

# Catégorisez automatiquement des questions



stackoverflow

**Our Q&A community for professional  
and enthusiast programmers**

1

vote

2

answers

19 views

**LINQ to Entities does not recognize the method 'System.DateTime  
GetValueOrDefault()'**

Struggling with very simple code that isn't working where similar code is working in other classes. It won't compile if I remove GetValueOrDefault(). Also I am using System.Linq. I'm getting this ...

c# .net linq

asked 11 mins ago



jfk

6 • 1

-4

votes

0

answers

19 views

**How can differentiate INTO and LET in LINQ C# programming [on hold]**

I'm looking for better example to understand "into and let" functionality in linq C# programming technique. Eg:  
var em = from e in emp group e by new { e.Salary, e.Id } ...

c# .net sql-server linq c#-3.0

asked 30 mins ago



Nagarajan S

35 • 1 • 4

PRÉDICTION DE TAGS

# Plan

## Introduction

## Traitement des données

- Présentation des données
- Préparation des données
- Features engineering

## Modélisation

- Apprentissage non supervisé
- Apprentissage supervisé
- Apprentissage semi-supervisé

## Présentation des résultats

## Déploiement du modèle

## Conclusion

# Introduction



stackoverflow

Our Q&A community for professional  
and enthusiast programmers

- Stack Overflow est un site célèbre de questions-réponses liées au développement informatique.

stackoverflow Questions Tags Users Badges Unanswered

Tagged Questions newest 1 featured faq votes active unanswered

83 votes 13 answers 57k views  
Pass a PHP string to a Javascript variable (including escaping newlines)  
What is the easiest way to encode a PHP string for output to a Javascript variable? I have a PHP string which includes quotes and newlines. I need the contents of this string to be put into a ...  
php javascript escaping newline  
asked Oct 3 '08 at 18:27 David Laing 1,642 4 15 29

55 votes 25 answers 70k views  
How can I compare two sets of 1000 numbers against each other?  
I must check approximately 1000 numbers against 1000 other numbers. I loaded both and compared them server-side: `foreach( $numbers1 as $n1 ) { foreach( $numbers2 as $n2 ) { if( $n1 == $n2 ) { ...`  
php javascript sql algorithm  
asked Oct 15 '10 at 13:15 baklap 692 1 5 15

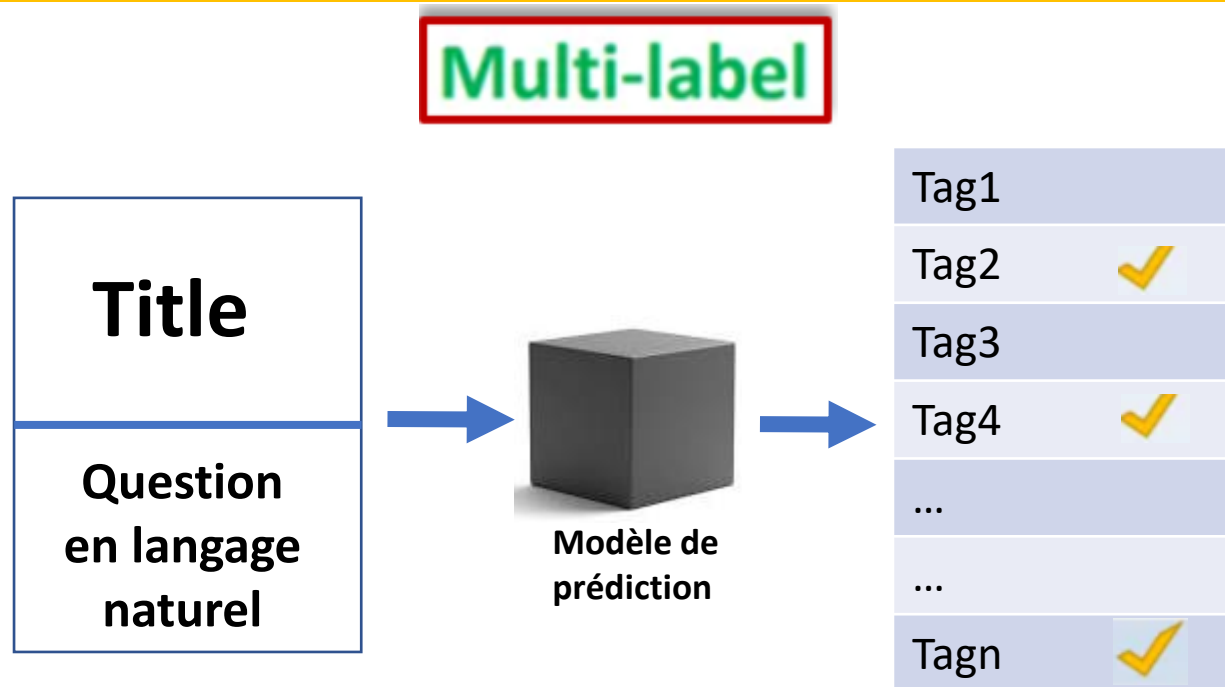
- Chaque question est constituée de trois parties:

➤ Le titre

➤ L'intitulé de la question

➤ Les tags

# ❑ Introduction



- ✔ Notre objectif est de proposer un système de suggestion de « **tags** » associés à chaque post.

# ❑ Traitement des données

## ○ Présentation des données

StackExchange Data Explorer

Home Queries Users Compose Query

Viewing Query

Enter a title for your query

edit description

stackoverflow  
Q&A for professional and enthusiast programmers

Database Schema

Posts	
Id	int
PostTypeId	tinyint
AcceptedAnswerId	int
ParentId	int
CreationDate	datetime
DeletionDate	datetime
Score	int
ViewCount	int
Body	nvarchar (max)

Revisions

Waiting for you to make your first edit...

✓ Requêtes SQL sur « StackExchange Data Explorer »

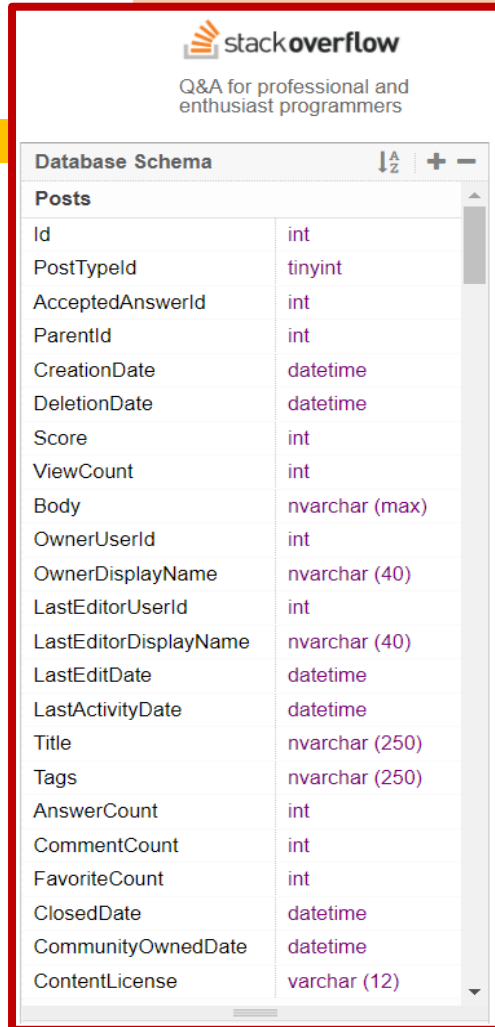
**SELECT FROM posts WHERE Id  
BETWEEN value1 AND value2;**

✓ Il y a une limite sur le temps  
d'exécution de chaque requête SQL

✓ Pour récupérer plus de données,  
nous allons exécuter notre requête 21  
fois en changeant à chaque fois les  
valeurs « valeur1 » et « valeur2 »

# ❑ Traitement des données

## ○ Présentation des données

A screenshot of the Stack Overflow database schema, showing a list of columns and their data types. The table is titled 'Database Schema' and has a search bar and a plus/minus icon. The columns are listed in two columns: the first column contains the column names, and the second column contains the data types. The data types are color-coded: 'int' is purple, 'datetime' is pink, 'nvarchar' is green, and 'varchar' is blue.

Database Schema	
Posts	
Id	int
PostTypeId	tinyint
AcceptedAnswerId	int
ParentId	int
CreationDate	datetime
DeletionDate	datetime
Score	int
ViewCount	int
Body	nvarchar (max)
OwnerUserId	int
OwnerDisplayName	nvarchar (40)
LastEditorUserId	int
LastEditorDisplayName	nvarchar (40)
LastEditDate	datetime
LastActivityDate	datetime
Title	nvarchar (250)
Tags	nvarchar (250)
AnswerCount	int
CommentCount	int
FavoriteCount	int
ClosedDate	datetime
CommunityOwnedDate	datetime
ContentLicense	varchar (12)

- ✔ Au total 21 fichiers csv ont été téléchargés de StackExchange
- ✔ Chaque fichier comporte 23 variables
- ✔ Le nombre de postes moyen par fichier est de 15280

# ❑ Traitement des données

## ○ Préparation des données

✓ Concatenation des datasets et selection des variables d'intérêt.

- ✓ 552487 questions
- ✓ 3 Variables:
  - Poste title
  - Poste body
  - Tag list

✓ Tirage aléatoire de 200 000 questions pour notre étude

✓ Concaténation title et body du poste

✓ Nettoyage du posts

✓ Tokenization et suppression stopwords

✓ Construction bigrams/trigrams

✓ Lemmatization

BeautifulSoup

gensim

spaCy

Text Brut

Tokens



# ❑ Traitement des données

## ○ Préparation des données

	Text	Tags
337049	[make, shrinkable, scrollbar, want, notice, sh...]	[html, css, scrollbars]
478707	[prefer, var, var, -PRON-, have, program, cder...]	[c, perl, coding-style]
191651	[sql, serverreporte, services, export, report,...]	[c#, asp.net, reporting-services]
228069	[wix, register, new, isapi, extension, script,...]	[installation, wix, windows-installer, isapi]
26946	[c, tetris, clone, can, not, get, block, respo...]	[c#, keydown, arrow-keys]
427083	[autorotate, view, uidevice, interface, notifi...]	[iphone, ipad, uiviewcontroller]
231319	[set, current, user, name, sharepoint, use, sh...]	[c#, asp.net, .net, visual-studio-2008, sharep...]
444393	[bidi, association, nhibernate, mapping, class...]	[c#, nhibernate, linq-to-nhibernate]
464623	[textbox, anchor, form, side, display, properl...]	[visual-studio-2005, user-controls, .net-2.0, ...]
454863	[gridview, sort, challenge, move, winform, asp...]	[asp.net, vb.net, gridview]

Séparation du  
jeu de données

- ✔ Train set (60% du jeu de données)
  - Entrainement des modèles de prediction
  - Optimisation des hyper-paramètres
- ✔ Validation set (20% du jeu de données)
  - Sélection du meilleur modèle
- ✔ Test set (20% du jeu de données)
  - Évaluation du meilleur modèle

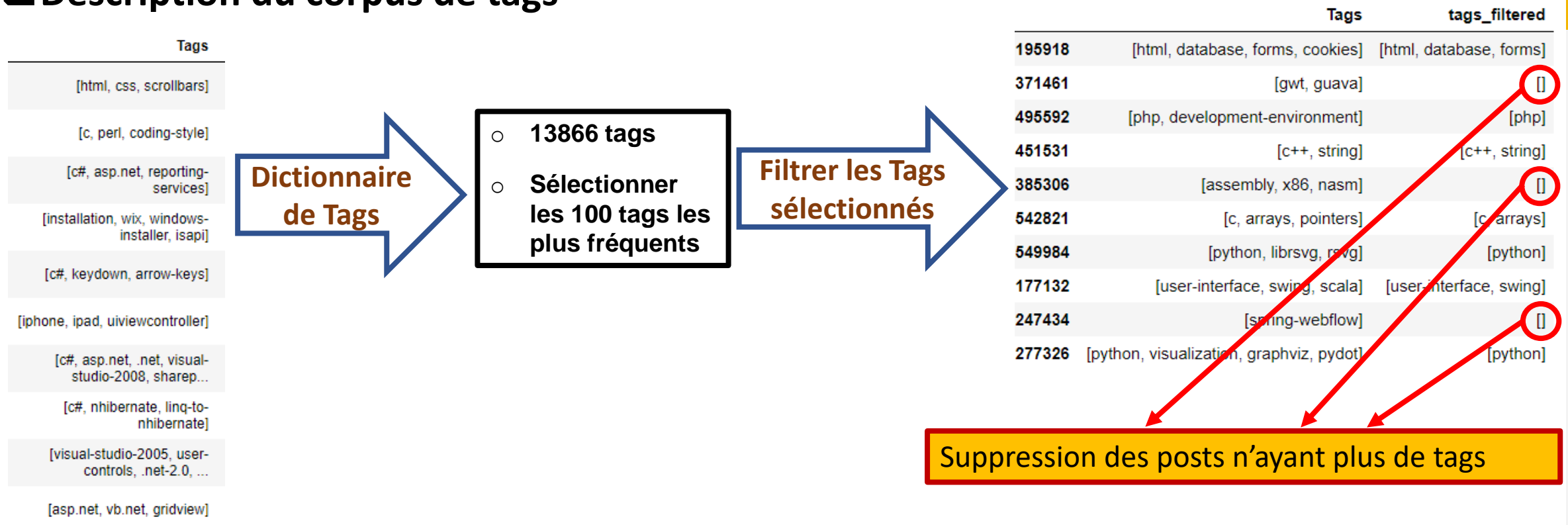


# Traitement des données

## ○ Préparation des données

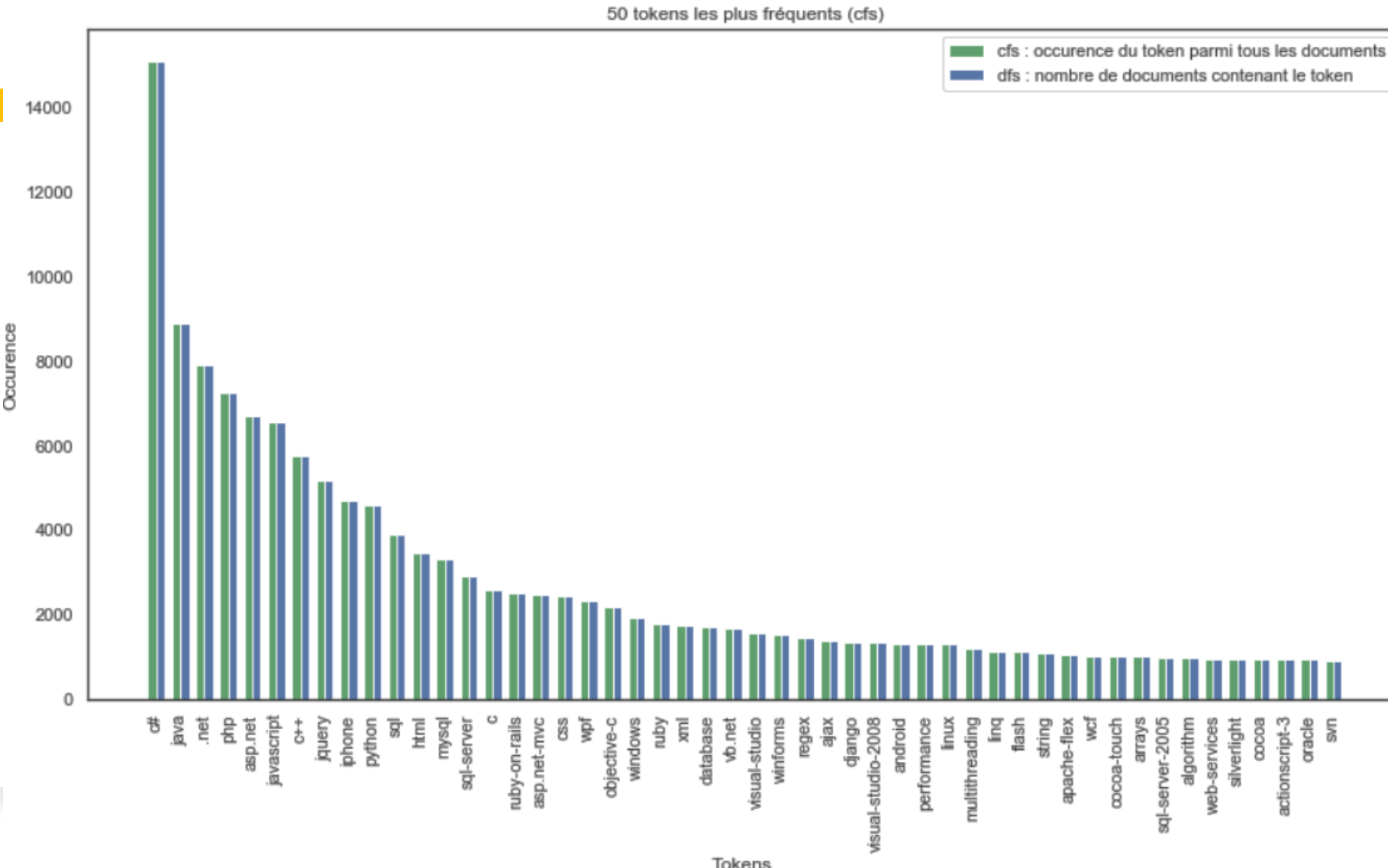
Dans cette partie nous allons procéder à l'analyse du corpus du **training set**

### T Description du corpus de tags



# □ Traitement des données

## ○ Préparation des données



✓ Pratiquement chaque token est employé une et une seule fois dans chaque tag.

# □ Traitement des données

## ○ Préparation des données

### □ Description du corpus de post

Text

[make, shrinkable, scrollbar,  
want, notice, sh...

[prefer, var, var, -PRON-, have,  
program, cder...

[sql, serverreporte, services,  
export, report,...

[wix, register, new, isapi,  
extension, script,...

[c, tetris, clone, can, not, get,  
block, respo...

[autorotate, view, uidevice,  
interface, notifi...

**Dictionnaire  
de posts**

○ 238700  
tokens

- Sélection des tokens porteurs de sens:
  - Moins de 1 post sur 10
  - Plus de 50 occurrences

1

○ 4187  
tokens

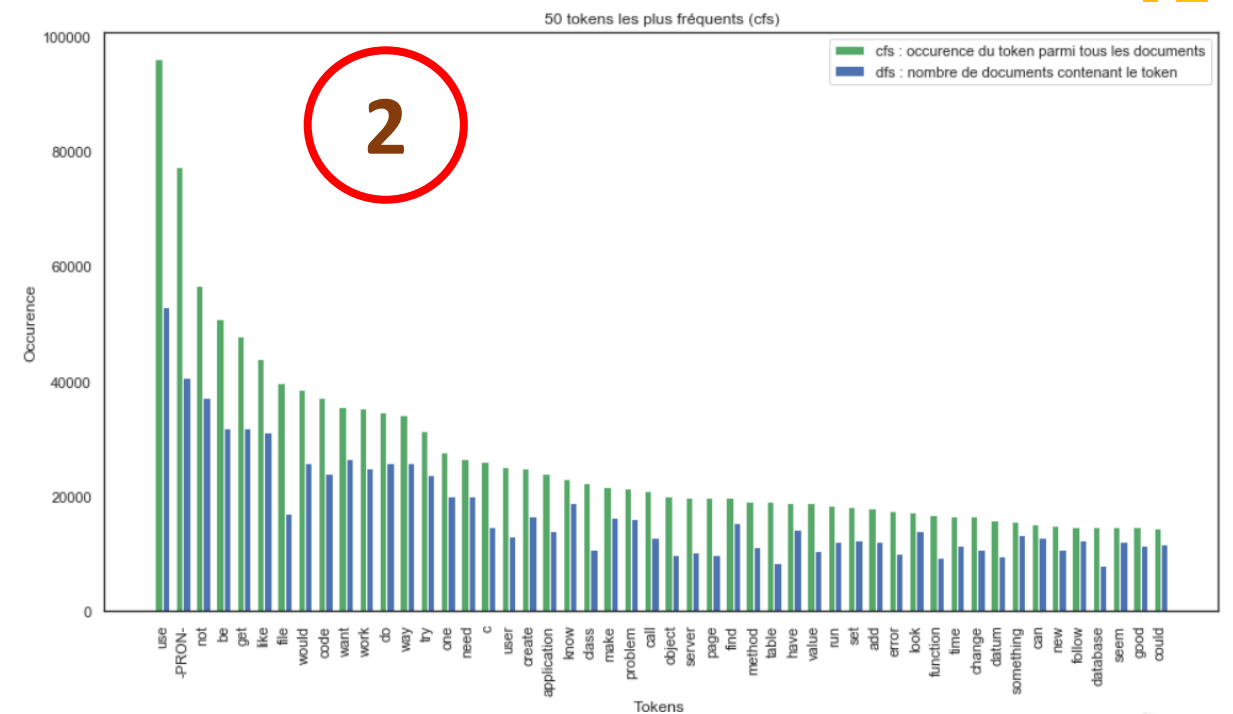
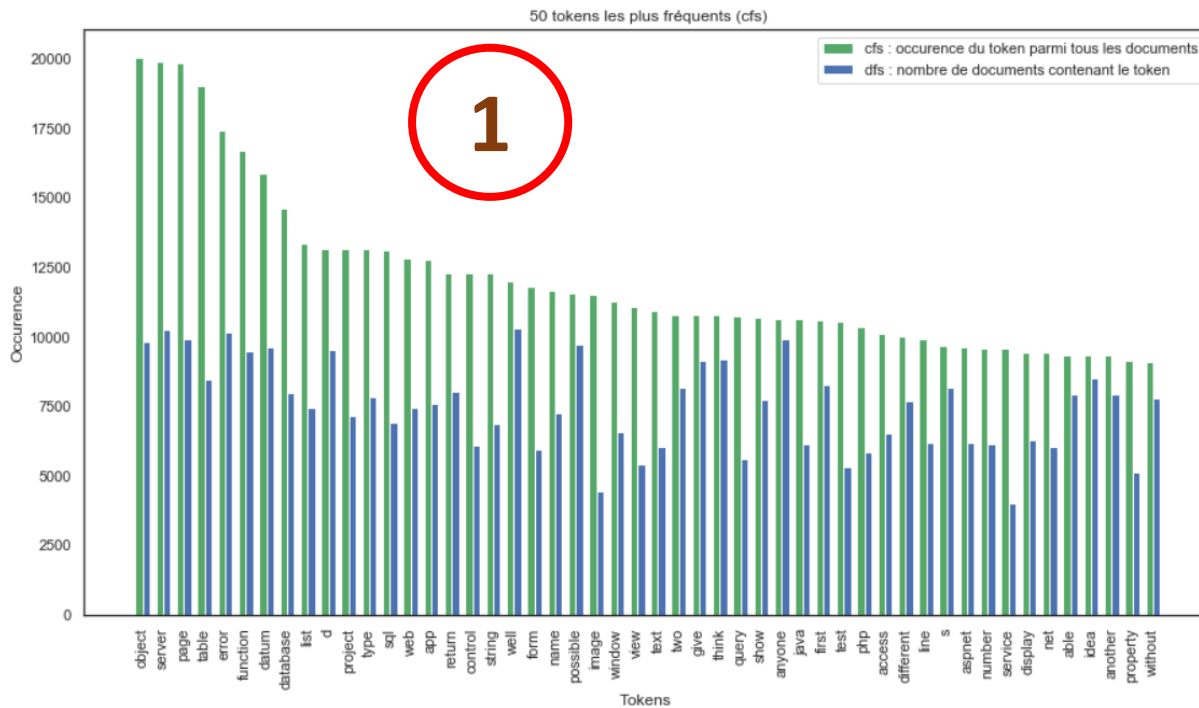
- conserver les relations entre les tokens composant le post (utile pour l'apprentissage non supervisé)

2

○ 7862  
tokens

# □ Traitement des données

## ○ Préparation des données



# ❑ Traitement des données

## ○ Features engineering

### ❑ Document embedding

- ✔ Trouvez des représentations numériques pour des documents entiers.
- ✔ Représenter le document dans un espace vectoriel de dimensions  $n$
- ✔ Transforme une entrée de taille variable en sortie à taille fixe



# ❑ Traitement des données

## ○ Features engineering

- ✔ Find numerical representations for whole documents

### ❑ Bag of words (bow)

**Corpus** {

<b>Dictionnaire</b>				
	Token 1	Token 2	...	Token t
Post 1	1	0	...	1
Post 2	0	3	...	0
...	...	...	...	...
Post m	1	1	...	1

Raw Text	Bag-of-words vector
it is a puppy and it is extremely cute	it 2
	they 0
	puppy 1
	and 1
	cat 0
	aardvark 0
	cute 1
	extremely 1
	... ...

# ❑ Traitement des données

## ○ Features engineering

### ❑ tf-idf

	Token 1	Token 2	...	Token t
Post 1	0.05	0	...	0,6
Post 2	0	0,4	...	0
...	...	...	...	...
Post m	0,2	0,23	...	0.7

Ligne BoW /  
nombre de  
token

||  
tf

Colonne BoW / idf

**Idf** =  $\log(\text{nombre de posts} / \text{nombre de posts comportant token})$

« Plus de poids aux tokens  
porteurs d'information »

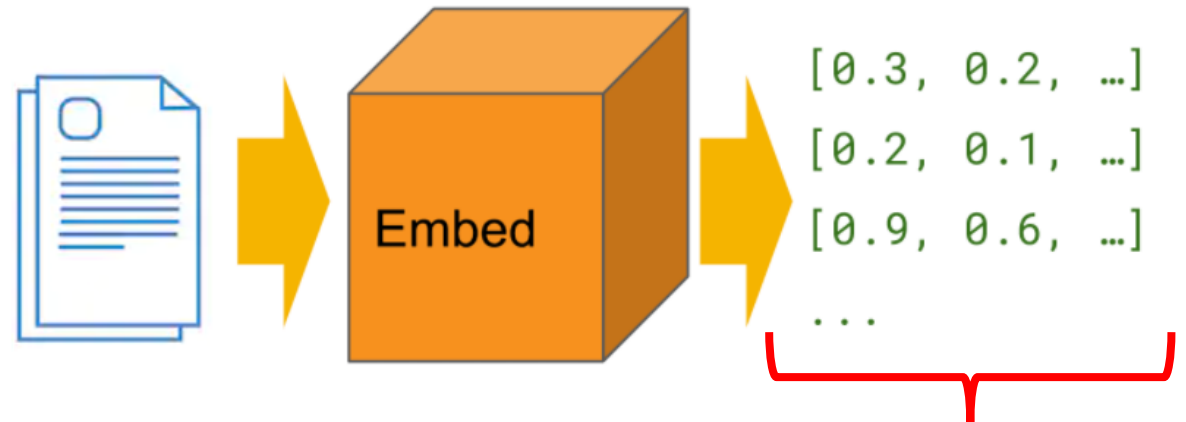


# ❑ Traitement des données

## ○ Features engineering

### ❑ Universal Sentence Encoder

- ✓ Capturer le sens d'un document
- ✓ 2 variantes:
  - Deep Averaging Network (DAN) encoder
  - Transformer encoder
- ✓ L'input est un texte anglais de longueur variable et la sortie est un vecteur à 512 dimensions.



# ❑ **Modélisation**

## ❑ Apprentissage non supervisé (Topics detection)

- ✔ Latent Dirichlet Allocation (LDA)
- ✔ Non-negative Matrix Factorization (NMF)

## ❑ Apprentissage supervisé

- ✔ Régression logistique
- ✔ SVM
- ✔ Random Forest
- ✔ Deep Averaging Network (DAN)
- ✔ Transformer

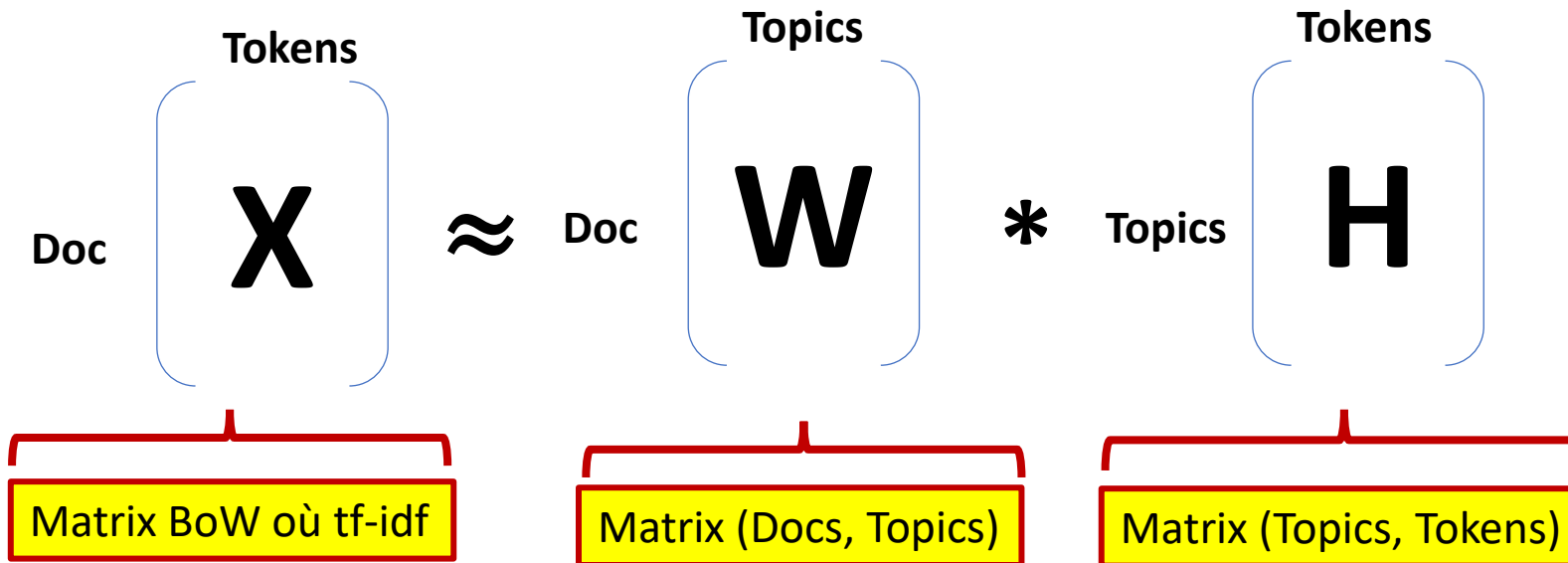
## ❑ Apprentissage semi-supervisé

- ✔ Apprentissage non-supervisé + Apprentissage supervisé
- ✔ Topic detection par LDA +
  - Régression logistique
  - SVM
  - Random Forest

# ❑ Modélisation

## ○ Apprentissage non supervisé

✔ Décomposition des matrix BoW où tf-idf :  $X \approx W * H$



❑ Input:

- ✔ Matrix BoW où tf-idf
- ✔ Nombre de topics à définir

❑ Résultat:

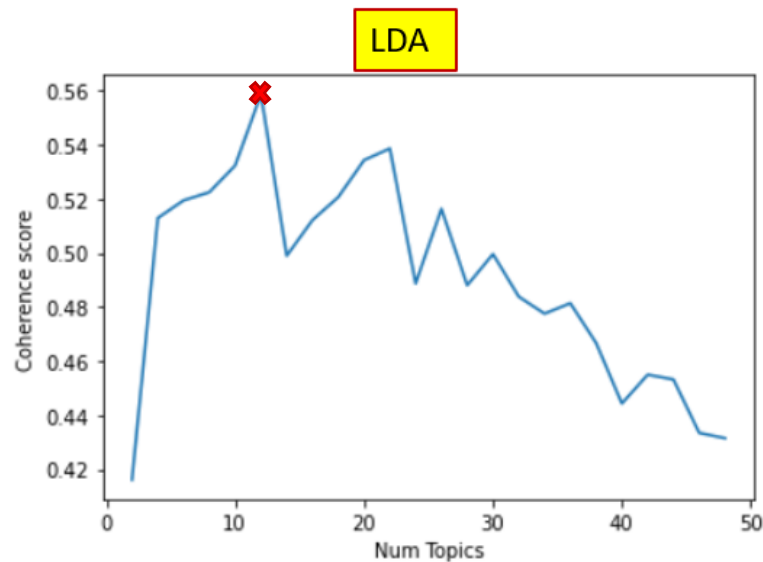
- ✔ Distribution des topics par document
- ✔ Distribution des tokens par topics

# ❑ Modélisation

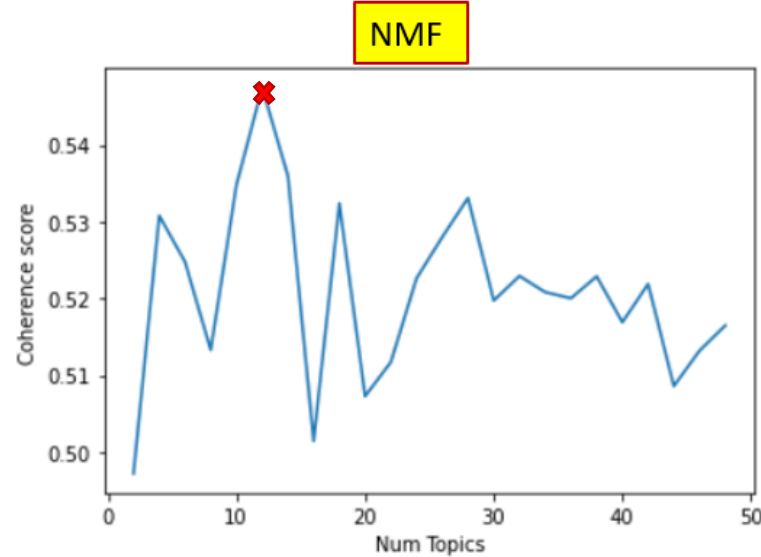
## ○ Apprentissage non supervisé

### ❑ Choix du nombre de topics

LDA et NMF nécessitent de choisir le nombre de topics



12 topics



18 topics

● Le score de coherence évalue la performance intrinsèque de la modélisation de topic.

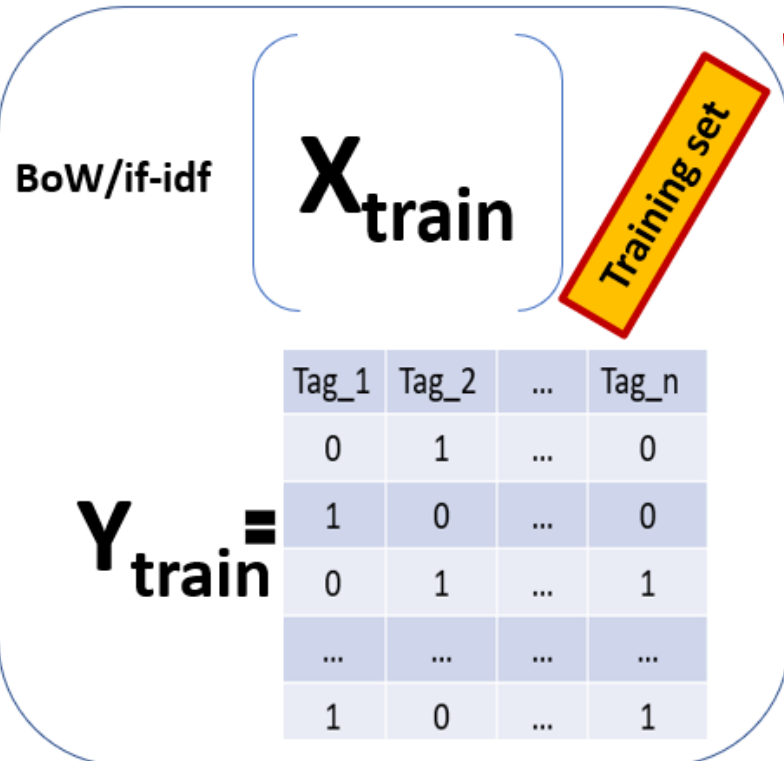
● Un topic est coherent si les tokens les plus représentatifs du topic sont similaires entre eux.

# ❑ Modélisation

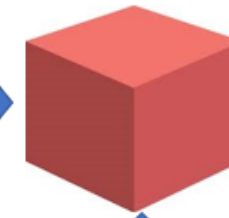
## ○ Apprentissage supervisé

### Algorithme supervisé One versus Rest

- Régression logistique
- SVM
- Random Forest



For training



Output

$Y_{\text{pred}}$

Prediction

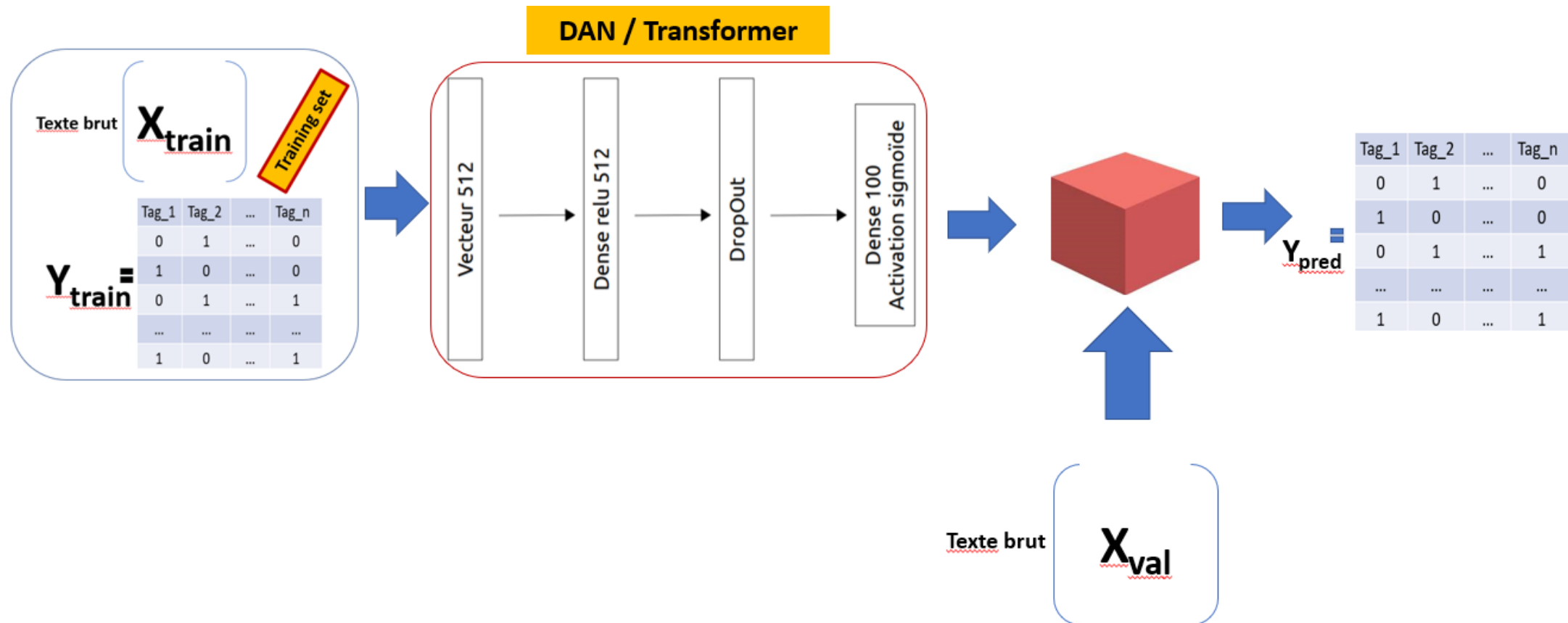
BoW/TF-IDF

$X_{\text{val}}$

# □ Modélisation

## ○ Apprentissage supervisé

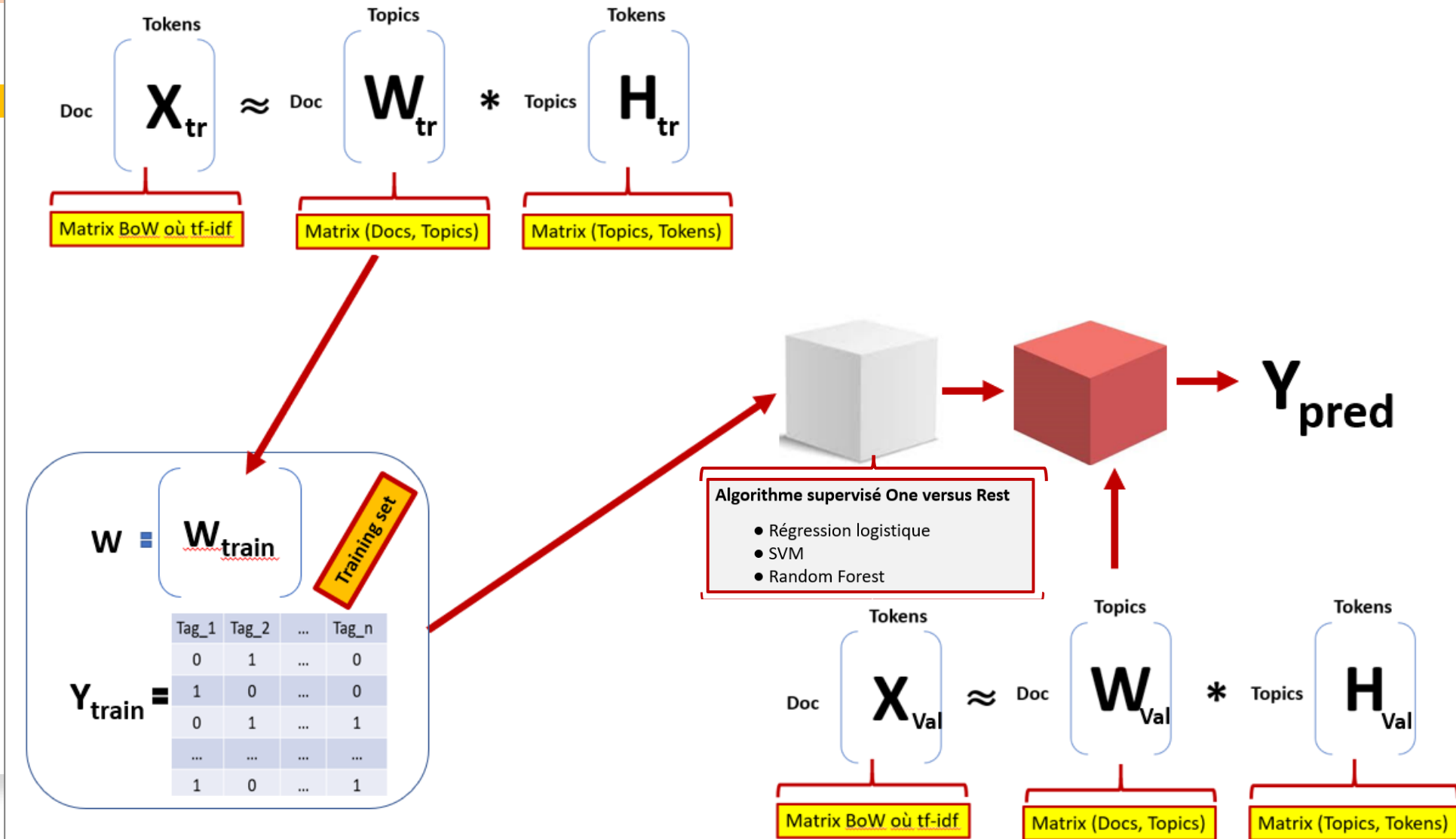
### □ Universal sentence encoder



# ❑ Modélisation

## ○ Apprentissage semi-supervisé

- ❑ Topics detection par LDA puis classification

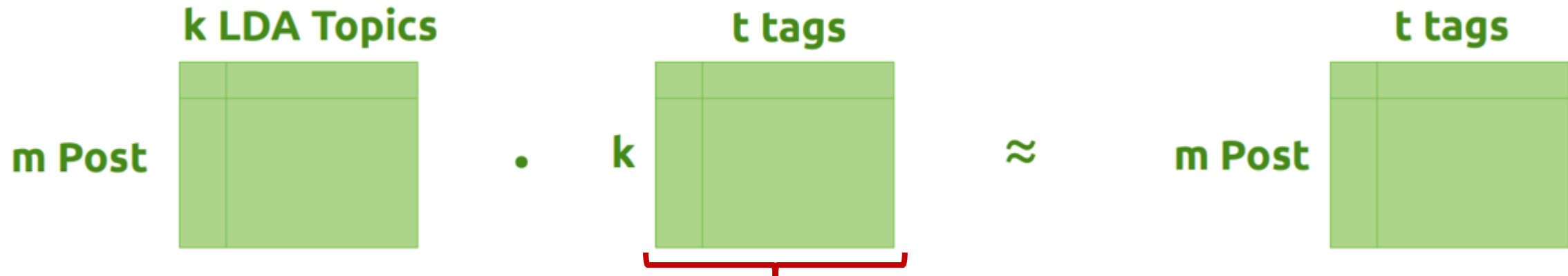




# ❑ Modélisation

## ○ Apprentissage semi-supervisé

### ❑ Matrice de transformation Topics-Tags



$$\text{Matrice de transformation} = \text{inverse}(\text{post-topic}) * (\text{post-tag})$$

# Présentation des résultats

Nombre de tags présents : 56448  
Nombre de tags prédits : 44409

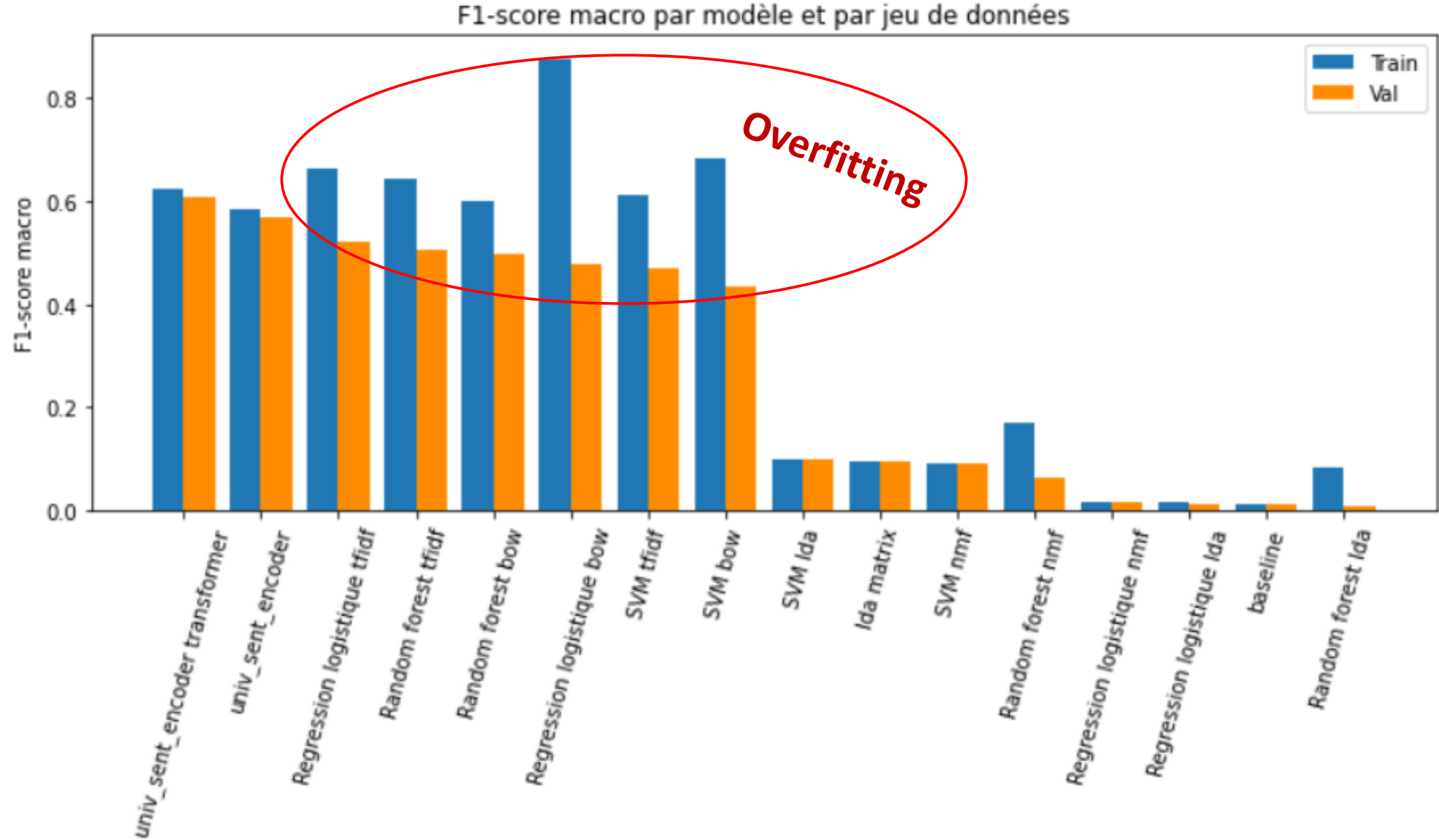
	precision	recall	f1-score	support
.net	0.56	0.25	0.35	2619
actionscrip-3	0.40	0.34	0.37	303
ajax	0.43	0.39	0.41	470
algorithm	0.46	0.41	0.44	333
android	0.84	0.81	0.82	438
apache	0.50	0.51	0.51	218
apache-flex	0.74	0.64	0.69	346
arrays	0.38	0.33	0.35	294
...	...	...	...	...
wpf	0.77	0.75	0.76	753
xaml	0.29	0.25	0.27	150
xcode	0.45	0.43	0.44	185
xml	0.63	0.55	0.58	649
micro avg	0.60	0.47	0.53	56448
macro avg	0.52	0.45	0.48	56448
weighted avg	0.60	0.47	0.52	56448
samples avg	0.53	0.51	0.48	56448

		True condition		
Total population		Condition positive	Condition negative	Prevalence = $\frac{\sum \text{Condition positive}}{\sum \text{Total population}}$
Predicted condition	Predicted condition positive	True positive	False positive, Type I error	Positive predictive value (PPV), Precision = $\frac{\sum \text{True positive}}{\sum \text{Predicted condition positive}}$
	Predicted condition negative	False negative, Type II error	True negative	False omission rate (FOR) = $\frac{\sum \text{False negative}}{\sum \text{Predicted condition negative}}$
		True positive rate (TPR), Recall, Sensitivity, probability of detection, Power = $\frac{\sum \text{True positive}}{\sum \text{Condition positive}}$	False positive rate (FPR), Fall-out, probability of false alarm = $\frac{\sum \text{False positive}}{\sum \text{Condition negative}}$	Positive likelihood ratio (LR+) = $\frac{\text{TPR}}{\text{FPR}}$
		False negative rate (FNR), Miss rate = $\frac{\sum \text{False negative}}{\sum \text{Condition positive}}$	Specificity (SPC), Selectivity, True negative rate (TNR) = $\frac{\sum \text{True negative}}{\sum \text{Condition negative}}$	Negative likelihood ratio (LR-) = $\frac{\text{FNR}}{\text{TNR}}$

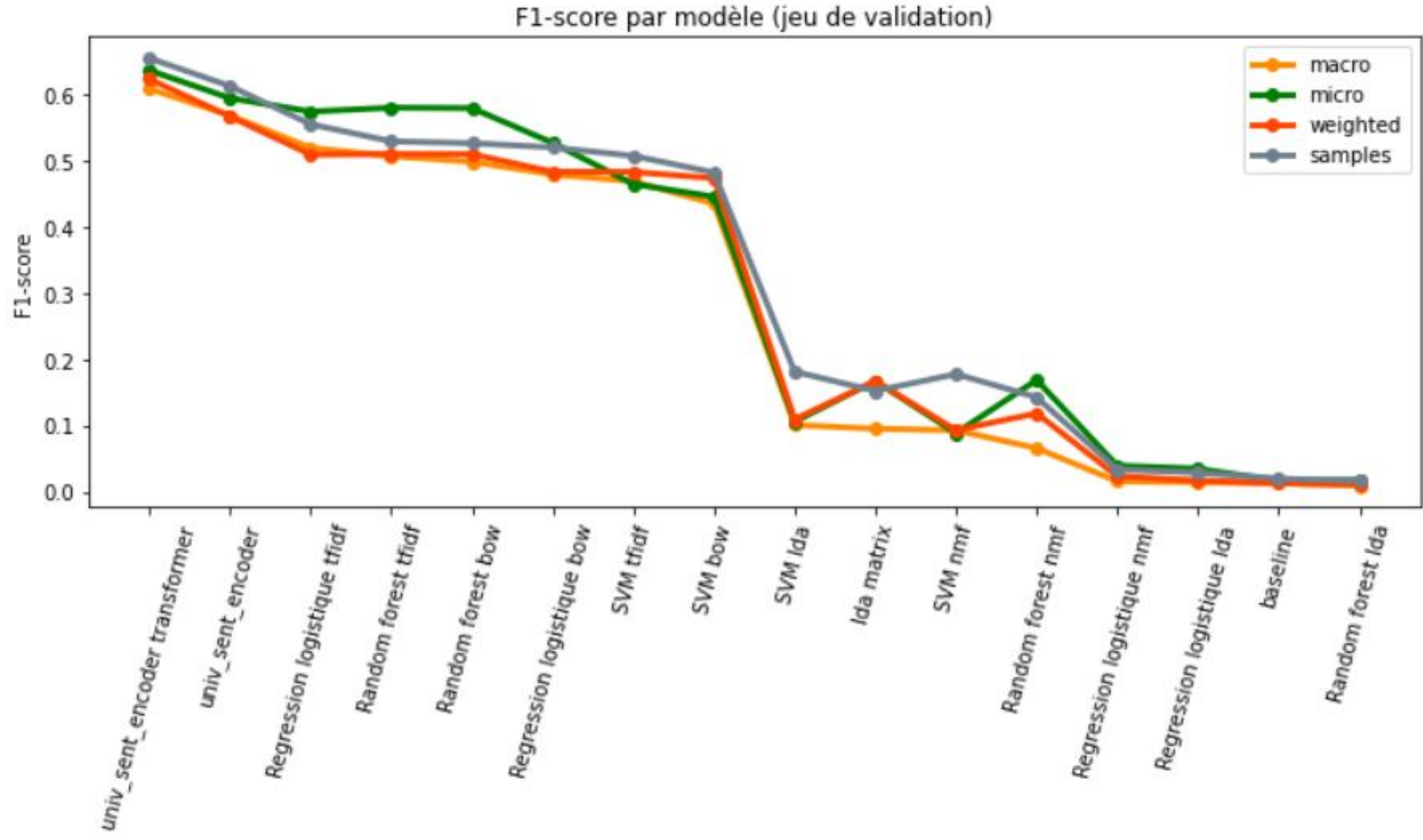
Accuracy (ACC) = $\frac{\sum \text{True positive} + \sum \text{True negative}}{\sum \text{Total population}}$
False discovery rate (FDR) = $\frac{\sum \text{False positive}}{\sum \text{Predicted condition positive}}$
Negative predictive value (NPV) = $\frac{\sum \text{True negative}}{\sum \text{Predicted condition negative}}$
Diagnostic odds ratio (DOR) = $\frac{\text{LR+}}{\text{LR-}}$
F1 score = $2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$

$$Pr = \frac{TP}{TP + FP}$$
$$Pr_{macro} = \frac{Pr_1 + Pr_2 + \dots + Pr_k}{k} = Pr_1 \frac{1}{k} + Pr_2 \frac{1}{k} + \dots + Pr_k \frac{1}{k}$$
$$Pr_{weighted-macro} = Pr_1 \frac{\#Obs_1}{N} + Pr_2 \frac{\#Obs_2}{N} + \dots + Pr_k \frac{\#Obs_k}{N}$$
$$Pr_{micro} = \frac{TP_1 + TP_2 + \dots + TP_k}{(TP_1 + TP_2 + \dots + TP_k) + (FP_1 + FP_2 + \dots + FP_k)}$$

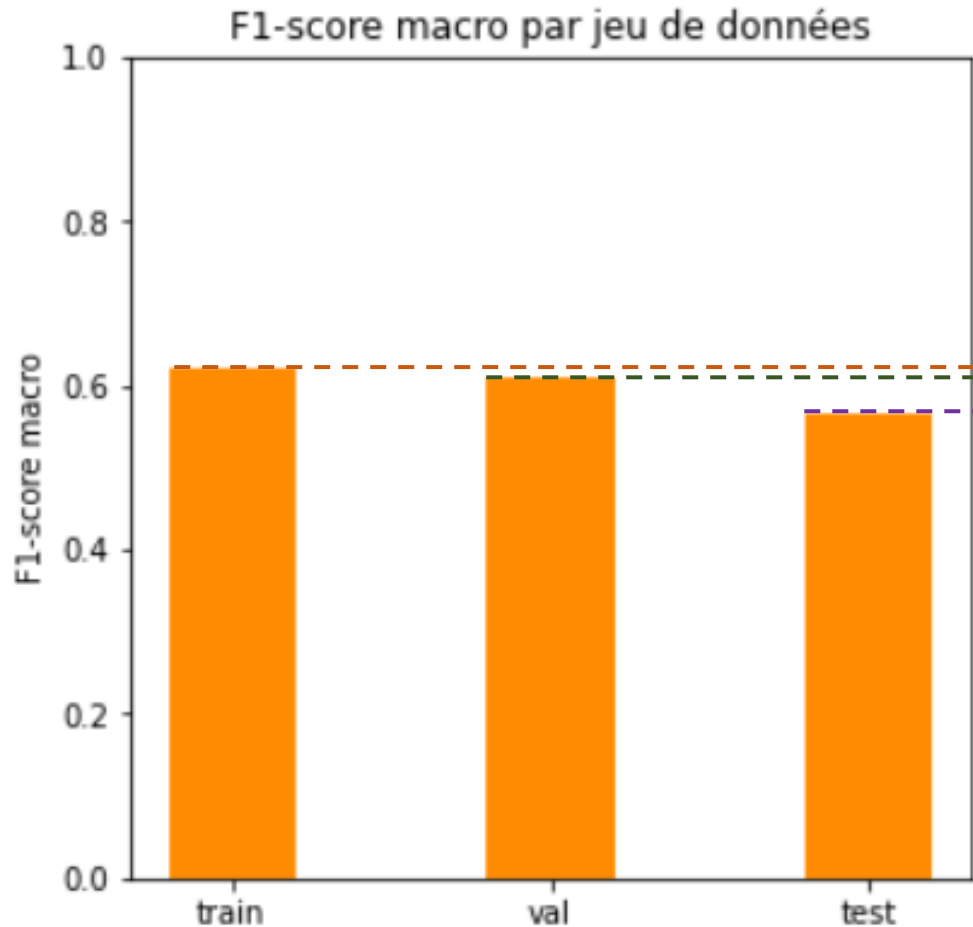
# ❑ Présentation des résultats



# ❑ Présentation des résultats



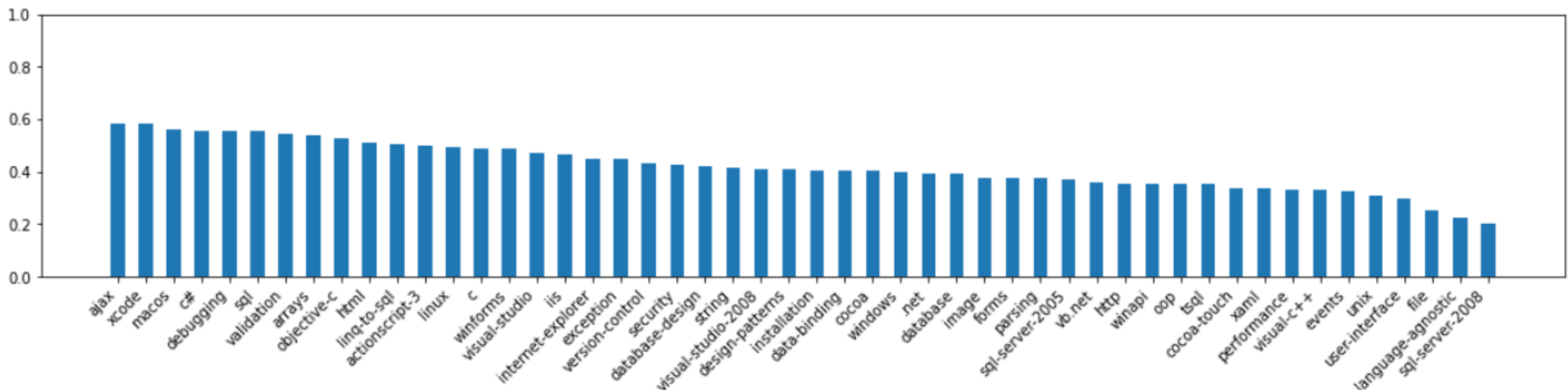
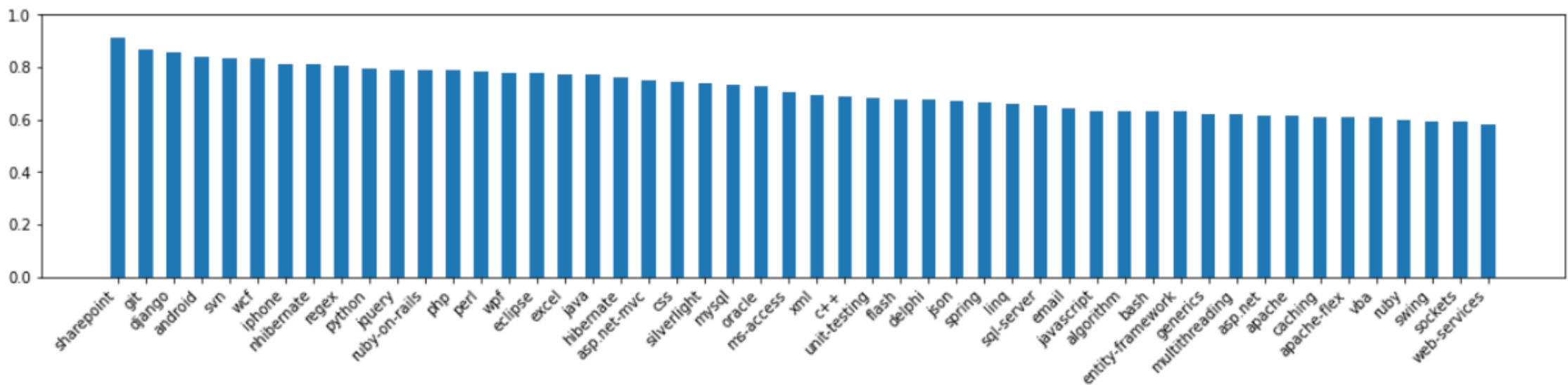
# ❑ Présentation des résultats



- ❑ Pas de sur-apprentissage
- ❑ Performance stable sur jeu de test
- ❑ Le modèle généralise bien !

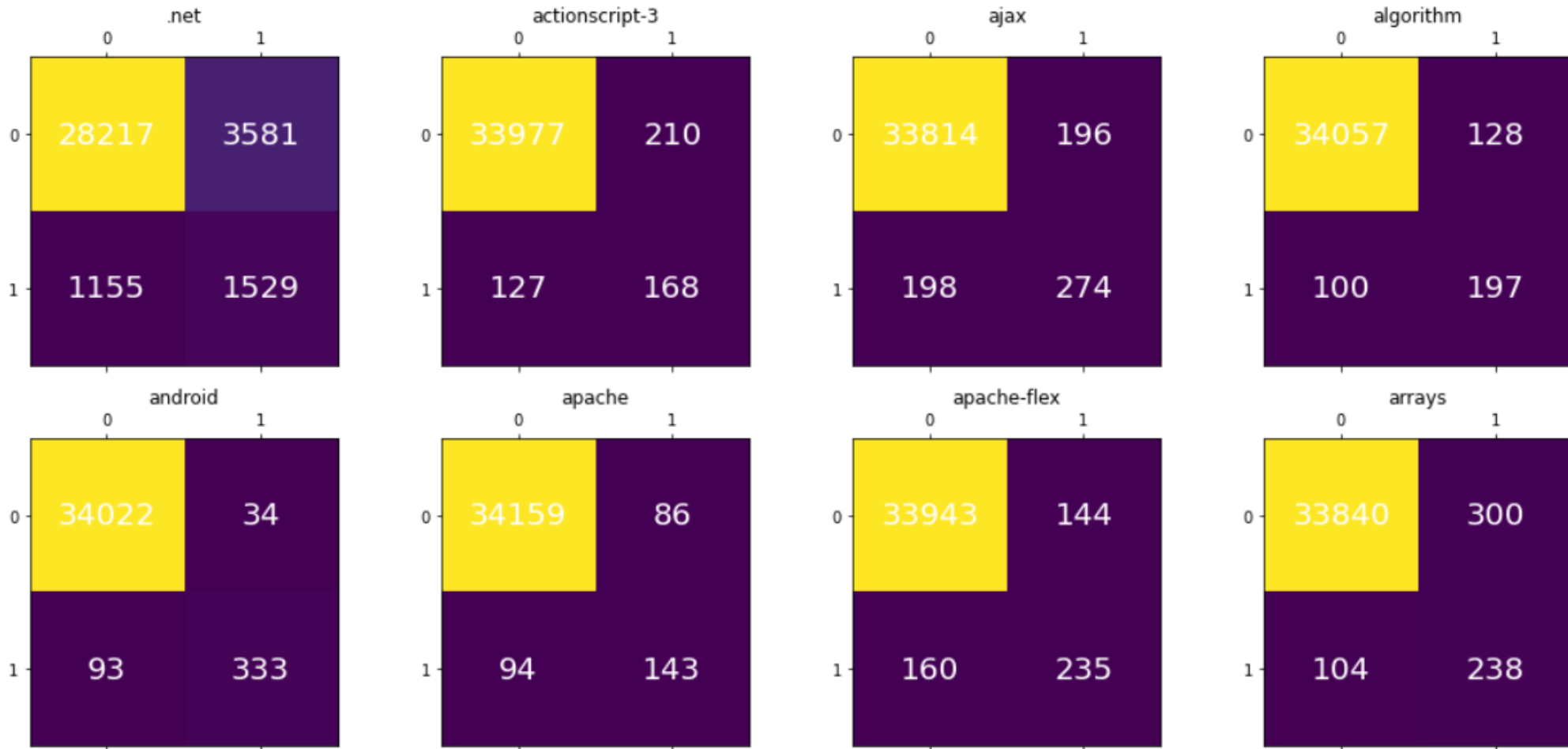
# Présentation des résultats

F1 score macro pour l'ensemble des tag (classé par ordre décroissant)



# ❑ Présentation des résultats

## ✔ Plot 8 first tags confusion matrix





# ❑ Déploiement du modèle

- Déploiement sur Heroku : régression logistique sur tf-idf

<https://stackoverflow-tags-prediction.herokuapp.com/>

Stackoverflow Tags prediction

By Didier ILBOUDO

Please write your Stackoverflow post here!

Submit

Tags :

# ❑ Conclusion

## ❑ Au final :

- ❖ Meilleure performance des modèles (DAN & Transformer) : **F1 score élevé ; modèle généralise bien ;**
- ❖ Au prix d'une demande élevée en puissance de calcul, le modèle DAN est choisi comme meilleur modèle;
- ❖ La limite maximale de la taille du slug sur Heroku (500 MB) nous conduit à deployer le modèle logistique.

# ❑ Conclusion

## ❑ Améliorations envisageables :

- ❖ Séparation stratifiées des jeux de données en vue d'avoir un échantillon représentatif (« **train** », « **val** », « **set** ») ;
- ❖ Prise en compte de la corrélation entre les différents tags lors de l'apprentissage (par exemple le tag « **pandas** » est certainement associé au tag « python ») (scikit-multilearn) ;
- ❖ Évaluation d'autres méthodes de features engineering (« word embedding », « POS »), et d'autres modèles ;
- ❖ Prédire plus de tags ou sélectionner manuellement un nombre limité de tags à prédire (**remplacer** « python-3.x » et « python-2.7 » par « python »).
- ❖ Résoudre le problème de « **imbalanced classification** »

*merci*

– BEACOU –

**POUR VOTRE ATTENTION**