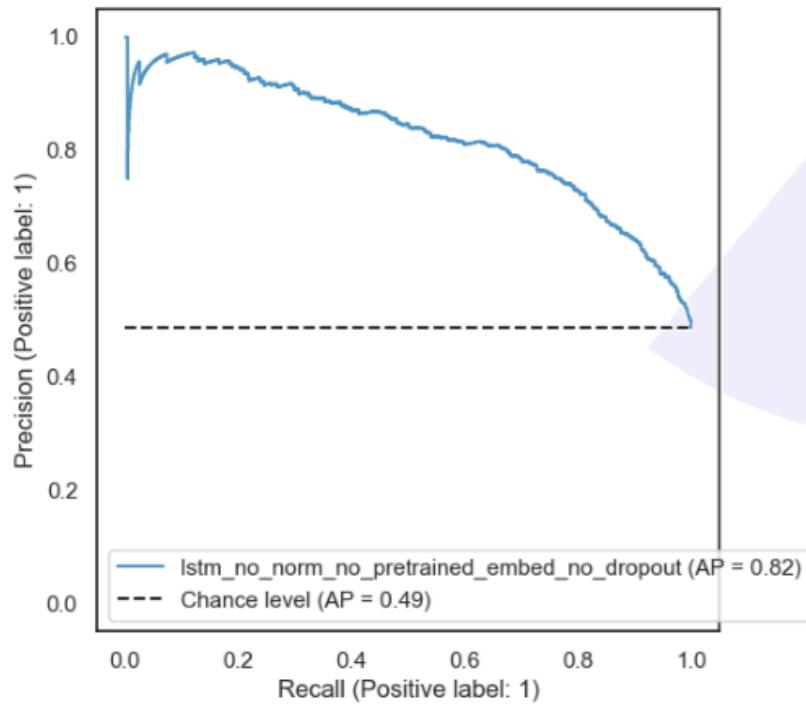
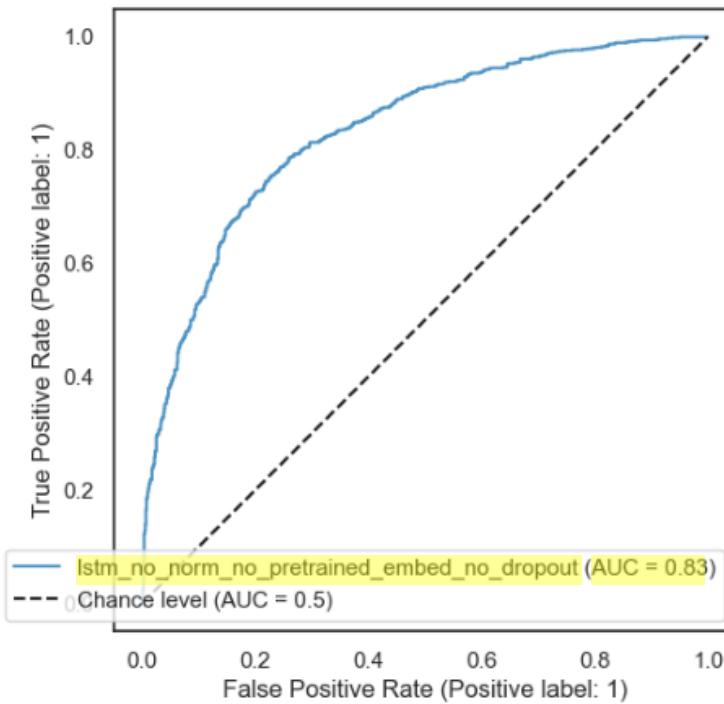


# Modèle avancé - baseline



# Modèle avancé – premier essai

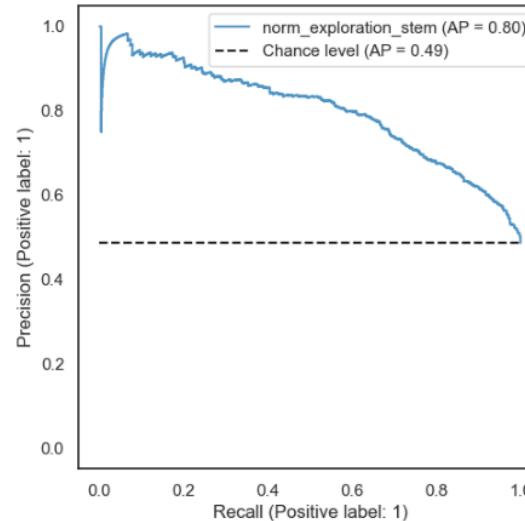
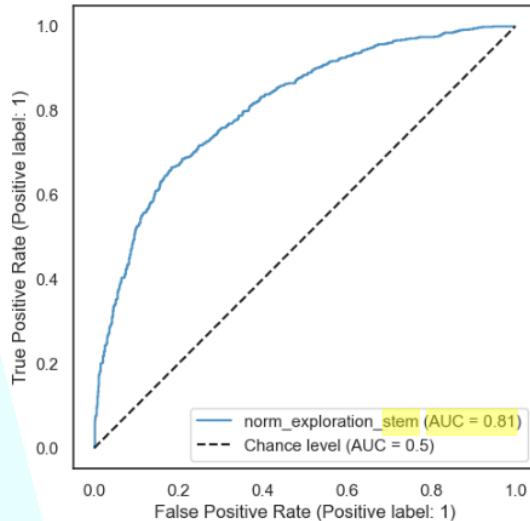
ROC and Precision-Recall curves  
from preds on Validation set



# Modèle avancé – recherche hyperparamètres

- normalization = "stem" ou normalization = "lem" ?

ROC and Precision-Recall curves  
from preds on Validation set



ROC and Precision-Recall curves  
from preds on Validation set

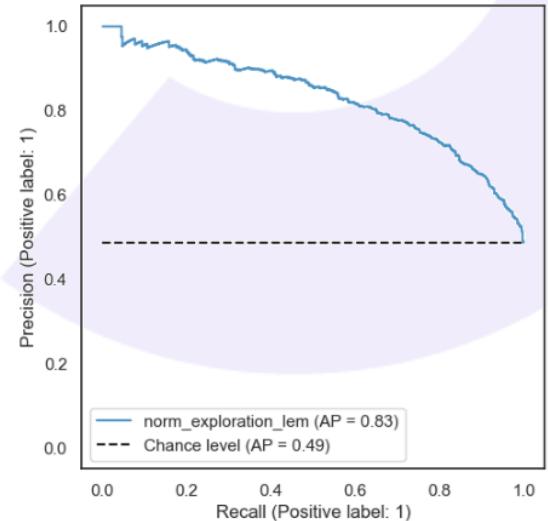
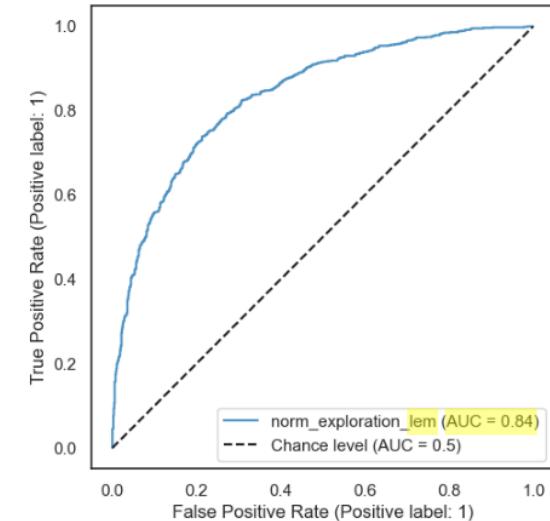
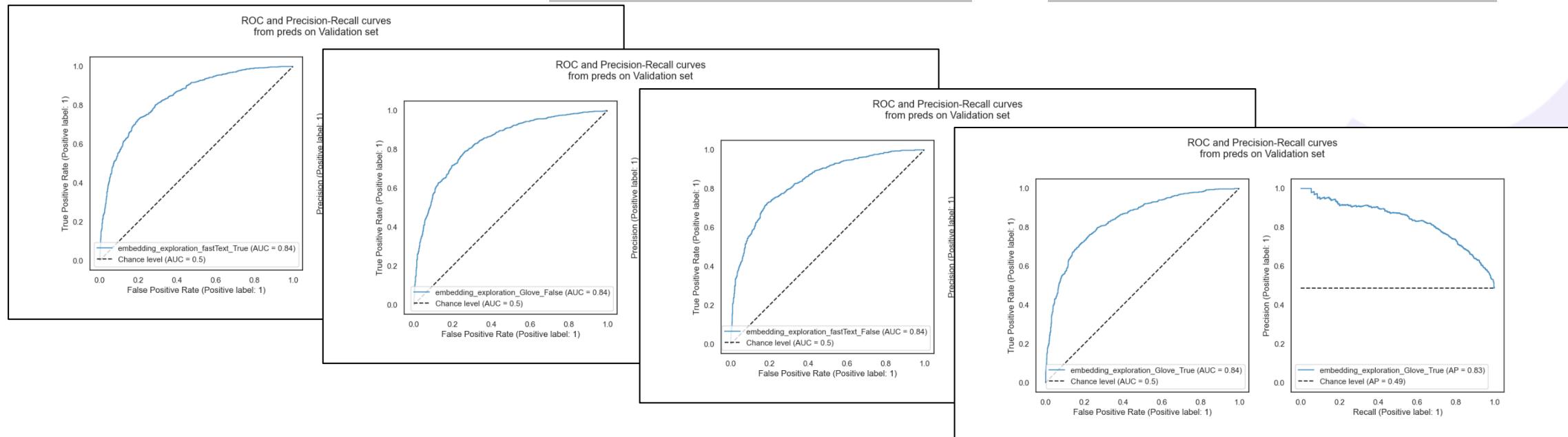


Table	Chart	Evaluation	Experimental	Metrics					Parameters	Tags
				Created	Duration	training_time	valid_Accuracy	valid_ROC_AUC	normalization	status
			● norm_exploration_lem	25 days ago	2.3min	138.859679...	0.75833332...	0.83659863...	lem	candidate
			● norm_exploration_stem	25 days ago	1.9min	115.127860...	0.73388886...	0.81029224...	stem	candidate
			● lstm_no_norm_no_pretrained_embed_no_dr...	25 days ago	1.6min	96.6895365...	0.76055556...	0.83386272...	None	candidate



# Modèle avancé – recherche hyperparamètres

- **GloVe ou fastText ?** trainable = True ou trainable = False ?

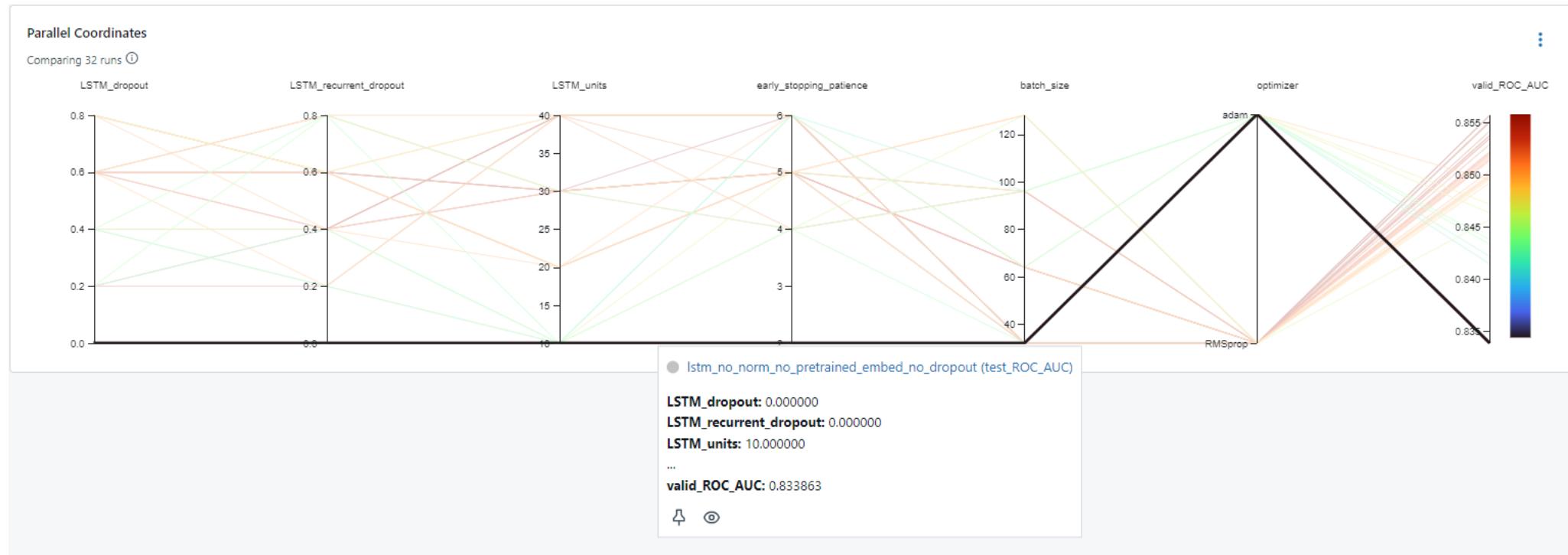


	Run Name	Created	Duration	Metrics	Parameters	Tags
<input type="checkbox"/>	embedding_exploration_fastText_True	25 days ago	3.1min	training_time valid_Accuracy valid_ROC_AUC	embedding_file_path pretrained_embeddings/fasttext-wiki-news-300d-1M.vec	trainable_embec status
<input type="checkbox"/>	embedding_exploration_Glove_True	25 days ago	2.4min	183.450798... 0.75166666... 0.83910292...	pretrained_embeddings/glove.6B.300d.txt	True candidate
<input type="checkbox"/>	embedding_exploration_fastText_False	25 days ago	3.1min	182.859122... 0.76666665... 0.83901143...	pretrained_embeddings/fasttext-wiki-news-300d-1M.vec	False candidate
<input type="checkbox"/>	embedding_exploration_Glove_False	25 days ago	2.2min	130.811452... 0.75944441... 0.83685868...	pretrained_embeddings/glove.6B.300d.txt	False candidate



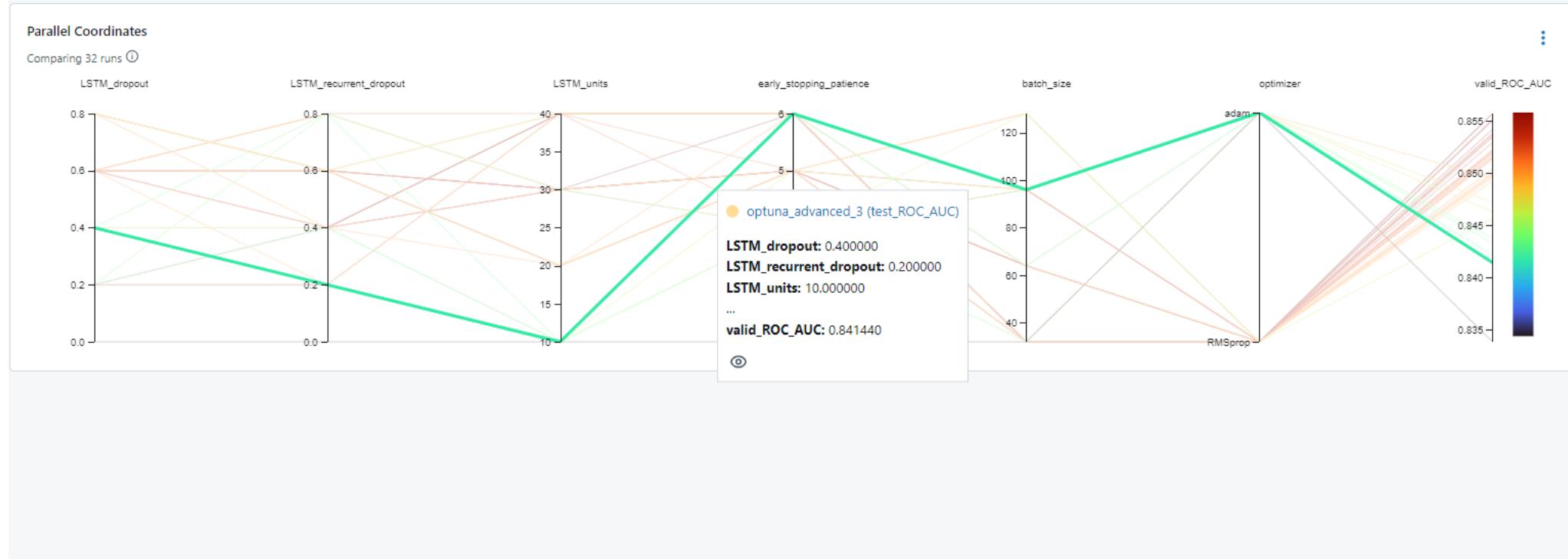
# Modèle avancé – recherche hyperparamètres

- Comme précédemment : **mlflow™** et  **OPTUNA**



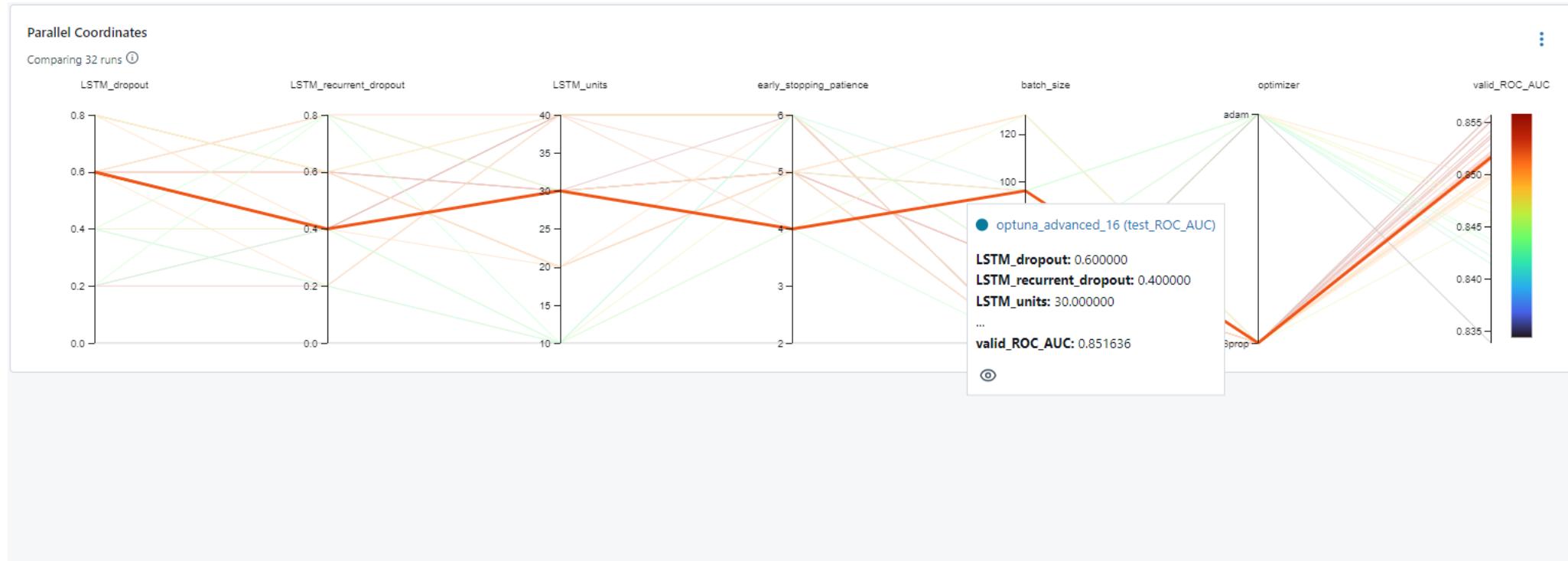
# Modèle avancé – recherche hyperparamètres

- Comme précédemment : **mlflow™** et  **OPTUNA**



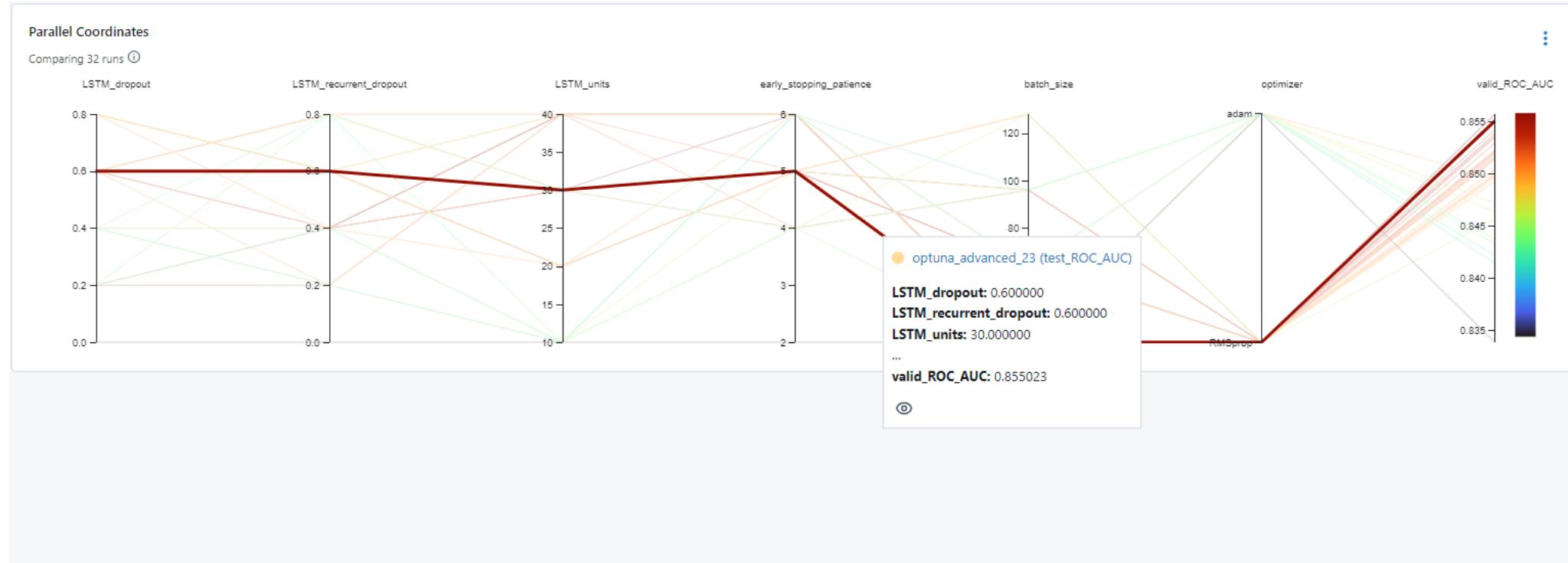
# Modèle avancé – recherche hyperparamètres

- Comme précédemment : **mlflow** et **OPTUNA**



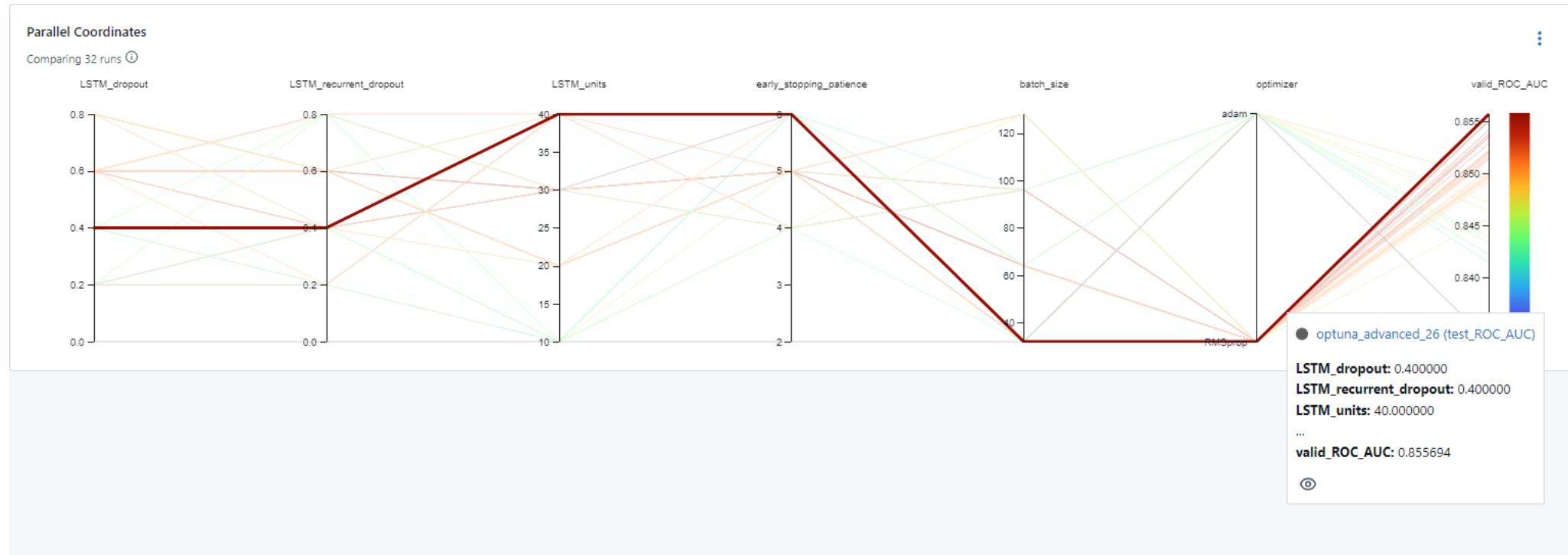
# Modèle avancé – recherche hyperparamètres

- Comme précédemment : mlflow™ et  OPTUNA



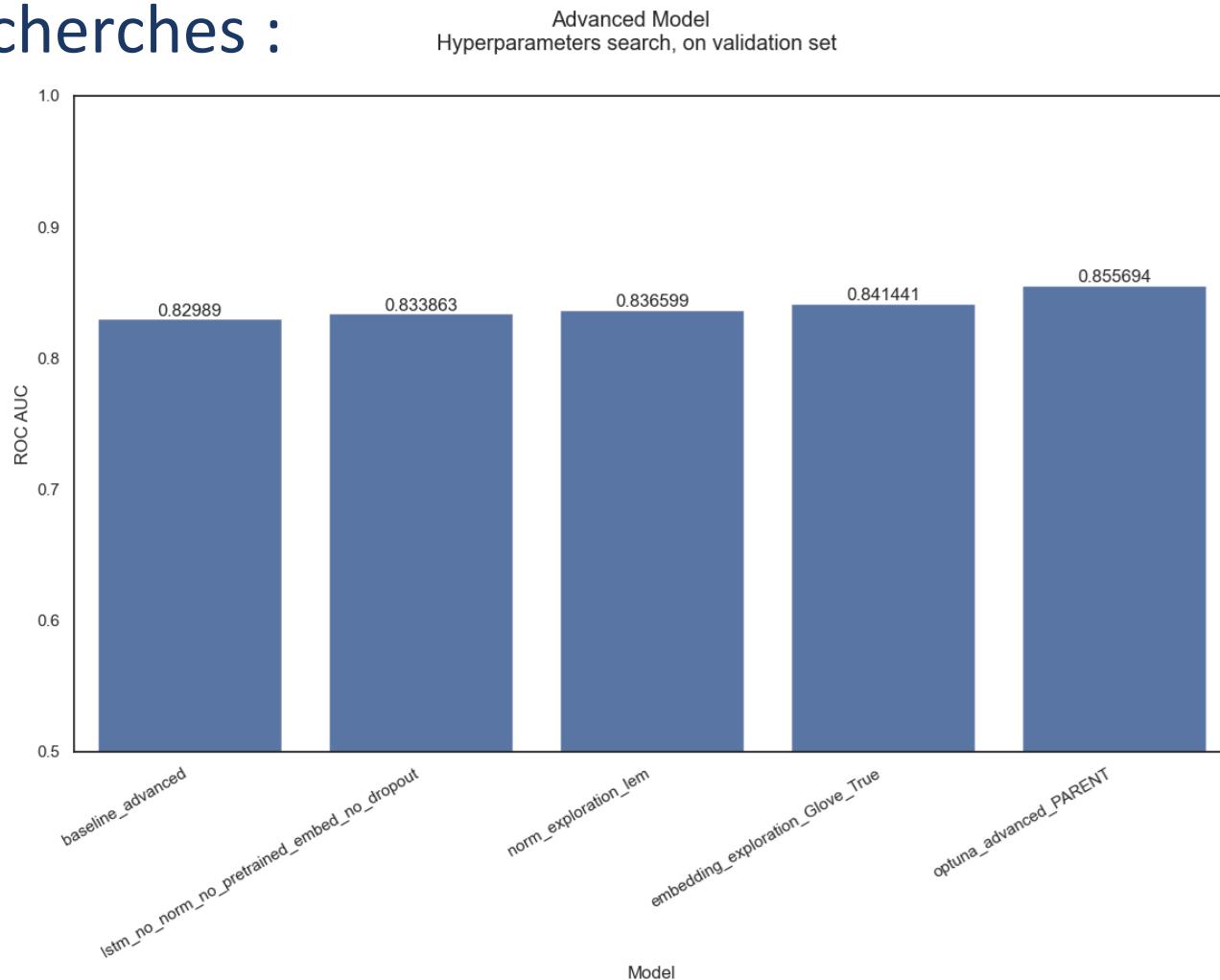
# Modèle avancé – recherche hyperparamètres

- Comme précédemment : **mlflow™** et  **OPTUNA**



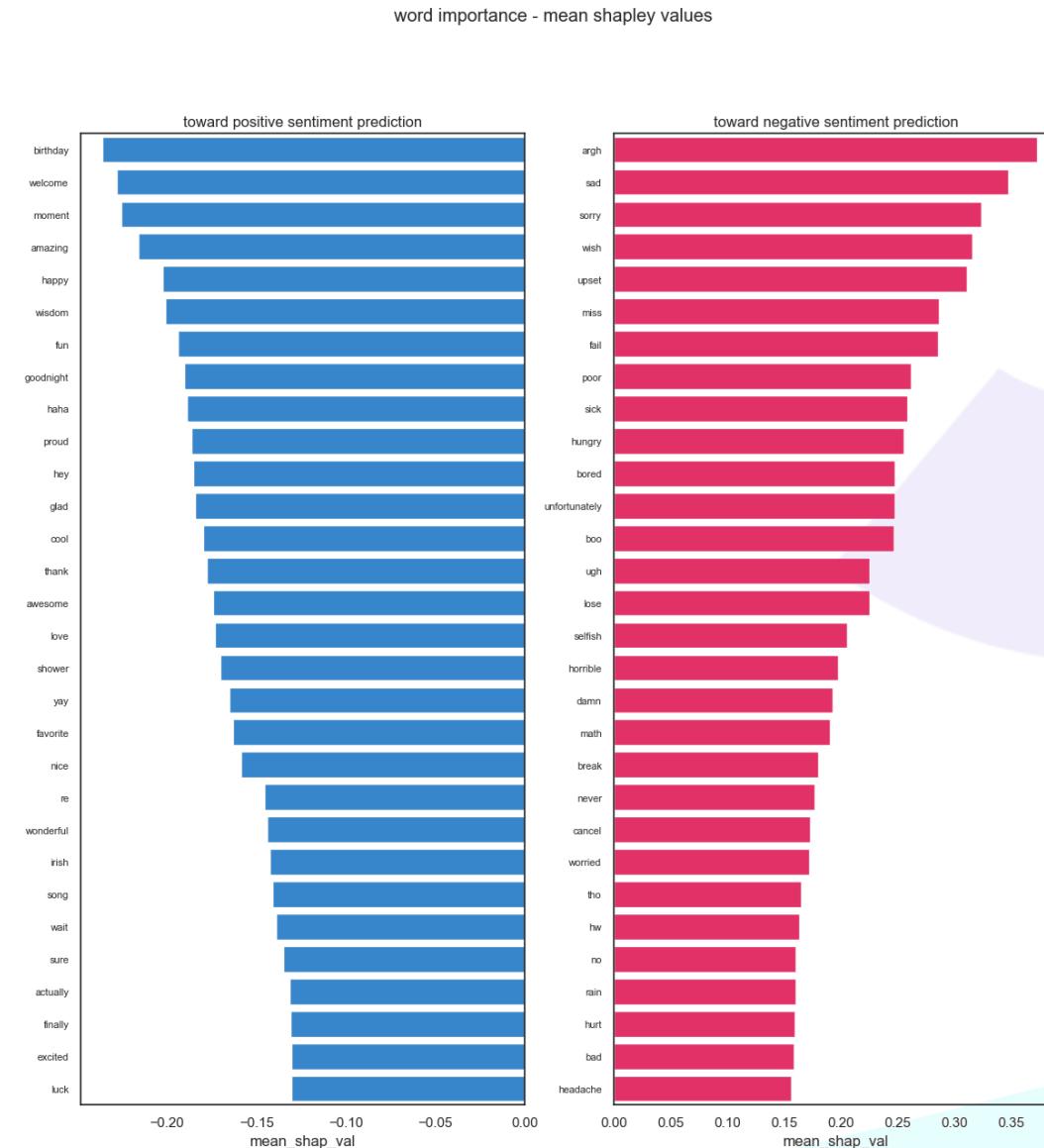
# Modèle avancé - recherche hyperparamètres

- Bilan des recherches :



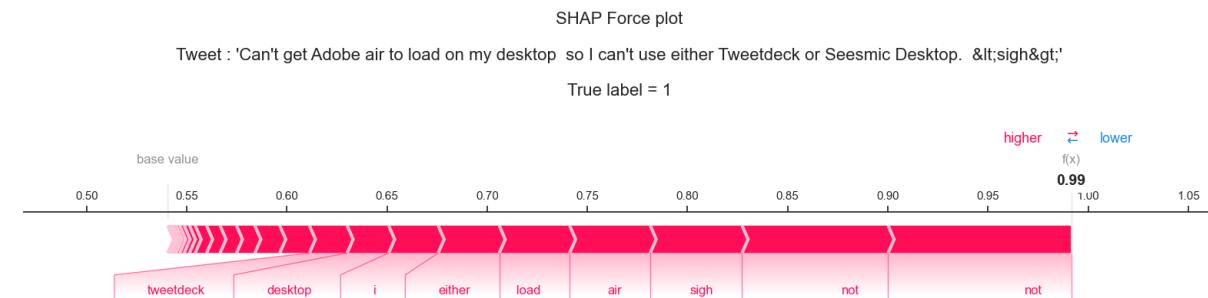
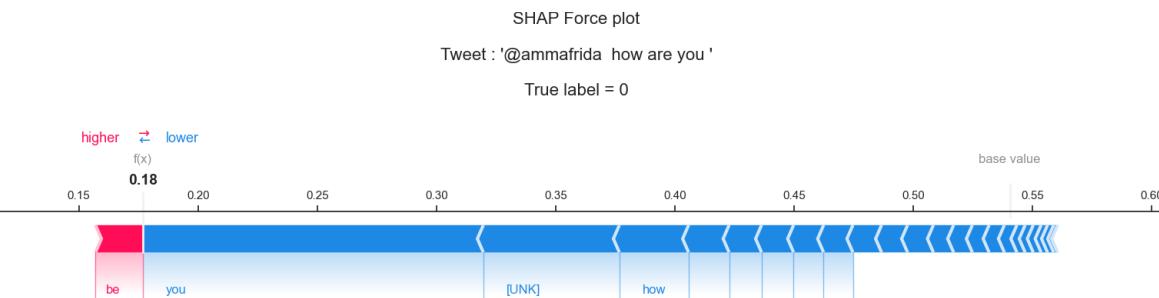
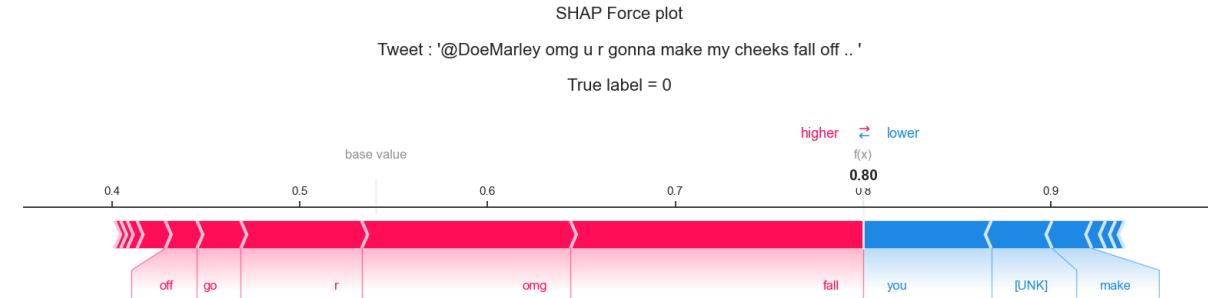
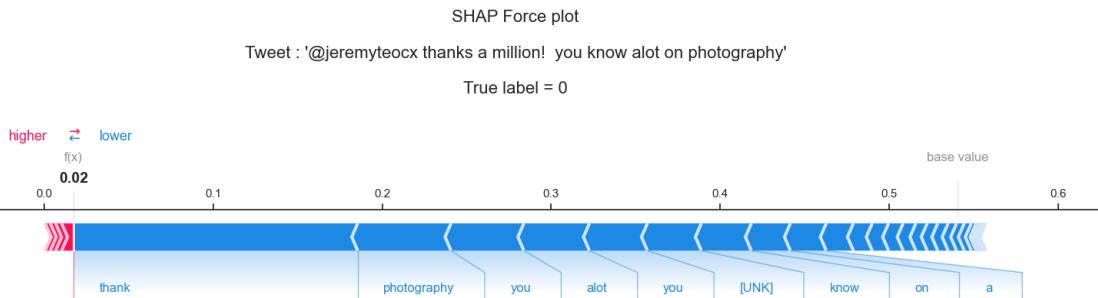
# Modèle avancé – expliquer les résultats

- Utilisation de  Shap
  - Au niveau **global** :



# Modèle avancé – expliquer les résultats

- Utilisation de  Shap
- Au niveau local :



# PARTIE 5 – TESTS AUTRES SOLUTIONS et BILAN



# BERT (Bidirectional Encoder Representations from Transformers)

- Modèle ayant considérablement fait avancer le NLP en termes de performance
- Architecture basée sur la partie **encoder** du transformer, et son **mécanisme d'attention**
- BERT n'a ainsi pas appris une tâche spécifique (comme de la traduction, etc.), mais une « **compréhension** » **très fine** du langage
- On peut alors appliquer, via transfer learning, **n'importe quel type de tâche**, comme de la classification de texte
- Ici : utilisation de **DistilBERT** (moins performant mais plus petit, plus rapide)

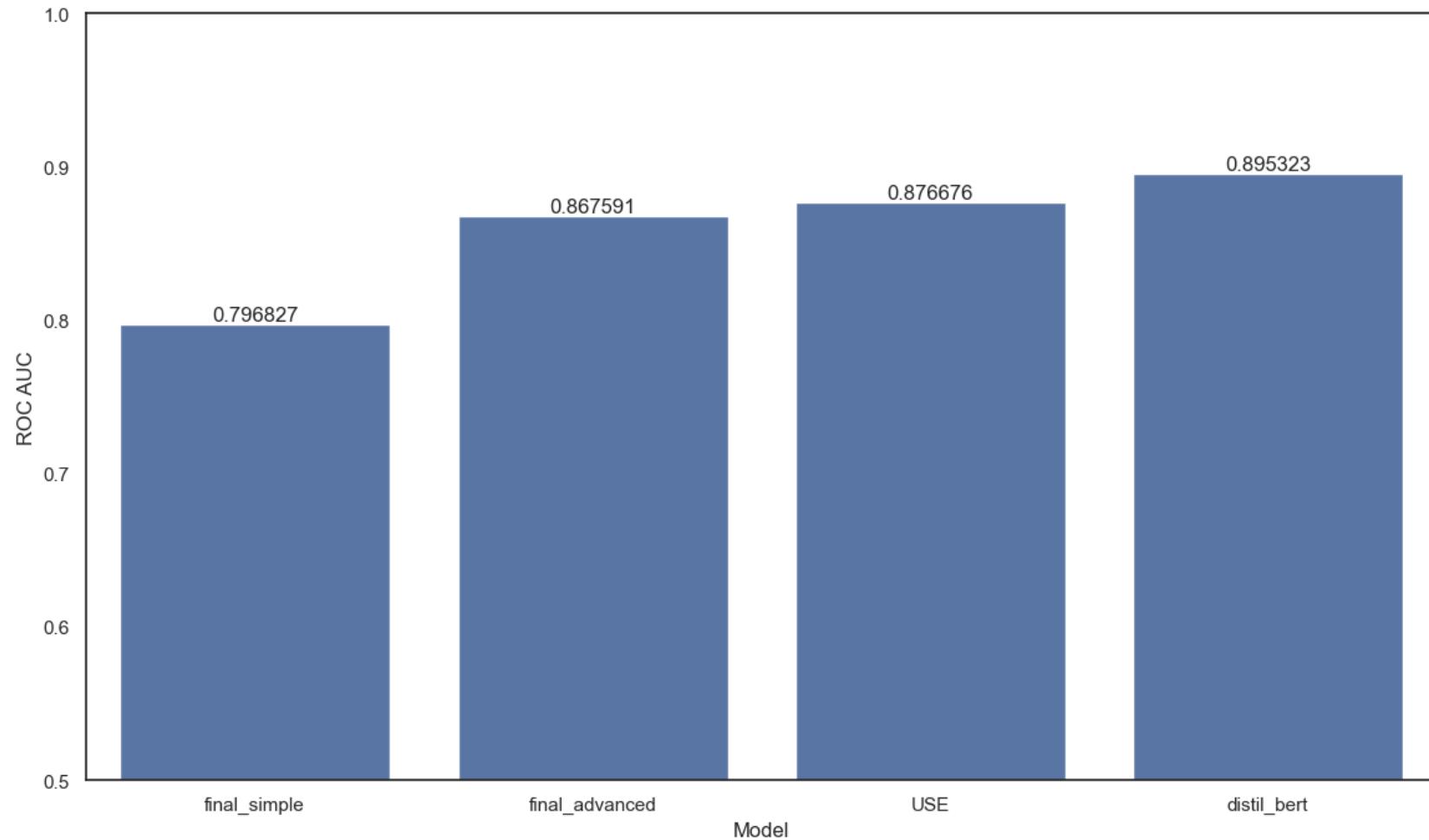
# USE (Universal Sentence Encoder)

- Ici, ***sentence embedding***
- Également développé par Google
- **Moins consommatrice** en ressource mémoire

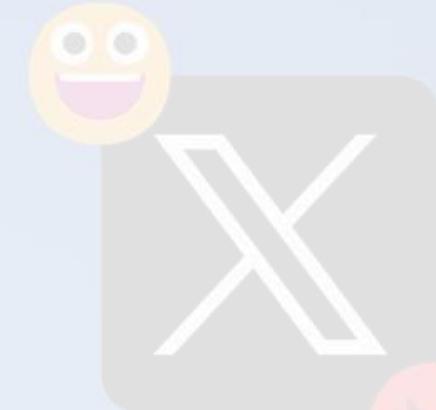


# Démarche de modélisation - bilan

Final models, on test set



# PARTIE 6 – DÉPLOIEMENT API et UI



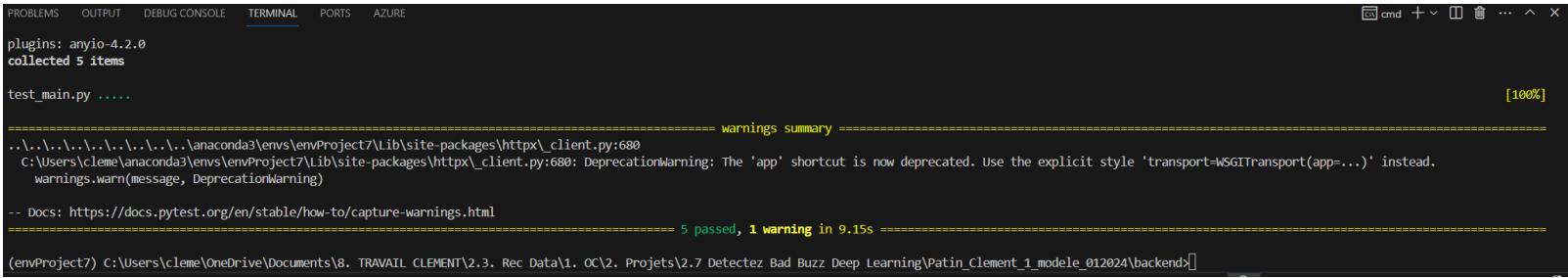
# API

- entraîner modèle sur plus de données
- enregistrer 
- main.py 
- test\_main.py pour les tests unitaires 
- Dockerfile 
- Construire image docker 
- *resource group* 
- *appservice plan* 
- *webapp* 
- Déploiement continu  +  GitHub Actions



# API

## Tests unitaires :



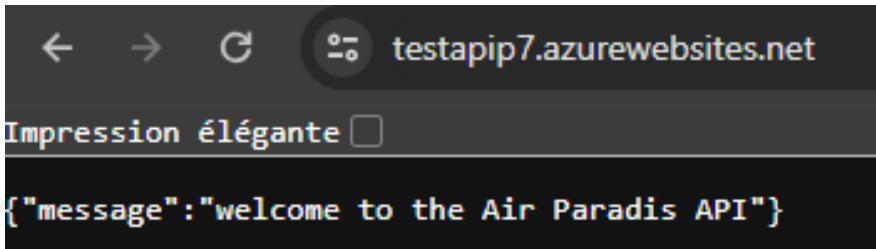
```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS AZURE
plugins: anyio-4.2.0
collected 5 items

test_main.py .... [100%]

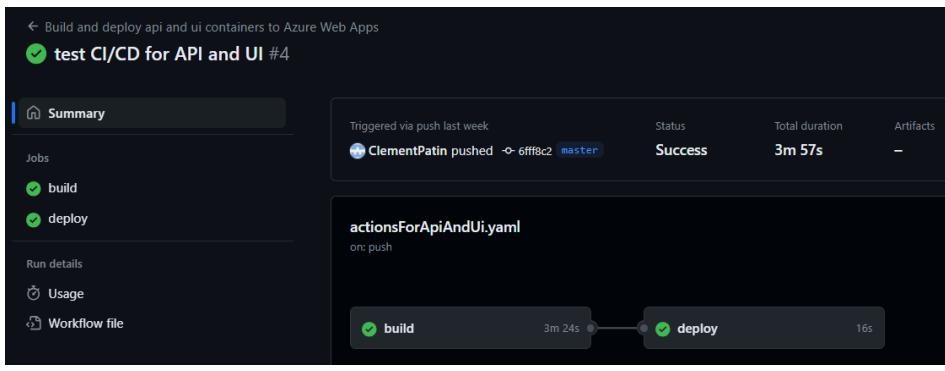
...\\..\\..\\..\\..\\..\\..\\..\\anaconda3\\envs\\envProject7\\Lib\\site-packages\\httpx\\client.py:680
C:\\Users\\cleme\\anaconda3\\envs\\envProject7\\Lib\\site-packages\\httpx\\client.py:680: DeprecationWarning: The 'app' shortcut is now deprecated. Use the explicit style 'transport=WSGITransport(app=...) instead.
  warnings.warn(message, DeprecationWarning)
-- Docs: https://docs.pytest.org/en/stable/how-to/capture-warnings.html
      5 passed, 1 warning in 9.15s

(envProject7) C:\\Users\\cleme\\OneDrive\\Documents\\8. TRAVAIL CLEMENT\\2.3. Rec Data\\1. OC\\2. Projets\\2.7 D\\etectez Bad Buzz Deep Learning\\Patin_Clement_1_modele_012024\\backend[]
```

## API déployée :



## CI/CD:



## Fichier YAML GitHub Actions :

```
! actionsForApiAndUi.yaml •
.github > workflows > ! actionsForApiAndUi.yaml
  10 name: Build and deploy api and ui containers to Azure Web Apps
  11
  12 env:
  13   AZURE_WEBAPP_NAME_API: testApiP7
  14   AZURE_WEBAPP_NAME_UI: testUiP7
  15
  16 on:
  17   push:
  18     branches:
  19       - master
  20
  21 permissions:
  22   contents: 'read'
  23   packages: 'write'
  24
  25 jobs:
  26   build:
  27     runs-on: ubuntu-latest
  28
  29 steps:
  30   - uses: actions/checkout@v4
  31
  32   - name: Set up Docker Buildx
  33     uses: docker/setup-buildx-action@v3
  34
  35   - name: Log in to GitHub container registry
  36     uses: docker/login-action@v3
  37     with:
  38       registry: ghcr.io
  39       username: ${{ github.actor }}
  40       password: ${{ secrets.GITHUB_TOKEN }}
  41
  42   - name: Lowercase the repo name
  43     run: echo "REPO=${GITHUB_REPOSITORY,,}" >>${GITHUB_ENV}
  44
  45   - name: Build and push container image BACKEND to registry
  46     uses: docker/build-push-action@v5
  47     with:
  48       push: true
  49       tags: ghcr.io/${{ env.REPO }}:${{ github.sha }}-api
  50       context: ./Patin_Clement_1_modele_012024/backend
  51
  52   - name: Build and push container image FRONTEND to registry
  53     uses: docker/build-push-action@v5
  54     with:
  55       push: true
  56       tags: ghcr.io/${{ env.REPO }}:${{ github.sha }}-ui
  57       context: ./Patin_Clement_1_modele_012024/frontend
  58
  59 deploy:
  60   runs-on: ubuntu-latest
  61
  62 needs: build
  63
  64 steps:
  65   - name: Lowercase the repo name
  66     run: echo "REPO=${GITHUB_REPOSITORY,,}" >>${GITHUB_ENV}
```

# API – test serving mlflow™

- Via la ligne de commande :
  - se placer dans le dossier contenant /mlruns
  - `mlflow models serve -m "models:/final_advanced_export/1" -p 1234 --no-conda`
- puis tester l'API grâce à la librairie `requests` :

The screenshot shows a Python code editor interface with a dark theme. A code editor window displays the following Python script:

```
import requests

inference_request = {
    "inputs": ["I hate humans", "I love humans", "I say I love humans but it's not true"]
}

endpoint = "http://127.0.0.1:1234/inversions"

response = requests.post(endpoint, json=inference_request)

print(response.text)
```

The code uses the `requests` library to send a POST request to the endpoint `http://127.0.0.1:1234/inversions` with a JSON payload containing three input strings. The response is then printed to the console.

Below the code editor, the terminal output shows several warning messages from TensorFlow and TensorRT, followed by the message "INFO:waitress:Serving on http://127.0.0.1:1234".

The interface includes standard VS Code navigation and search tools at the top, and a bottom bar with tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL, PORTS, JUPYTER, and AZURE, along with a Python icon.



# UI

- `streamlit_app.py`  Streamlit
- Dockerfile  docker
- Construire image docker  docker
- *resource group* 
- *appservice plan* 
- *webapp* 
- Déploiement continu  +  GitHub Actions

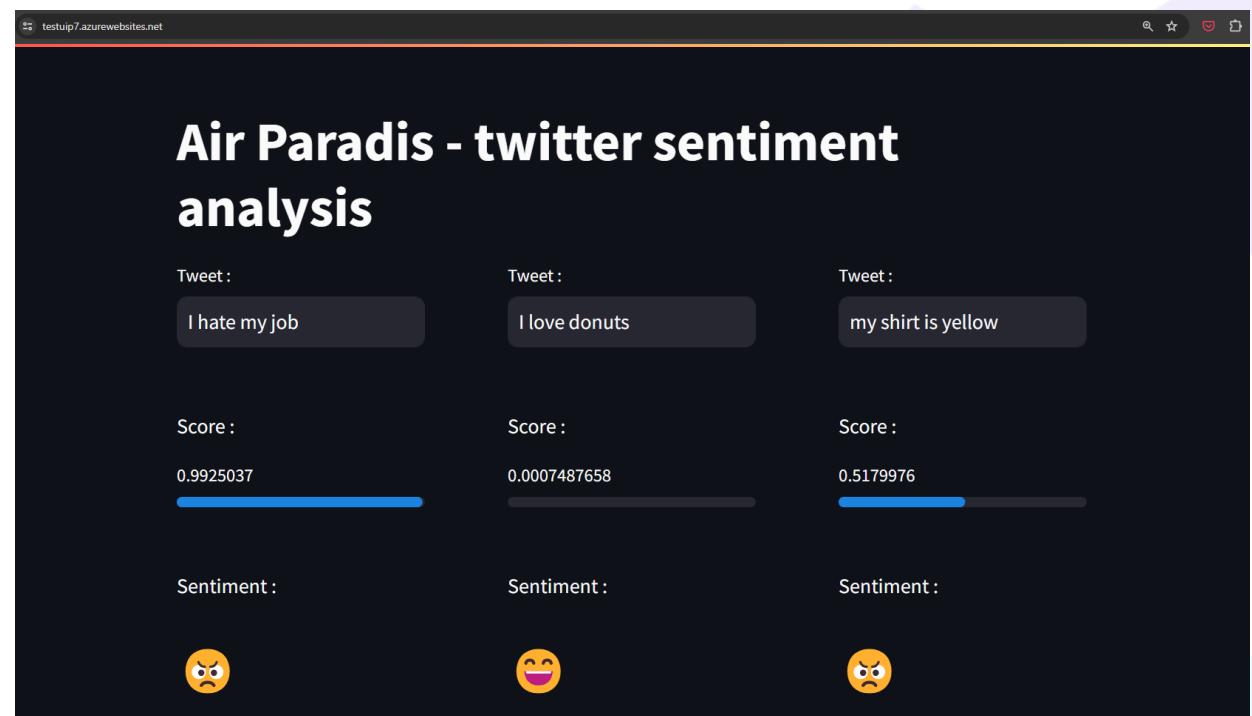


# UI

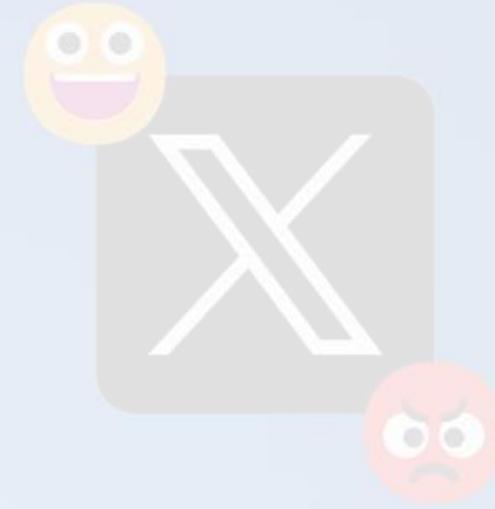
## app streamlit :

```
streamlit_app.py x
Patin_Clement_1_modele_012024 > frontend > streamlit_app.py > ...
1 import streamlit as st
2 import requests
3
4 st.title("Air Paradis - twitter sentiment analysis")
5
6 col1, col2, col3 = st.columns(3, gap="large")
7 baseInputs = ["I hate my job", "I love donuts", "my shirt is yellow"]
8
9 for i,(col,baseInput) in enumerate(zip([col1, col2, col3], baseInputs)) :
10    with col :
11
12        tweet = st.text_input(label="Tweet :", value=baseInput, key=i)
13
14        st.write("")
15        st.write("")
16
17        API_URL = "https://testapi7.azurewebsites.net"
18
19        response = requests.post(API_URL+ "/predict", json={"text":tweet}).json()
20        # response = requests.post(API_URL+ "/predict?text="+tweet)
21
22        if "score" not in response.keys() :
23            st.write(":exclamation: Error :exclamation: :")
24            st.write("Detail : "+response["detail"])
25
26        else :
27            st.write("Score :")
28            st.progress(float(response["score"]), text = response["score"])
29            st.write("")
30            st.write("")
31
32            st.write("Sentiment :")
33            if response["sentiment"] == "positive" :
34                st.header(":smile:")
35            else :
36                st.header(" :angry:")
37
38
```

## app streamlit :



# CONCLUSION



merci

