

# Prédiction de tags

MARQUER Matthieu  
MLE- Projet 5 – décembre 2023



# Sommaire:

**01**

Introduction

**02**

Exploration des  
données

**03**

Stackapi

**04**

Approche non  
supervisée

**05**

Approche  
supervisée

**06**

API

**07**

Conclusion

Projet portant sur la prédiction de Tags pour Stack Overflow:

Utilisation de StackExchange pour récupérer des données de Stack Overflow

Évaluation de StackAPI pour une intégration future

Création de modèles de prédiction de tags et utilisation de FastAPI

Suivi de performance avec MLFlow

Intégration de GitHub Actions pour automatiser le processus

Déploiement sur Heroku pour la mise en production de la solution

## 1. Requête SQL :

<https://data.stackexchange.com/stackoverflow/query/new>

```
SELECT TOP 50000
  Id, Title, Body, Tags, Score, ViewCount, FavoriteCount, AnswerCount, CreationDate
FROM
  Posts
WHERE
  PostTypeId = 1
  AND ViewCount > 10
  AND Score > 5
  AND AnswerCount > 0
  AND LEN(Tags) - LEN(REPLACE(Tags, '<', '')) >= 5
  -- AND Id BETWEEN 1000 AND 100000
ORDER BY
  Score DESC;
```

2. Vue global du dataframe, vérification du nombre de lignes (50 000), des types, des valeurs manquantes et unique
3. Récupération des tags, limite aux 50 tags les plus utilisés
4. Création du variable reprenant le titre et le corps
5. Suppression des liens html, des stop word et limitation à 25 caractères. Garde seulement les lettres
6. Lemmatisation/stematisation en fonction du besoin
7. Sauvegarde du CSV nettoyé

Top tags:

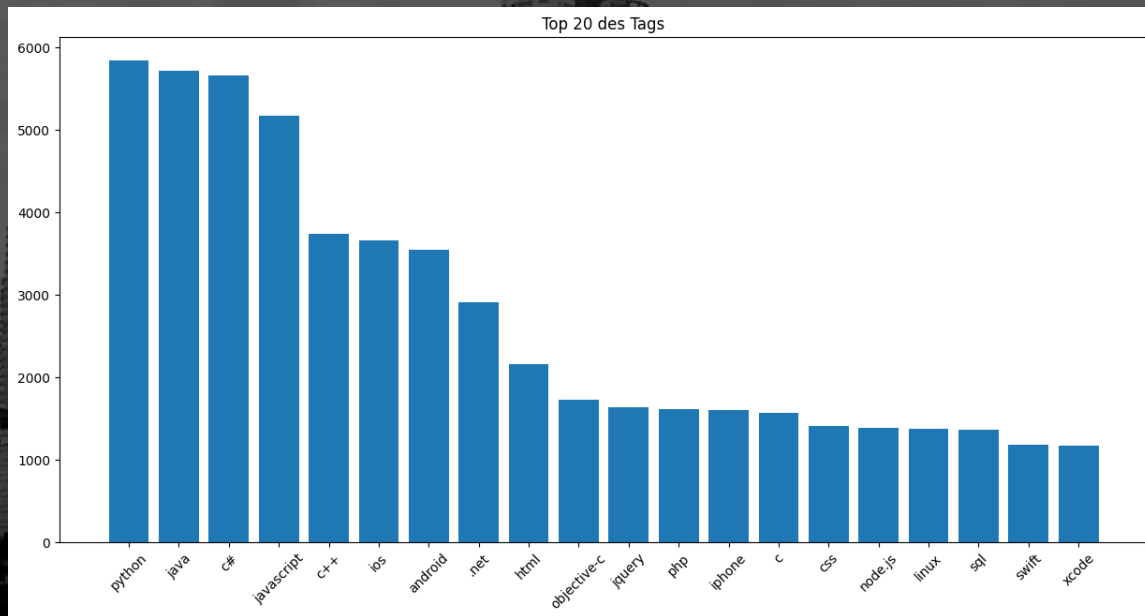
Python

Java

C#

Javascript

### Top 20 des tags les plus utilisés



	date	title	tags	score
0	2017-01-10 16:27:43	What is the difference between venv, pyenv, p...	[python, virtualenv, virtualenvwrapper, pyenv,...	2083
1	2017-03-26 11:11:36	Purpose of &quot;%matplotlib inline&quot;	[python, matplotlib, jupyter-notebook, ipython]	935
2	2017-07-25 17:41:20	Fixed digits after decimal with f-strings	[python, python-3.x, f-string]	930
3	2018-12-06 06:41:15	Pandas Merging 101	[python, pandas, join, merge, concatenation]	924
4	2016-04-28 17:46:30	Truth value of a Series is ambiguous. Use a.em...	[python, pandas, dataframe, boolean, filtering]	850
5	2018-03-06 09:49:50	Removing Conda environment	[python, jupyter, conda, environment]	824
6	2017-11-02 06:10:46	Your CPU supports instructions that this Tenso...	[python, tensorflow, cpu, avx]	777
7	2016-10-11 14:59:23	Are dictionaries ordered in Python 3.6+?	[python, python-3.x, dictionary, python-intern...	745
8	2016-04-06 16:25:46	Specify which pytest tests to run from a file	[python, pytest]	716
9	2017-11-02 09:03:54	How to update/upgrade a package using pip?	[python, pip]	711
10	2016-08-11 12:28:24	TensorFlow not found using pip	[python, tensorflow, pip]	710
11	2016-08-02 18:01:36	How do I add default parameters to functions w...	[python, type-hinting, python-typing]	679

Une expérience avec StackAPI a été réalisée pour obtenir un DataFrame de 50 lignes.

Cependant, le format de la date (par exemple : fromdate=1457136000) et le système de limitation des questions (pagesize=10) pourraient nécessiter des ajustements afin d'améliorer la clarté et de garantir que les résultats correspondent aux besoins spécifiques.

Une approche Bag of Words a été adoptée avec l'utilisation de TF-IDF Vectorizer. Cette technique a permis de représenter la structure du texte en attribuant des poids aux termes selon leur importance dans le corpus.

Les 10 mots les plus représentatifs par topic:

```
Topic 0:  
loop var object display width convert code table really mode  
Topic 1:  
object browser length ios include like landscape stack really lot  
Topic 2:  
http works error install errno running play inline command loading  
Topic 3:  
mac sql server files cookie line code text python implement  
Topic 4:  
array global debug app matrix url spaces elements resttemplate main  
Topic 5:  
object node like javascript jquery nodemodules used example starting web  
Topic 6:  
swift mysql authentication database model save value user float def  
Topic 7:  
data message spring navigation guess current default debug python value  
Topic 8:  
return error file import class using method type use code  
Topic 9:  
point currently string users app box order model django pages
```



Présentation des thèmes cachés dans le texte  
grâce à pyLDavis, basé sur l'algorithme LDA et  
ce avec un interface interactive.

Le jeu de données a été splitté 70/30

Nombre optimal de sujet: 3

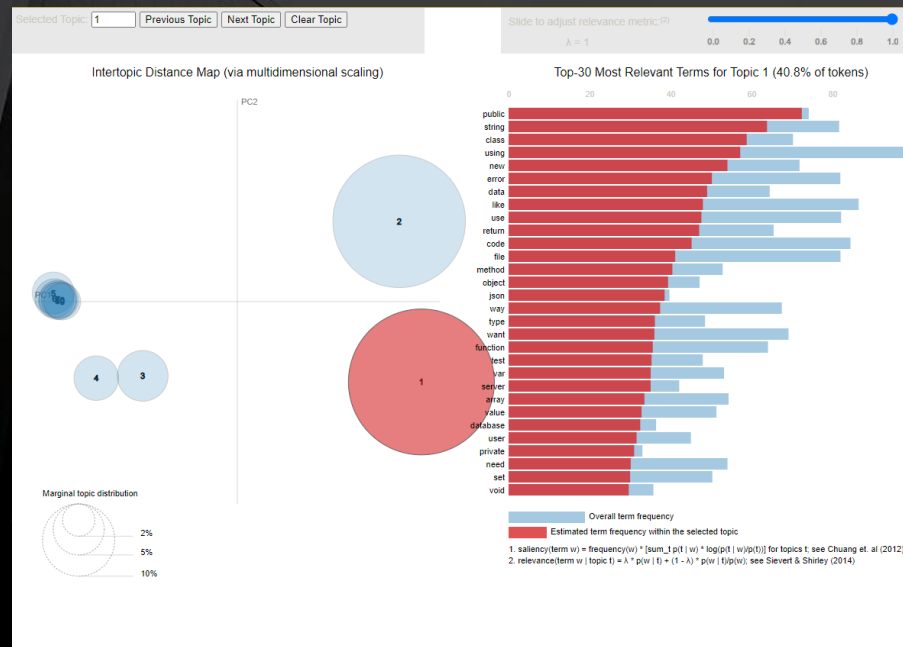
Moyenne des scores de Jaccard :

20K lignes: 0.0

10K lignes: 0.0

1000 lignes: 0.0

300 lignes: 0.008















Utilisation de MultiLabelBinarizer pour encoder les tags, chaque modèle de classification dédié prédit la présence ou l'absence d'une catégorie spécifique.

Les données sont divisées en ensembles d'entraînement et de test (70/30) avec une stratégie de fractionnement cohérente.

L'utilisation de TF-IDF (Term Frequency-Inverse Document Frequency) est appliquée pour la vectorisation du texte, améliorant la représentation des mots dans le modèle.

Les modèles incluent Logistic Regression, Decision Tree, Random Forest, et XGBoost. La recherche de grille est utilisée pour trouver les meilleurs paramètres pour chaque modèle, optimisant ainsi les performances du système de prédiction de tags.

Utilisation de Mlflow pour le suivi de performance:

Run Name	Created 	Duration	Indice de Jaccard
  luxuriant-dog-893	 22 hours ago	10.6h	-
 XGBClassifier Rows: ...	 20 hours ago	8.4h	0.5665309523809524
 RandomForestClassi...	 21 hours ago	48.3min	0.34769484126984124
 DecisionTreeClassifi...	 21 hours ago	17.3min	0.489093253968254
 LogisticRegression R...	 22 hours ago	1.1h	0.5564361111111111

Word2Vec: Accuracy 0.55 Jaccard 0.23

Bert: Accuracy 0.08 Jaccard 0.24

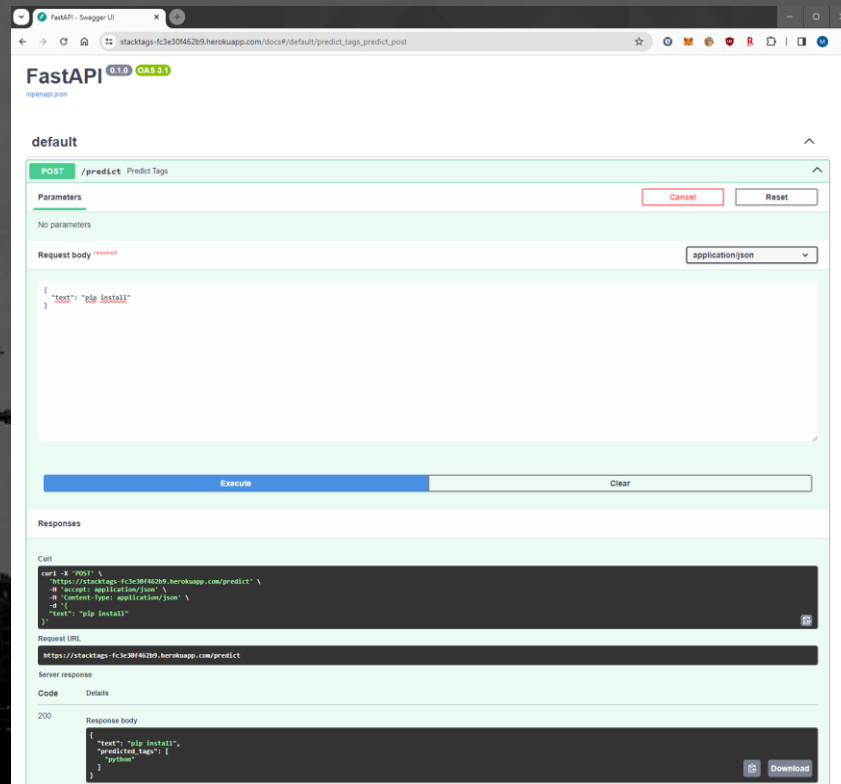
USE: Accuracy 0.06 Jaccard 0.27

# 06 API

Création de l'API avec FastAPI

Déploiement de l'API sur Heroku:

<https://stacktags-fc3e30f462b9.herokuapp.com/>



# 06 API

Développement du code dans l'environnement VS Code avec l'utilisation de Git et GitHub pour assurer une gestion efficace des versions.

Mise en place d'une exécution automatique des tests unitaires lors du déploiement continu grâce à GitHub Actions.

Lien Github:

<https://github.com/MARQUERM/p5>

master

1 Branch

0 Tags

Go to file

Add file

Code

MARQUERM

mise de cote de predict

262e7a1 · 36 minutes ago

69 Commits

.github/workflows	mise de cote de predict	36 minutes ago
__pycache__	maj payload	20 hours ago
a_supprimer	mise de cote de predict	36 minutes ago
img	init project	2 weeks ago
mlruns/0	master	2 days ago
tests	maj	yesterday
.gitignore	A	2 weeks ago
MARQUER_Matthieu_1_notebook_exploration_...	Maj def	5 days ago
MARQUER_Matthieu_2_notebook_requete_API...	A	last week
MARQUER_Matthieu_3_notebook_approche_n...	A	last week
MARQUER_Matthieu_4_notebook_approche_s...	master	2 days ago
MARQUER_Matthieu_5_code_API_122023.ipynb	init project	2 weeks ago

# 06 API

## Commit

### All workflows

Showing runs from all workflows

#### 25 workflow runs

##### mise de cote de predict

Python Tests #22: Commit [262e7a1](#) pushed by MARQUERM

master

1 minute ago  
In progress

...

##### maj payload

Python Tests #21: Commit [6291c74](#) pushed by MARQUERM

master

20 hours ago  
1m 47s

...

##### add payload

Python Tests #20: Commit [6ac4787](#) pushed by MARQUERM

master

20 hours ago  
2m 16s

...

##### modif result

Python Tests #19: Commit [b7caf90](#) pushed by MARQUERM

master

yesterday  
1m 43s

...

##### maj

Python Tests #18: Commit [181105d](#) pushed by MARQUERM

master

yesterday  
1m 46s

...

##### parent dir

Python Tests #17: Commit [6cd2f04](#) pushed by MARQUERM

master

yesterday  
1m 45s

...

### Commits

master

All users

All time

Commits on Feb 14, 2024

#### mise de cote de predict

MARQUERM committed 17 minutes ago · 1 / 1

262e7a1

<>

Commits on Feb 13, 2024

#### maj payload

MARQUERM committed 20 hours ago · 1 / 1

6291c74

<>

#### add payload

MARQUERM committed 20 hours ago · 1 / 1

6ac4787

<>

#### modif result

MARQUERM committed yesterday · 1 / 1

b7caf90

<>

#### maj

MARQUERM committed yesterday · 1 / 1

181105d

<>

# 07 Conclusion

Le choix final s'est porté sur la régression logistique avec TF-idf en mode supervisé en raison de ses performances.

Le modèle a atteint un score Jaccard de 56% (dans le contexte de la proposition de tags), surpassant ainsi d'autres modèles tout en optimisant le temps de calcul.

Merci pour votre attention

Thanks

Avez-vous des questions?

