

Catégorisez automatiquement des questions



par Ana Bernal

Mars 2023

OpenClassrooms



Mentor: Samir Tanfous

Programme

- 1** Rappel **mission**
- 2** Pre-processing + exploration
- 3** **Modélisation** + évaluation + choix de modèle
- 4** Développement **API**
- 5** **Conclusions**

1

Mission

Mission

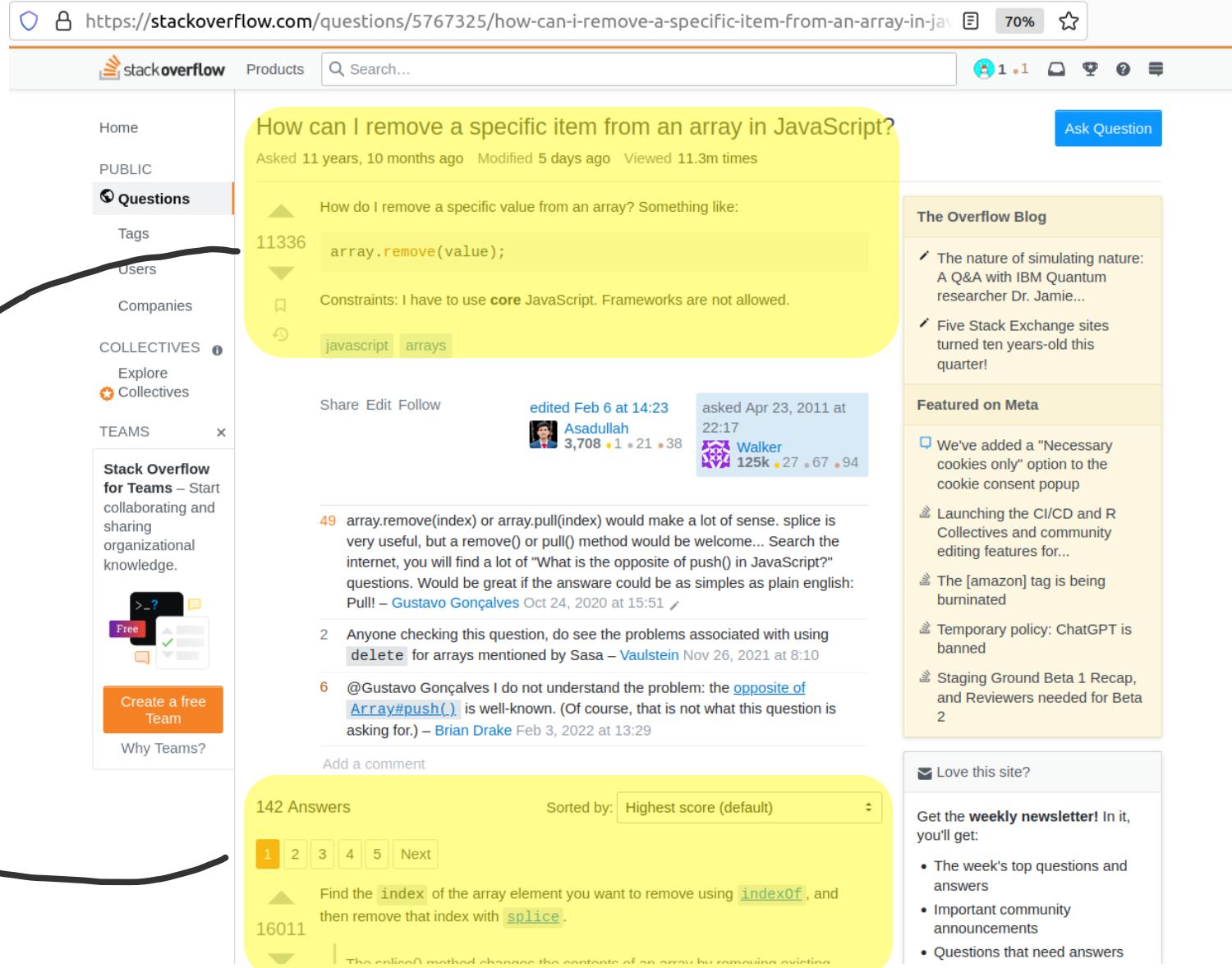
1

À propos de



question

réponses



A screenshot of a Stack Overflow question page. The URL in the address bar is <https://stackoverflow.com/questions/5767325/how-can-i-remove-a-specific-item-from-an-array-in-jav>. The page shows a question titled "How can I remove a specific item from an array in JavaScript?" with 11336 answers. The top answer, by user 11336, suggests using `array.remove(value);`. A constraint is noted: "I have to use core JavaScript. Frameworks are not allowed." The question was asked 11 years, 10 months ago, modified 5 days ago, and viewed 11.3m times. The post was edited on Feb 6 at 14:23 and answered on Apr 23, 2011 at 22:17. The accepted answer is by user Walker with 125k reputation. The sidebar on the left includes links for Home, PUBLIC Questions, Tags, Users, Companies, COLLECTIVES, Explore Collectives, and TEAMS. The right sidebar features "The Overflow Blog" with posts about simulating nature and Stack Exchange sites turning ten, and "Featured on Meta" with news about cookie consent, CI/CD, and temporary policy changes. A "Love this site?" button and a newsletter sign-up are also present.

Mission

1

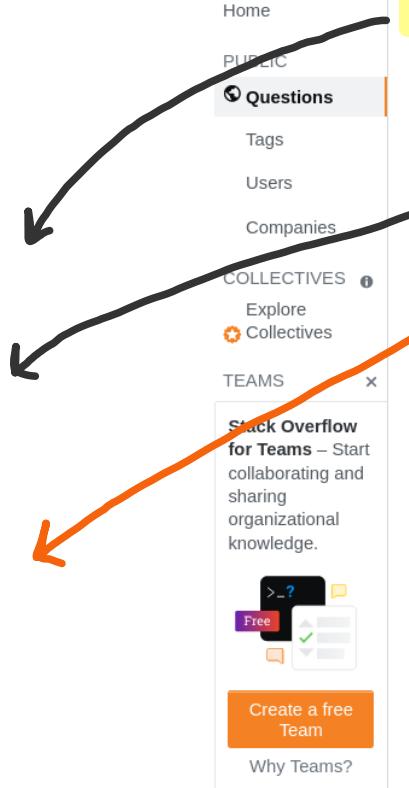
À propos de



titre

corps

étiquettes
(tags)



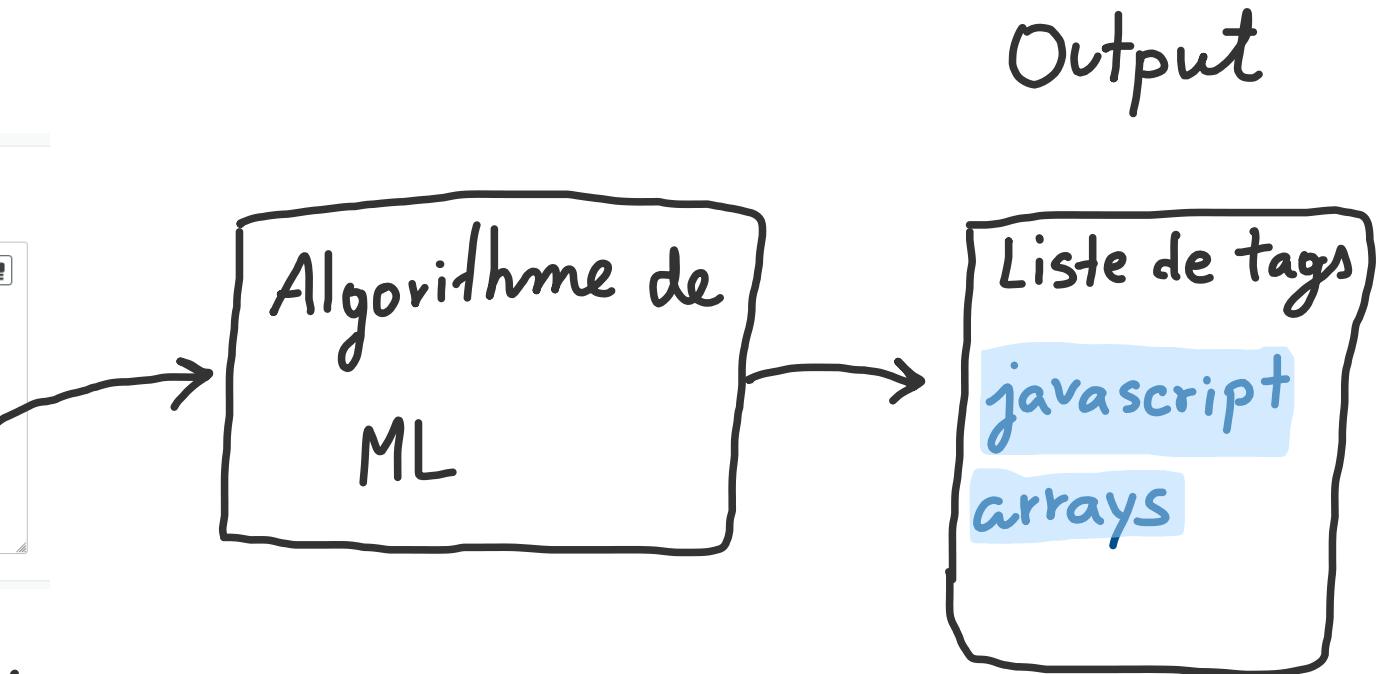
A screenshot of a Stack Overflow question page. The URL in the address bar is <https://stackoverflow.com/questions/5767325/how-can-i-remove-a-specific-item-from-an-array-in-jav>. The page title is "How can I remove a specific item from an array in JavaScript?". The question was asked 11 years, 10 months ago and modified 5 days ago, with 11.3m views. The question text is: "How do I remove a specific value from an array? Something like:
`array.remove(value);`" Constraints: "I have to use core JavaScript. Frameworks are not allowed." The question has 11336 upvotes and 11336 downvotes. It is tagged with `javascript` and `arrays`. The last edit was on Feb 6 at 14:23 by Asadullah, and it was asked on Apr 23, 2011 at 22:17 by Walker. There are 142 answers, sorted by highest score. The first answer suggests using `splice` to find the index of the element and then remove it. The sidebar on the right includes sections for "The Overflow Blog" and "Featured on Meta".

Mission

1

Notre mission:

Input



Texte : question sur programmation (en anglais)

- * Avec API
 - * système gestion version

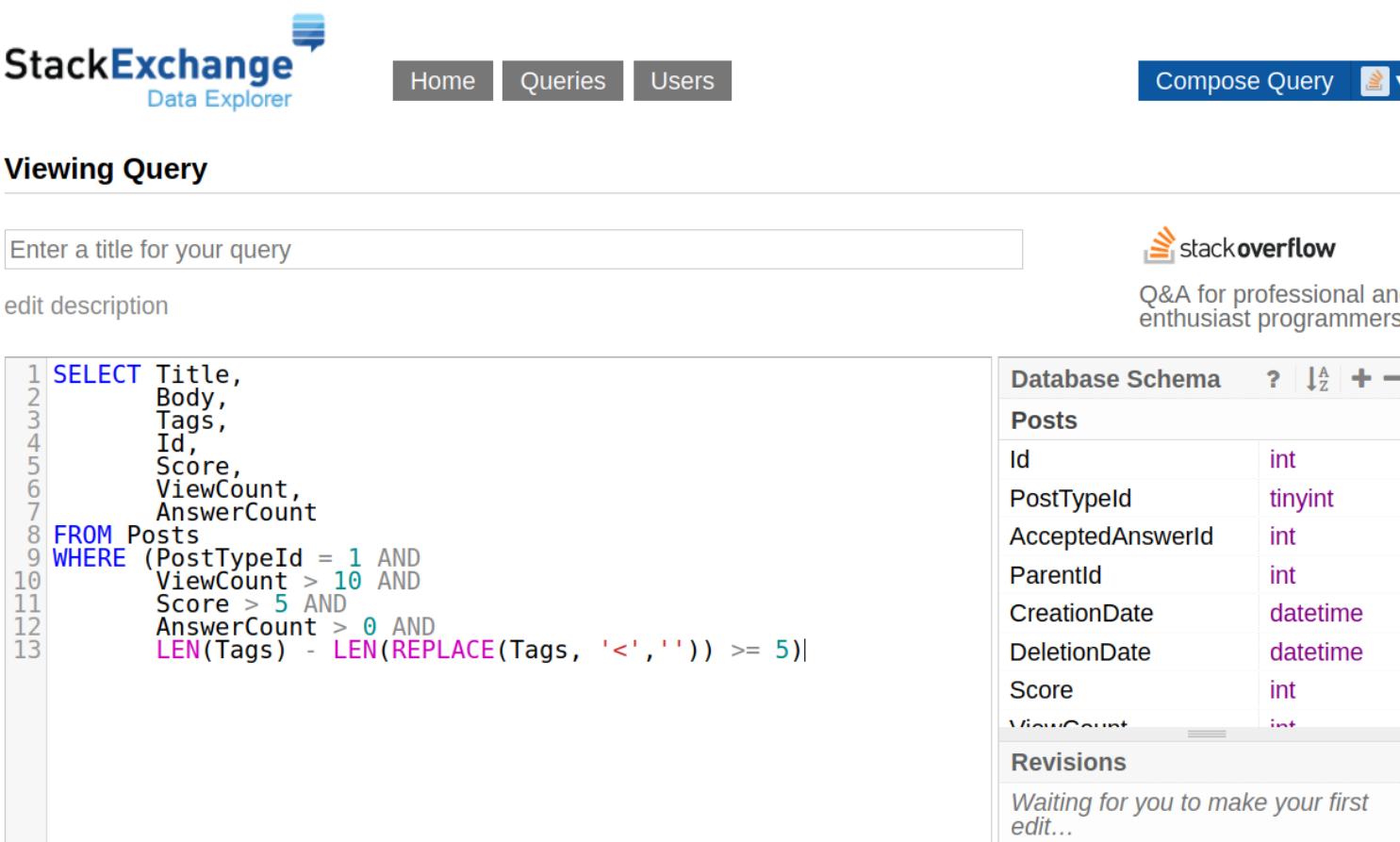


2

Pre-processing

Pre-processing

Obtention des données avec 



The screenshot shows the StackExchange Data Explorer interface. At the top, there's a navigation bar with 'Home', 'Queries', and 'Users' buttons, and a 'Compose Query' button with a dropdown arrow. Below the navigation is a section titled 'Viewing Query' with a text input field for 'Enter a title for your query'. To the right of this is the 'stackoverflow' logo and the text 'Q&A for professional and enthusiast programmers'. The main area contains a SQL query editor with the following code:

```

1 SELECT Title,
2      Body,
3      Tags,
4      Id,
5      Score,
6      ViewCount,
7      AnswerCount
8 FROM Posts
9 WHERE (PostTypeId = 1 AND
10        ViewCount > 10 AND
11        Score > 5 AND
12        AnswerCount > 0 AND
13        LEN(Tags) - LEN(REPLACE(Tags, '<', '')) >= 5)

```

To the right of the query editor is a 'Database Schema' viewer for the 'Posts' table, showing the following columns:

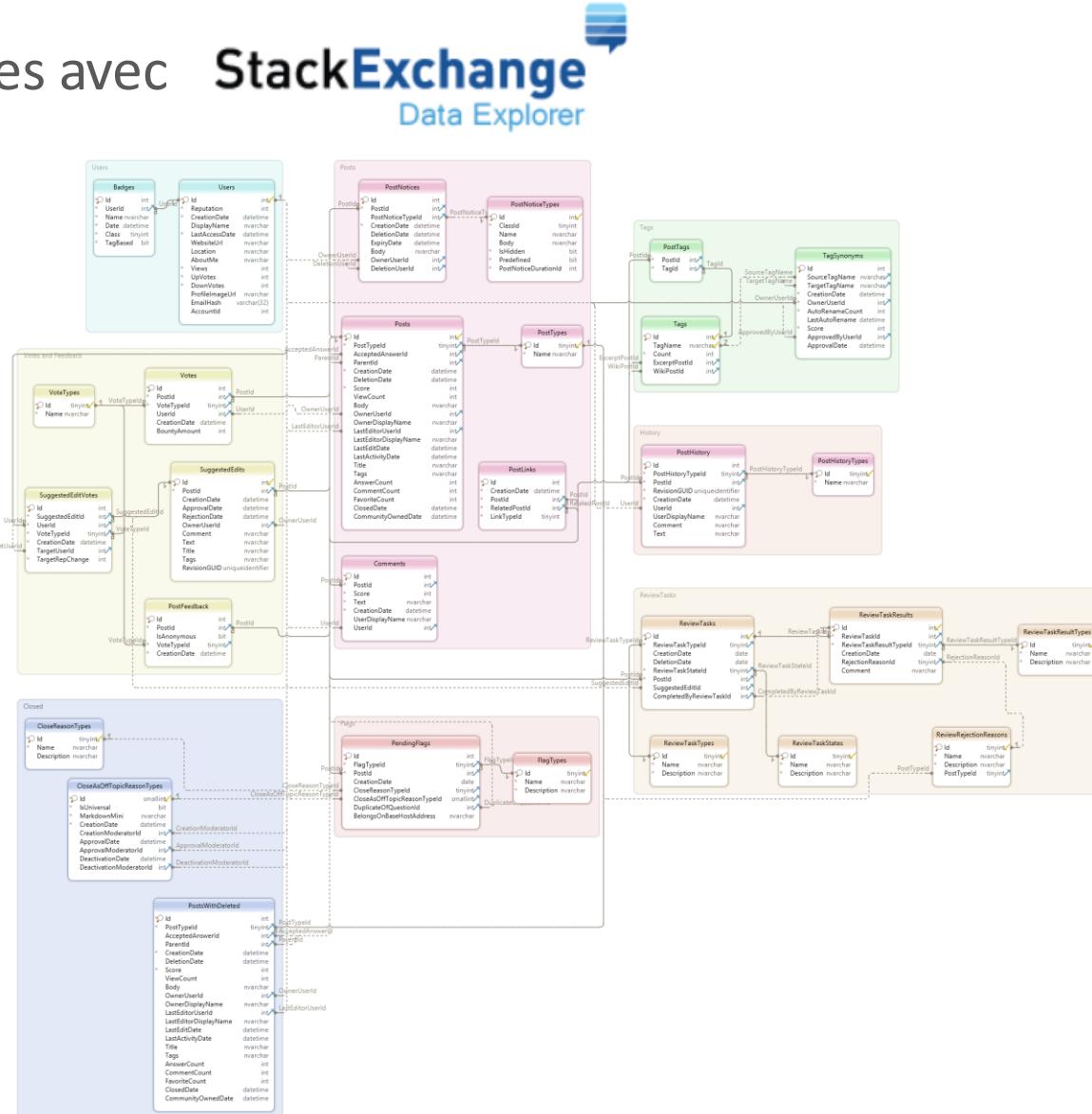
	Database Schema
Id	int
PostTypeId	tinyint
AcceptedAnswerId	int
ParentId	int
CreationDate	datetime
DeletionDate	datetime
Score	int
ViewCount	int

Below the 'Posts' schema, there's a section for 'Revisions' with the message 'Waiting for you to make your first edit...'. There are also 'A' and 'Z' sort buttons at the top of the schema table.

Pre-processing

2

Obtention des données avec StackExchange Data Explorer



Pre-processing

Obtention des données avec 

The screenshot shows the StackExchange Data Explorer interface, specifically the 'Posts' table schema. The table has 25 columns:

	Column Name	Type
1	Id	int
2	PostTypeId	tinyint
3	AcceptedAnswerId	int
4	ParentId	int
5	CreationDate	datetime
6	DeletionDate	datetime
7	Score	int
8	ViewCount	int
9	Body	nvarchar
10	OwnerUserId	int
11	OwnerDisplayName	nvarchar
12	LastEditorUserId	int
13	LastEditorDisplayName	nvarchar
14	LastEditDate	datetime
15	LastActivityDate	datetime
16	Title	nvarchar
17	Tags	nvarchar
18	AnswerCount	int
19	CommentCount	int
20	FavoriteCount	int
21	ClosedDate	datetime
22	CommunityOwnedDate	datetime

Pre-processing

2

Obtention des données avec 

 Home Queries Users Compose Query 

Viewing Query

Enter a title for your query  stackoverflow Q&A for professional and enthusiast programmers

edit description

```
1 SELECT Title,  
2     Body,  
3     Tags,  
4     Id,  
5     Score,  
6     ViewCount,  
7     AnswerCount  
8 FROM Posts  
9 WHERE (PostTypeId = 1 AND  
10    ViewCount > 10 AND  
11    Score > 5 AND  
12    AnswerCount > 0 AND  
13    LEN(Tags) - LEN(REPLACE(Tags, '<', '')) >= 5)|
```

Database Schema	
? A + -	
Posts	
Id	int
PostTypeId	tinyint
AcceptedAnswerId	int
ParentId	int
CreationDate	datetime
DeletionDate	datetime
Score	int
ViewCount	int
Revisions	
Waiting for you to make your first edit...	

Pre-processing

2

Aperçu des données

	Title	Body	Tags	Id	Score	ViewCount	AnswerCount
0	Multithreading in a stateless session bean?	<p>The EJB 3.0 specification does not allow a ...	<java><multithreading><jakarta-ee><ejb-3.0><ejb>	3816286	9	13503	4
1	base64 JSON encoded strings in nodejs	<p>How do I create a base64 JSON encoded strin...	<javascript><json><node.js><base64><buffer>	22515180	14	23710	3
2	How do you get a directory listing in C?	<p>How do you scan a directory for folders and...	<c><file><directory><cross-platform><common-ta...	12489	69	142856	9

HTML

créer liste

Concaténer

Taille

Nombre de lignes	50 000
Nombre de colonnes	7

Valeurs manquantes et dtypes

#	Column	Non-Null Count	Dtype
0	Title	50000 non-null	object
1	Body	50000 non-null	object
2	Tags	50000 non-null	object
3	Id	50000 non-null	int64
4	Score	50000 non-null	int64
5	ViewCount	50000 non-null	int64
6	AnswerCount	50000 non-null	int64

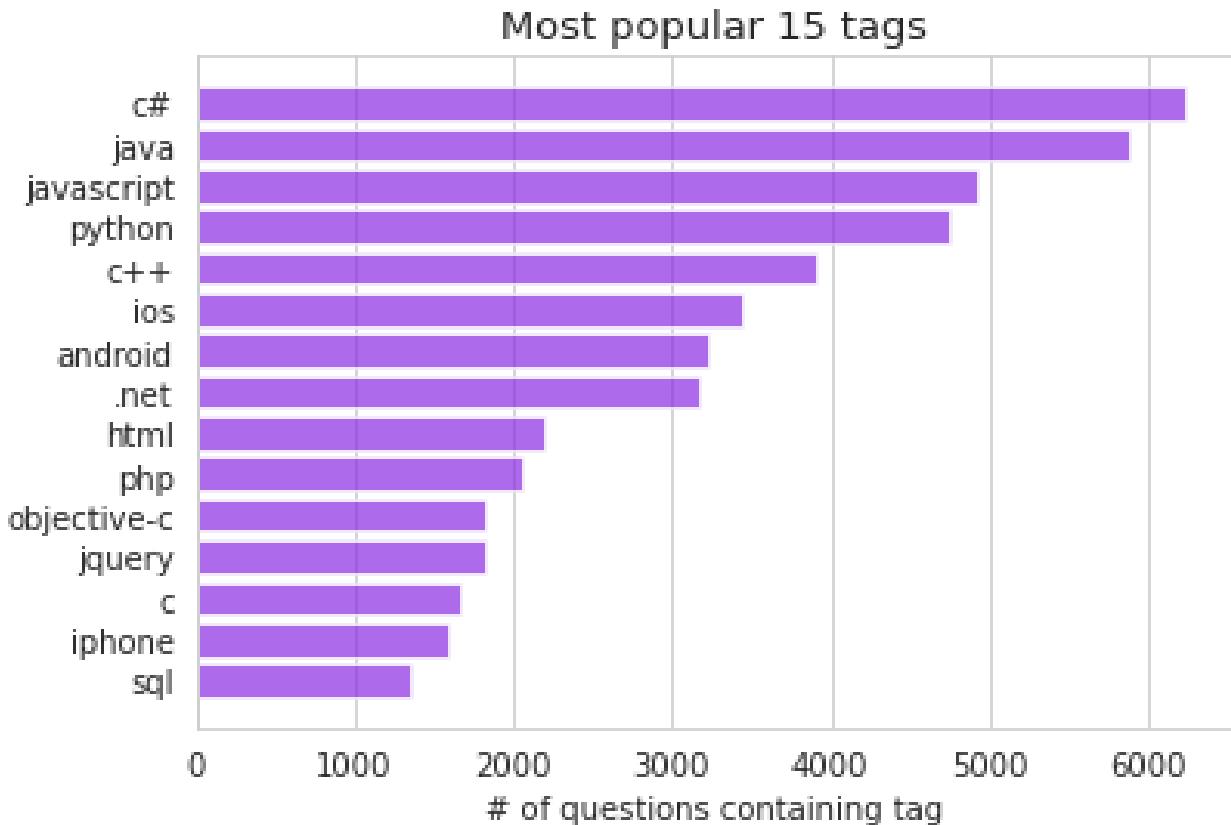
Pre-processing

2

Filtrer ensemble de tags: + garder docs. contenant au moins 1 de ces tags

The total # of popular tags is 38
(appearing in more than 700 questions)

Tag	# of quest
c#	6226.0
java	5880.0
javascript	4920.0
python	4747.0
c++	3904.0
ios	3449.0
android	3223.0
.net	3164.0
html	2197.0
php	2050.0
objective-c	1816.0
jquery	1810.0
c	1655.0
iphone	1592.0
sql	1349.0
asp.net	1339.0
css	1331.0
linux	1286.0
node.js	1253.0
performance	1076.0



Pre-processing

2

Création de target avec ces 38 tags:

The total # of popular tags is 38 (appearing in more than 700 questions)	
<hr/>	
Tag	# of quest
c#	6226.0
java	5880.0
javascript	4920.0
python	4747.0
c++	3904.0
ios	3449.0
android	3223.0
.net	3164.0
html	2197.0
php	2050.0
objective-c	1816.0
jquery	1810.0
c	1655.0
iphone	1592.0
sql	1349.0
asp.net	1339.0
css	1331.0
linux	1286.0
node.js	1253.0
performance	1076.0

Creating target

```
1 mlb = MultiLabelBinarizer(classes=popular_tags)
2 mlb.fit([popular_tags])
```

MultiLabelBinarizer

```
MultiLabelBinarizer(classes=['c#', 'java', 'javascript', 'python', 'c++', 'ios',
'android', '.net', 'html', 'php', 'objective-c',
'jquery', 'c', 'iphone', 'sql', 'asp.net', 'css',
'linux', 'node.js', 'performance', 'spring',
'windows', 'swift', 'xcode', 'ruby-on-rails',
'mysql', 'json', 'sql-server', 'multithreading',
'asp.net-mvc', ...])
```

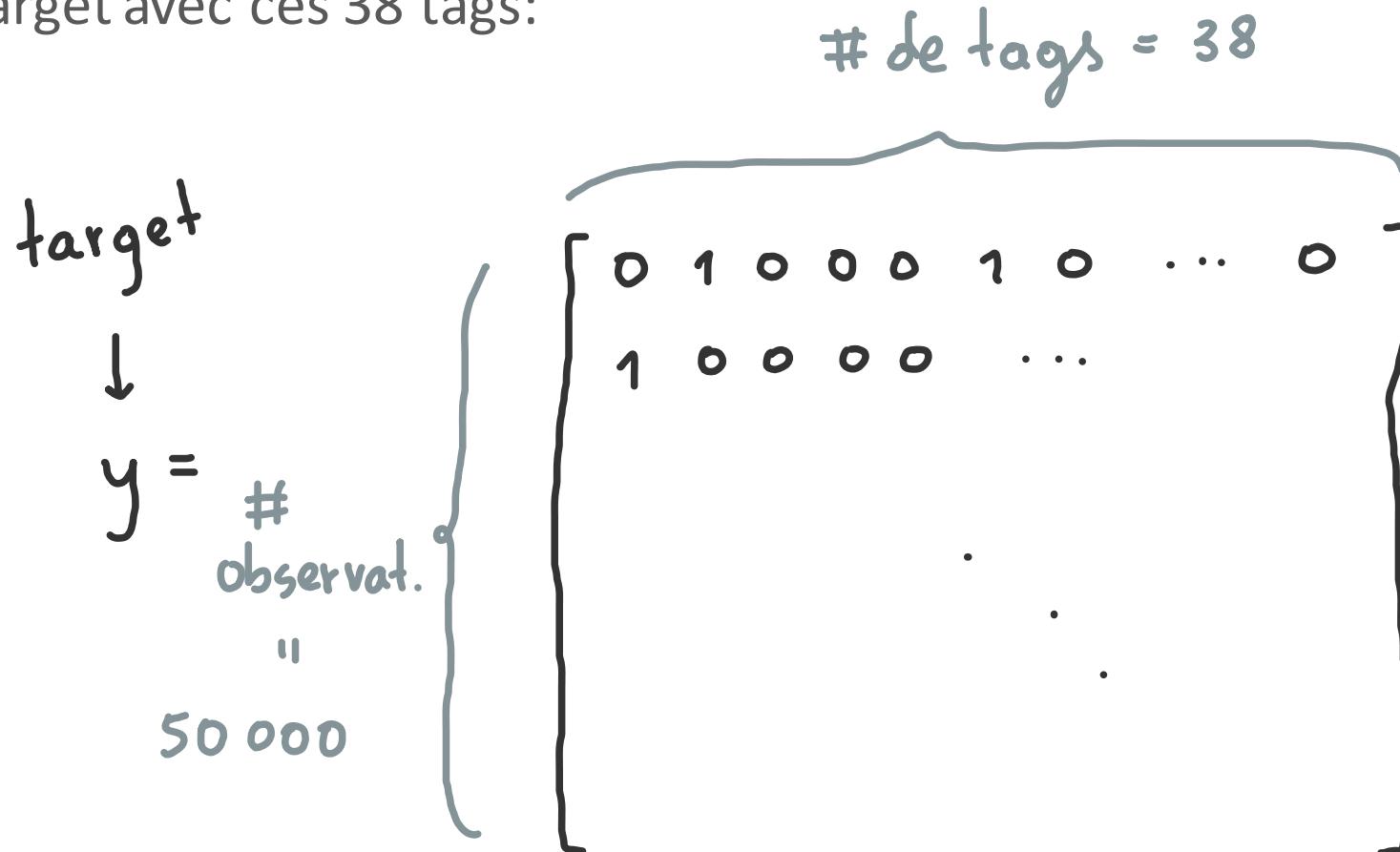
tags codifiées → [0, 0, 1, 0, 0, 0, 1, 0, ..., 0] taille : 38

linux → python

Pre-processing

2

Création de target avec ces 38 tags:



Pre-processing

Création de fonction de nettoyage/tokenisation de texte

```
1 def clean(text, tokenize=False, strict=False, **kwargs):
2     """
3         Returns a dictionary with keys 'text' or 'tokens', where
4         'tokens' corresponds to the list of lemmatized tokens from
5         the string text. Omitting stopwords and punctuation, and the text is
6         the joint text.
7
8     Parameters:
9         - text: str
10            If True returns list of tokens, if False returns string.
11        - strict: bool
12            If true only keeps nouns
13
14    """
15
16
17    # Removing <code>some code</code>
18    clean_txt = remove_code(text)
19
20    # Removing HTML tags
21    soup = BeautifulSoup(clean_txt, features='html.parser')
22    clean_txt = soup.get_text()
23
24    # Removing new line character: \n
25    clean_txt = clean_txt.replace('\n', ' ')
26
27    # Removing unicode characters
28    clean_txt = clean_txt.encode("ascii", "ignore").decode()
29
30    # Removing digits
31    clean_txt = ''.join(char for char in clean_txt if not char.isdigit())
32
```

①

Enlever balises
HTML , new line \h,
emojis

②

chiffres

Pre-processing

Création de fonction de nettoyage/tokenisation de texte

```
32 # Replacing 'c ++' and 'c #' for 'c++' and 'c#' and others
33 clean_txt = clean_txt.replace('c ++', 'c++')
34 clean_txt = clean_txt.replace('c #', 'c#')
35 clean_txt = clean_txt.replace('C ++', 'c++')
36 clean_txt = clean_txt.replace('C #', 'c#')
37 clean_txt = clean_txt.replace('C#", "c#')
38 clean_txt = clean_txt.replace('C ++', 'c++')
39
40 # Adding special case rule
41 special_case = [{ORTH: "c#"}]
42 nlp.tokenizer.add_special_case("c#", special_case)
43 special_case = [{ORTH: ".net"}]
44 nlp.tokenizer.add_special_case(".net", special_case)
45 special_case = [{ORTH: "objective-c"}]
46 nlp.tokenizer.add_special_case("objective-c", special_case)
47 special_case = [{ORTH: "asp.net"}]
48 nlp.tokenizer.add_special_case("asp.net", special_case)
49 special_case = [{ORTH: "node.js"}]
50 nlp.tokenizer.add_special_case("node.js", special_case)
51 special_case = [{ORTH: "ruby-on-rails"}]
52 nlp.tokenizer.add_special_case("ruby-on-rails", special_case)
53 special_case = [{ORTH: "sql-server"}]
54 nlp.tokenizer.add_special_case("sql-server", special_case)
55 special_case = [{ORTH: "unit-testing"}]
56 nlp.tokenizer.add_special_case("unit-testing", special_case)
```

③ Attention aux caractères propres aux noms des langages

C ++ → C++

Pre-processing

Création de fonction de nettoyage/tokenisation de texte

```

58
59     # Tokenize with spacy
60     doc = nlp(clean_txt)
61
62     # Tokenize properties
63     if strict == True:
64         tokens = [token.lemma_.lower() for token in doc
65                     if token.pos_ in ['NOUN', 'PROPN', 'VERB'] and
66                     (not (token.is_stop or
67                           token.is_punct or
68                           token.is_space
69                           )
70                     )
71             ]
72     else:
73         tokens = [token.lemma_.lower() for token in doc
74                     if not (token.is_stop or
75                           token.is_punct or
76                           token.is_space
77                           )
78             ]
79
80     clean_txt = ' '.join(tokens)
81
82     # Ask if return text or tokens
83     if tokenize == True:
84         result = tokens
85     else:
86         result = clean_txt
87
88     # Option for list of entities in output
89     if 'ent' in kwargs:
90         result = {'output':result, 'ents': doc.ents}
91
92     return result

```

4

Tokenisation avec spaCy

5

Filtrer : noms
noms propres
verbes

6

Résultat

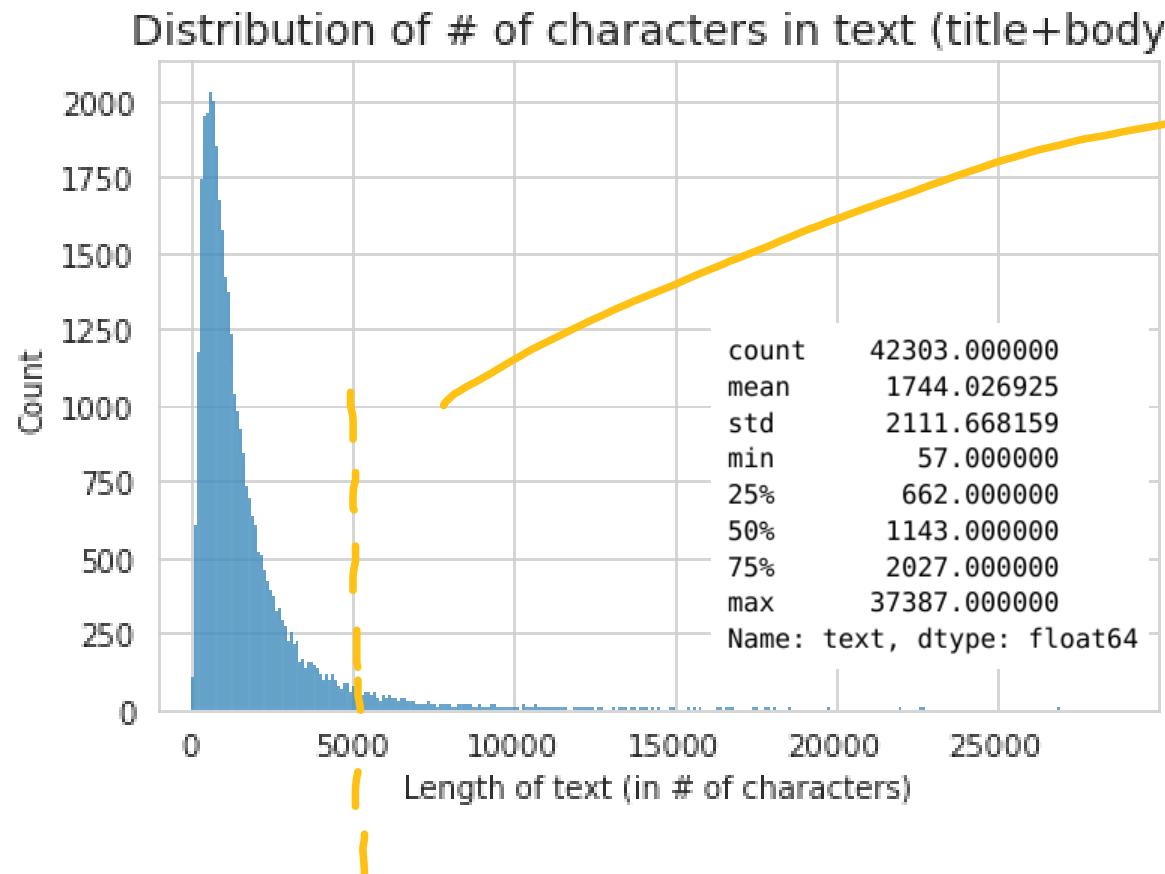
texte
ou
tokens

Voir example dans notebook

Pre-processing

2

Derniers détails préparation texte



① Concaténer

② filtrer (≤ 5000) car.

Shape of dataset before filter: (42303, 45)
Shape of dataset after filter: (40166, 45)
Number of deleted rows: 2137

③ Appliquer ma
fonction nettoyage

Total time to process text: 1233.7 seconds. (= 20.56 minutes)

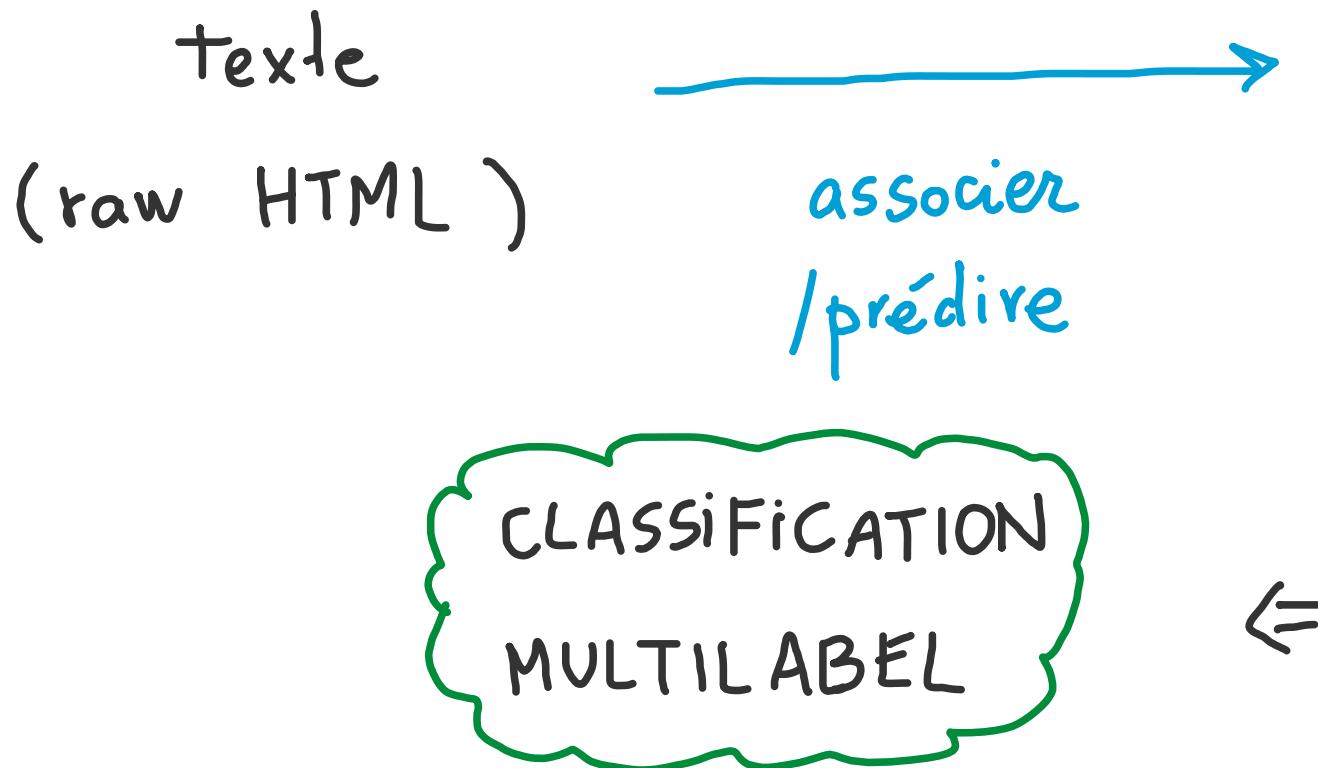


EXPORTER DONNÉES !

3

Modélisation +
choix de modèle

Notre problème :

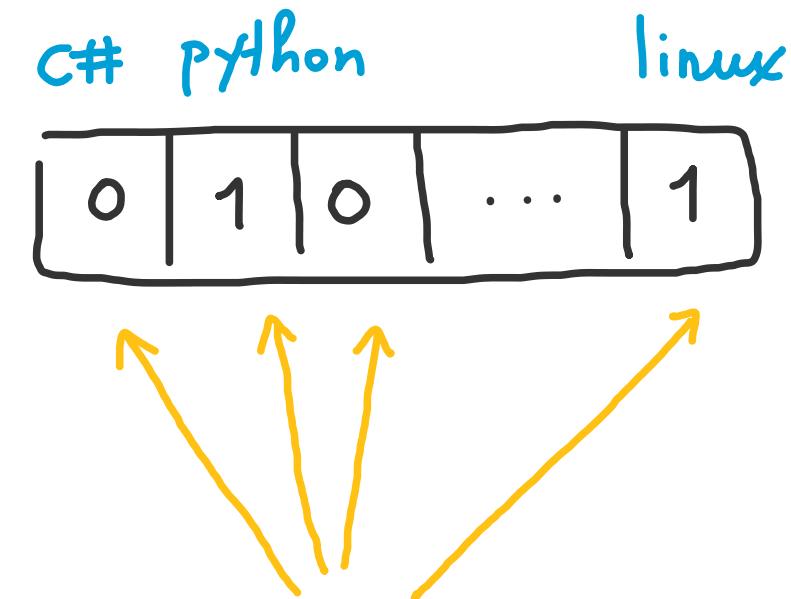
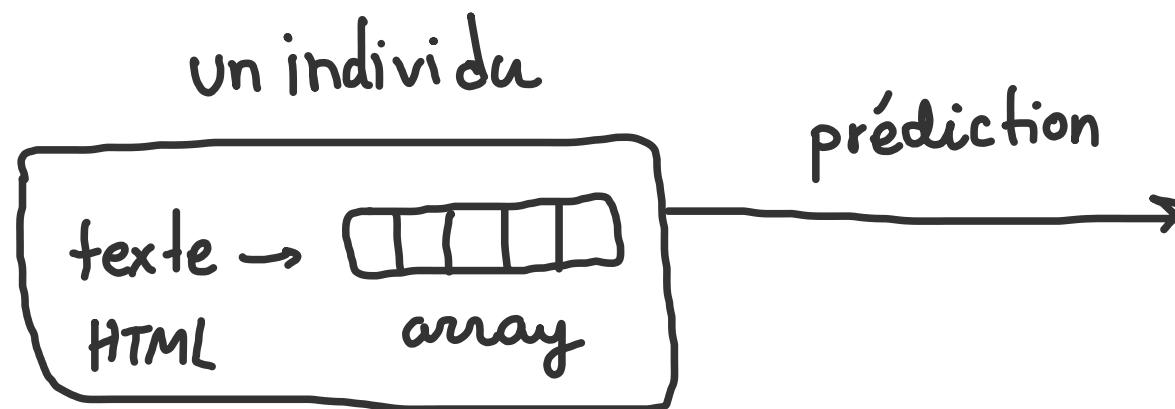


étiquettes : tags

c# python linux
sql pandas

↓
plusieurs sont possibles
pour 1 seul input

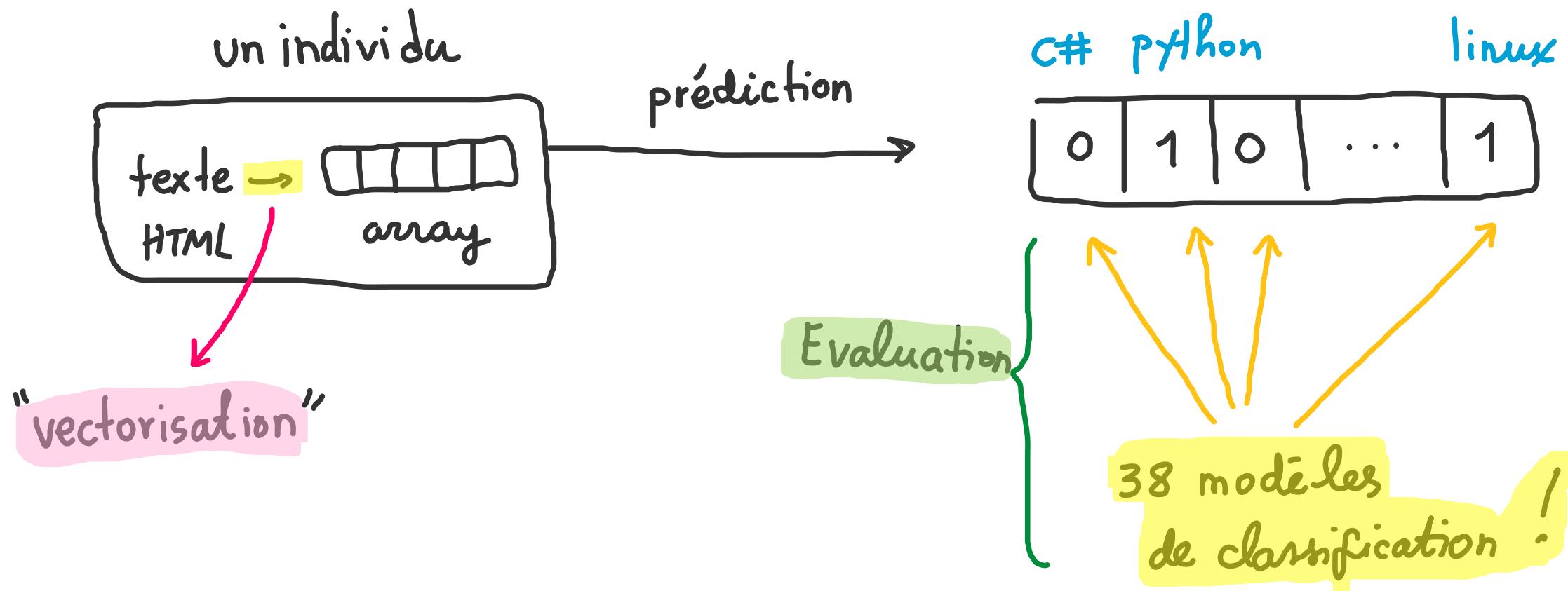
Notre stratégie :  OneVsRestClassifier



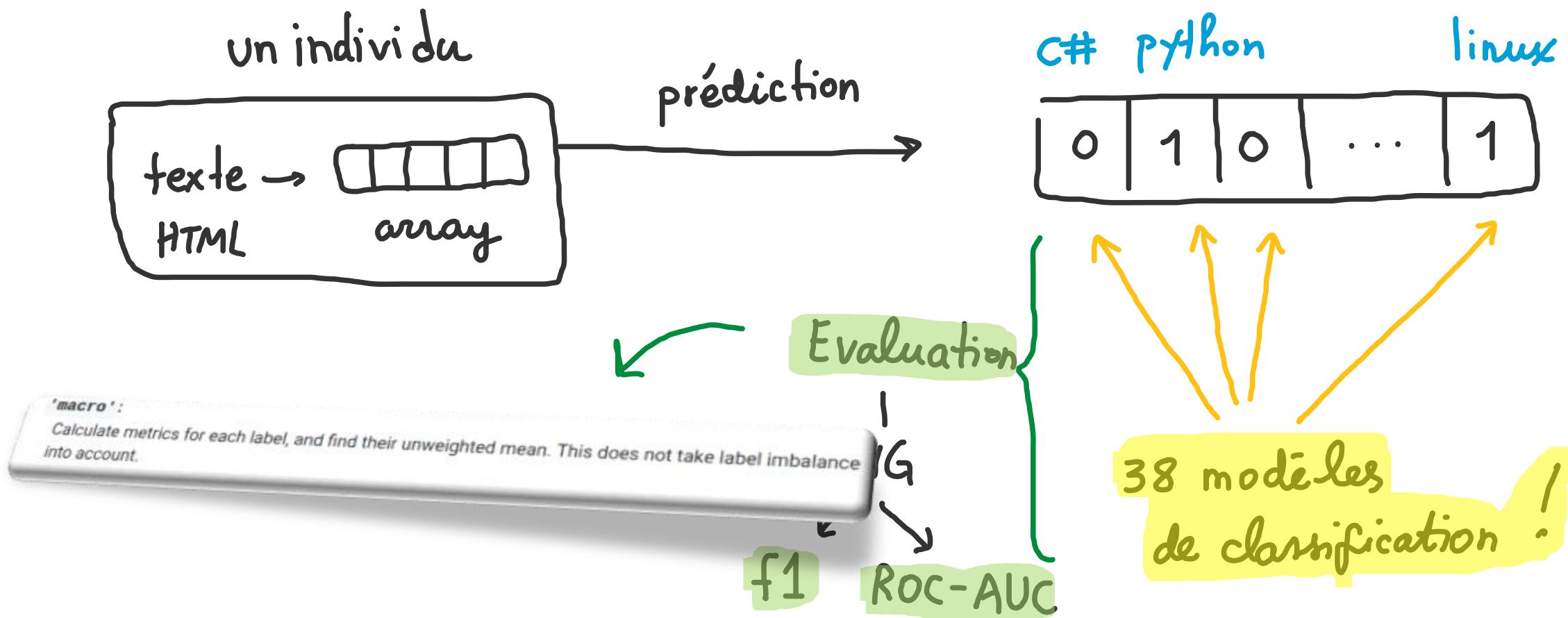
CLASSIFICATION
MULTILABEL

38 modèles
de classification !

Notre stratégie :  OneVsRestClassifier



Notre stratégie :  OneVsRestClassifier



Modélisation

3

→ texte →  array

VECTORISER

CLASSIFIER

non supervisé

Bag of words : TF - IDF

LDA

Word 2 Vec

supervisé

BERT

Logistic Regression

USE

Random Forest Classifier

Multinomial Naive Bayes

Modélisation

3

→ texte →  array

VECTORISER

CLASSIFIER

non supervisé

Bag of words : TF - IDF

LDA

Word 2 Vec

supervisé

BERT

Logistic Regression

USE

Random Forest Classifier

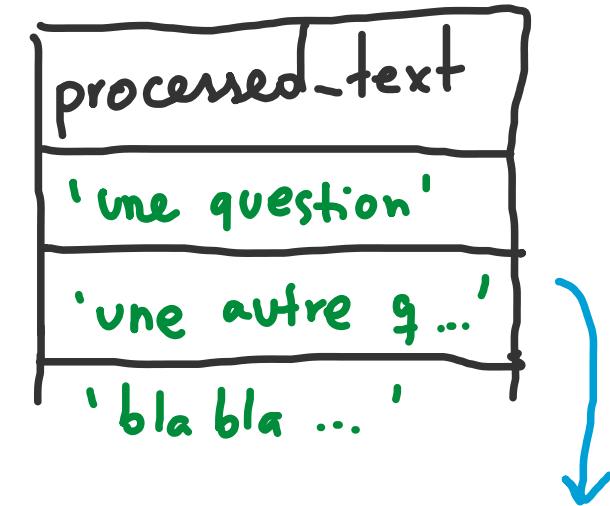
Multinomial Naive Bayes

Modélisation (supervisée)

3

Bag of words: TF-IDF

```
● ● ●  
1 # Instantiate the vectorizer  
2 tfidf = TfidfVectorizer(max_df=0.75,  
3                         min_df=0.0015  
4                         )  
5  
6 # Vectorize the processed text  
7 X = tfidf.fit_transform(df['processed_text'])  
8  
9 # Defining target vector  
10 columns_tags = ['tag_'+tag for tag in tag_list]  
11 y = df[columns_tags]  
12  
13 # Splitting into train and test set  
14 X_train, X_test, y_train, y_test = train_test_split(  
15     X,y, test_size=0.33, random_state=5  
16 )
```



matrice de features

10 000

	mot1	mot2	...	mot k
0	1	3	0	... 4
1	0	2	0	... 0
3	4	0	0	.. 0
			:	

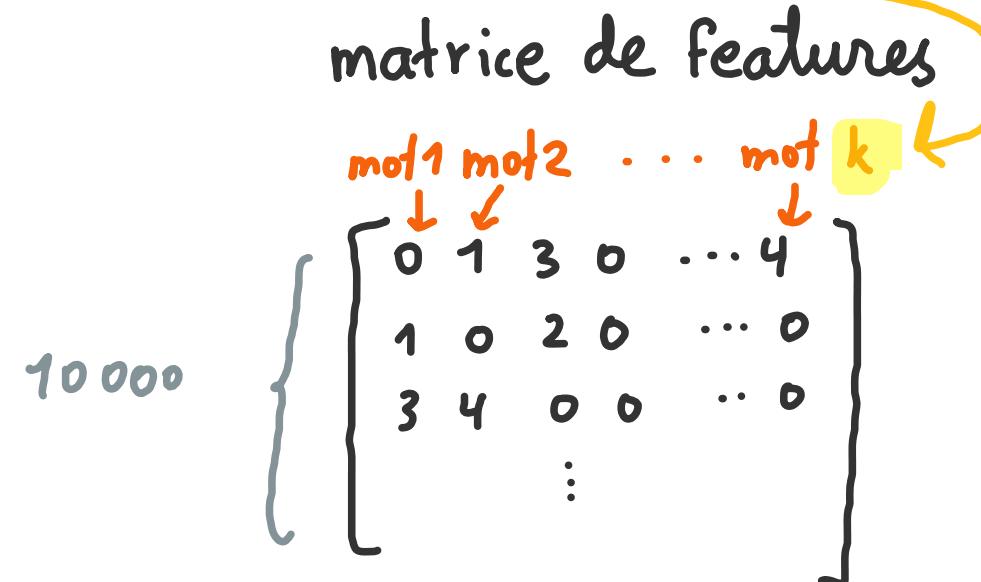
Modélisation (supervisée)

3

Bag of words: TF-IDF

```
● ● ●  
1 # Instantiate the vectorizer  
2 tfidf = TfidfVectorizer(max_df=0.75,  
3                         min_df=0.0015  
4                         )  
5  
6 # Vectorize the processed text  
7 X = tfidf.fit_transform(df.processed_text)  
8  
9 # Defining target vector  
10 columns_tags = ['tag_'+tag for tag in tag_list]  
11 y = df[columns_tags]  
12  
13 # Splitting into train and test set  
14 X_train, X_test, y_train, y_test = train_test_split(  
15             X,y, test_size=0.33, random_state=5  
16 )
```

processed_text
'une question'
'une autre q ...'
'bla bla ...'

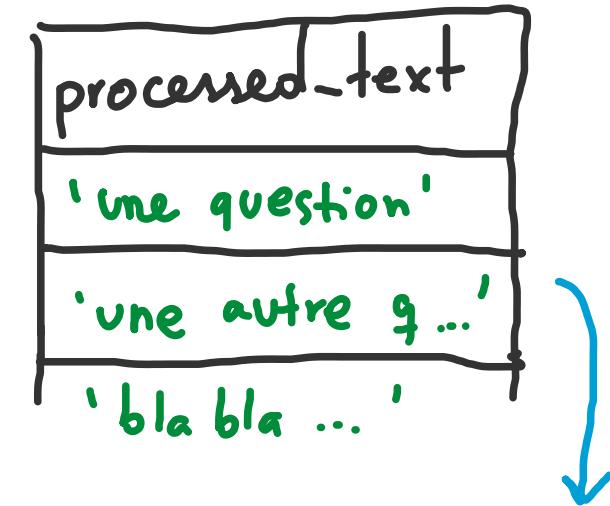


Modélisation (supervisée)

3

Bag of words: TF-IDF

```
● ● ●  
1 # Instantiate the vectorizer  
2 tfidf = TfidfVectorizer(max_df=0.75,  
3                         min_df=0.0015  
4                         )  
5  
6 # Vectorize the processed text  
7 X = tfidf.fit_transform(df['processed_text'])  
8  
9 # Defining target vector  
10 columns_tags = ['tag_'+tag for tag in tag_list]  
11 y = df[columns_tags]  
12  
13 # Splitting into train and test set  
14 X_train, X_test, y_train, y_test = train_test_split(  
15     X,y, test_size=0.33, random_state=5  
16 )
```



matrice de features

A hand-drawn diagram of a sparse matrix representing the Bag of Words features. The columns are labeled "mot1", "mot2", "...", "mot", and "2014". The matrix has 10,000 rows, indicated by a bracket on the left. The first few rows are shown:

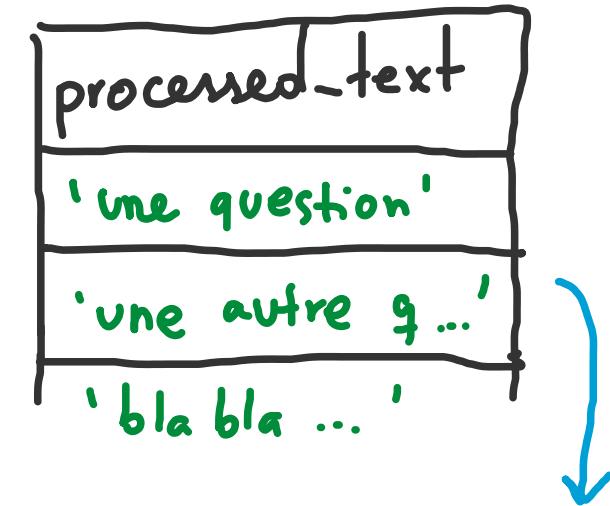
	mot1	mot2	...	mot	2014
0	0	1	3	0	... 4
1	1	0	2	0	... 0
3	3	4	0	0	.. 0
				:	

Modélisation (supervisée)

3

Bag of words: **TF-IDF** → term frequency

```
● ● ●  
1 # Instantiate the vectorizer  
2 tfidf = TfidfVectorizer(max_df=0.75,  
3                         min_df=0.0015  
4                         )  
5  
6 # Vectorize the processed text  
7 X = tfidf.fit_transform(df['processed_text'])  
8  
9 # Defining target vector  
10 columns_tags = ['tag_'+tag for tag in tag_list]  
11 y = df[columns_tags]  
12  
13 # Splitting into train and test set  
14 X_train, X_test, y_train, y_test = train_test_split(  
15     X,y, test_size=0.33, random_state=5  
16 )
```



matrice de features

A hand-drawn diagram of a sparse matrix representing the Bag of Words model. The columns are labeled "mot1", "mot2", "...", "mot 2014". The matrix has 10,000 rows, indicated by a bracket on the left. Arrows point from the column labels to the first few columns of the matrix. The matrix entries are mostly zero, with non-zero values in the first few columns of each row.

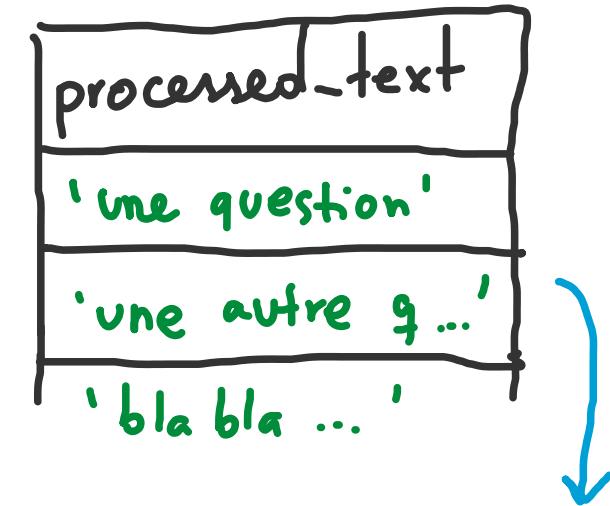
	mot1	mot2	...	mot 2014		
1	0	1	3	0	...	4
2	1	0	2	0	...	0
3	3	4	0	0	...	0
...						

Modélisation (supervisée)

3

Bag of words: **TF-IDF** → term frequency

```
● ● ●  
1 # Instantiate the vectorizer  
2 tfidf = TfidfVectorizer(max_df=0.75,  
3                         min_df=0.0015  
4                         )  
5  
6 # Vectorize the processed text  
7 X = tfidf.fit_transform(df.processed_text)  
8  
9 # Defining target vector  
10 columns_tags = ['tag_'+tag for tag in tag_list]  
11 y = df[columns_tags]  
12  
13 # Splitting into train and test set  
14 X_train, X_test, y_train, y_test = train_test_split(  
15     X,y, test_size=0.33, random_state=5  
16 )
```



matrice de features

A hand-drawn diagram of a sparse matrix representing the Bag of Words model. The columns are labeled "mot1", "mot2", "...", "mot 2014". The matrix has 10,000 rows, indicated by a bracket on the left. Arrows point from the column labels to the first four columns of the matrix. The matrix entries are mostly zero, with non-zero values in the first few columns of each row.

	mot1	mot2	...	mot 2014		
1	0	1	3	0	...	4
2	1	0	2	0	...	0
3	3	4	0	0	...	0
...						

Modélisation (supervisée)

Bag of words: **TF-IDF**

```

1 # Instantiate the vectorizer
2 tfidf = TfidfVectorizer(max_df=0.75,
3                         min_df=0.0015
4                         )
5
6 # Vectorize the processed text
7 X = tfidf.fit_transform(df.processed_text)
8
9 # Defining target vector
10 columns_tags = ['tag_'+tag for tag in tag_list]
11 y = df[columns_tags]
12
13 # Splitting into train and test set
14 X_train, X_test, y_train, y_test = train_test_split(
15     X,y, test_size=0.33, random_state=5
16 )

```

term frequency
 \times Inverse doc.
 frequency

processed_text
'une question'
'une autre q ...'
'bla bla ...'

matrice de features

10 000

$$\left[\begin{array}{cccc} \text{mot 1} & \text{mot 2} & \dots & \text{mot k} \\ 0 & 0.7 & 0 & \dots 0.1 \\ 0.8 & 0 & 0.17 & \dots 0 \\ 0.42 & 0.5 & 0 & \dots 0 \\ \vdots & & & \end{array} \right]$$

Modélisation (supervisée)

Bag of words: TF-IDF + LogisticRegression

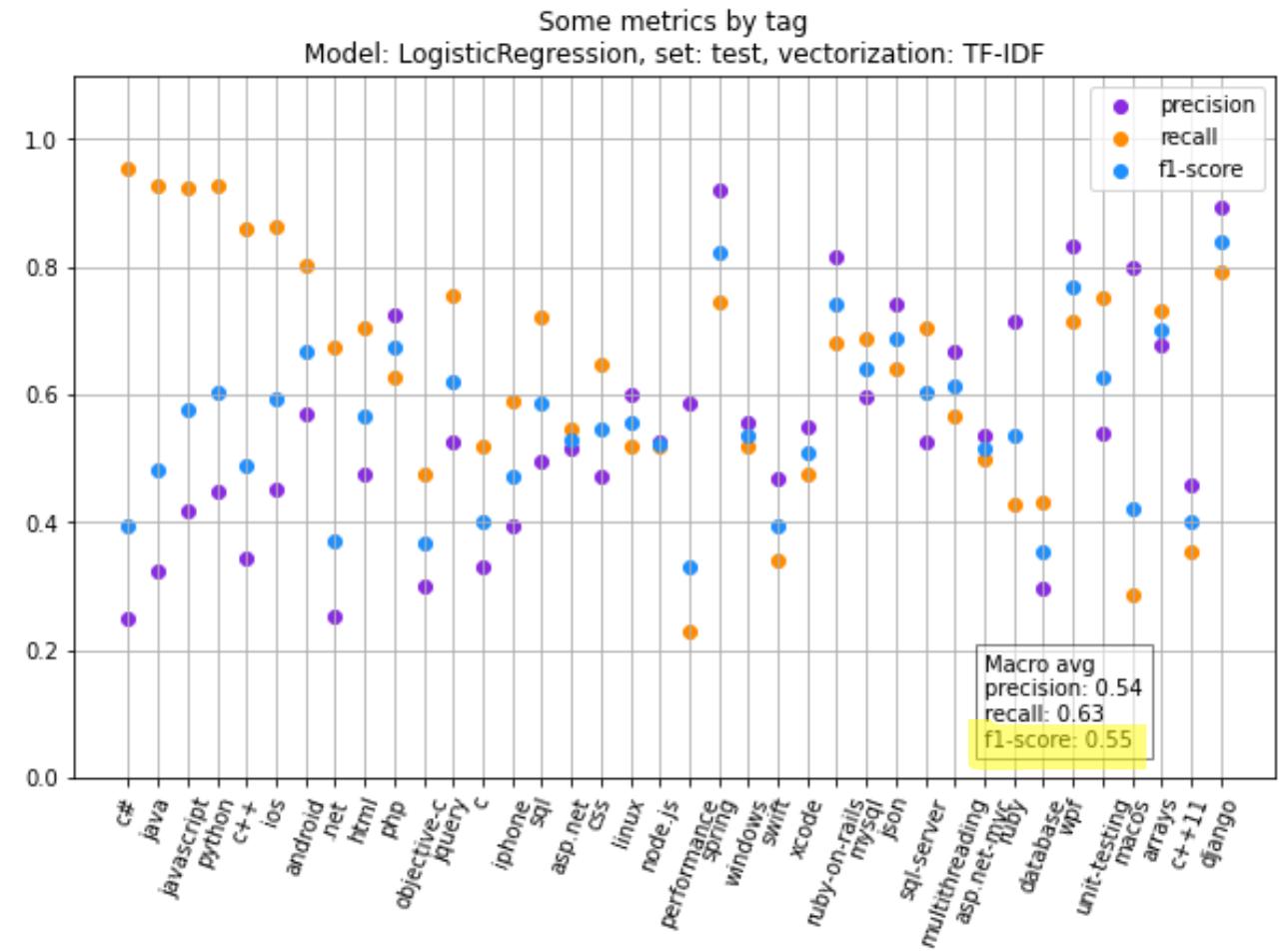
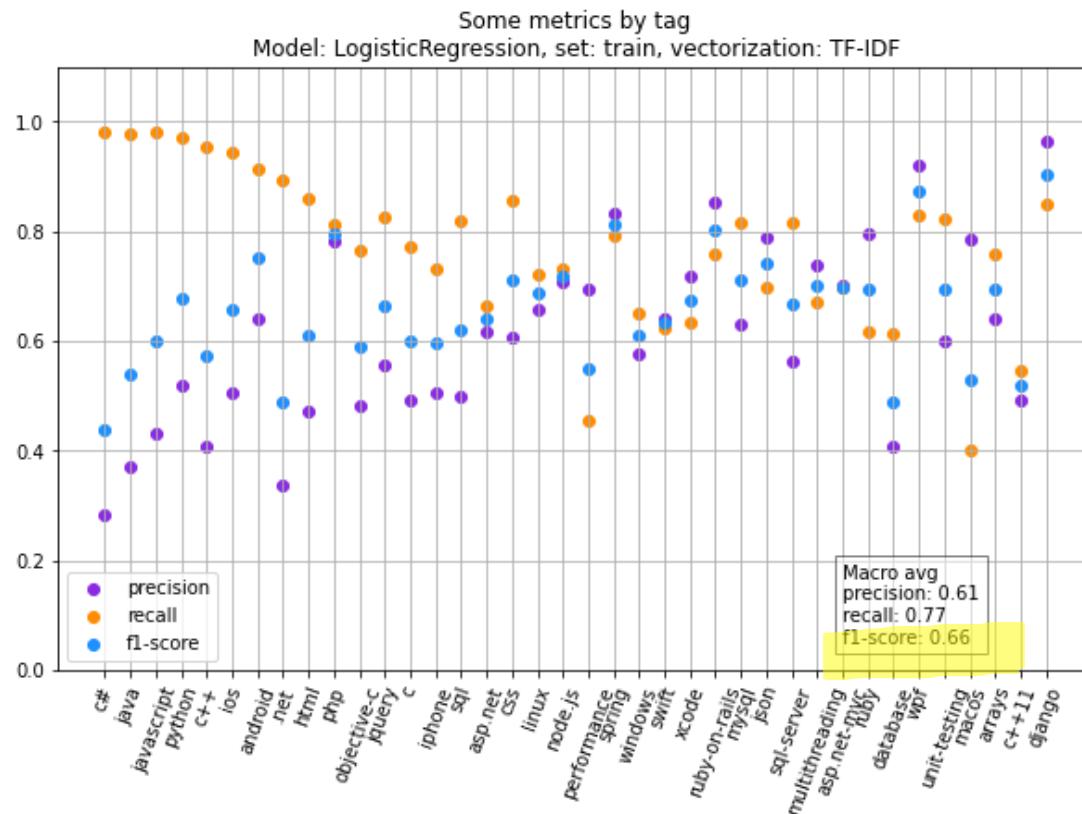
```
● ● ●  
1 # Hyperparameters for logistic regression  
2 parameters = {  
3     'estimator__penalty' : ['l1', 'l2'],  
4     'estimator__tol' : [1e-6, 1e-5, 1e-4, 1e-3],  
5     'estimator__C' : [0.01, 0.05, 0.1, 0.5, 0.7, 1],  
6     'estimator__fit_intercept' : [True, False],  
7     'estimator__solver' : ['liblinear', 'sag', 'saga']  
8 }  
9  
10 # Instantiating OneVsRest Classifier  
11 cl = OneVsRestClassifier(LogisticRegression())  
12  
13 # Random search for best hyperparameters  
14 random_search = RandomizedSearchCV(  
15     estimator = cl,  
16     param_distributions= parameters,  
17     n_iter=40,  
18     scoring='roc_auc_ovr',  
19     random_state=5,  
20     n_jobs=-1  
21 )  
22  
23 # Perform random search  
24 random_search.fit(X_train, y_train)  
25  
26 # Instantiating BEST OneVsRest Classifier  
27 cl = random_search.best_estimator_  
28  
29 # Fitting classifier  
30 start = time.time()  
31 cl.fit(X_train,y_train)  
32 finish = time.time()  
33 fittime = finish - start  
34 print(f'fit time: {fittime} secs.')  
35
```

validation croisée avec
Randomized Search

évaluation du temps aussi

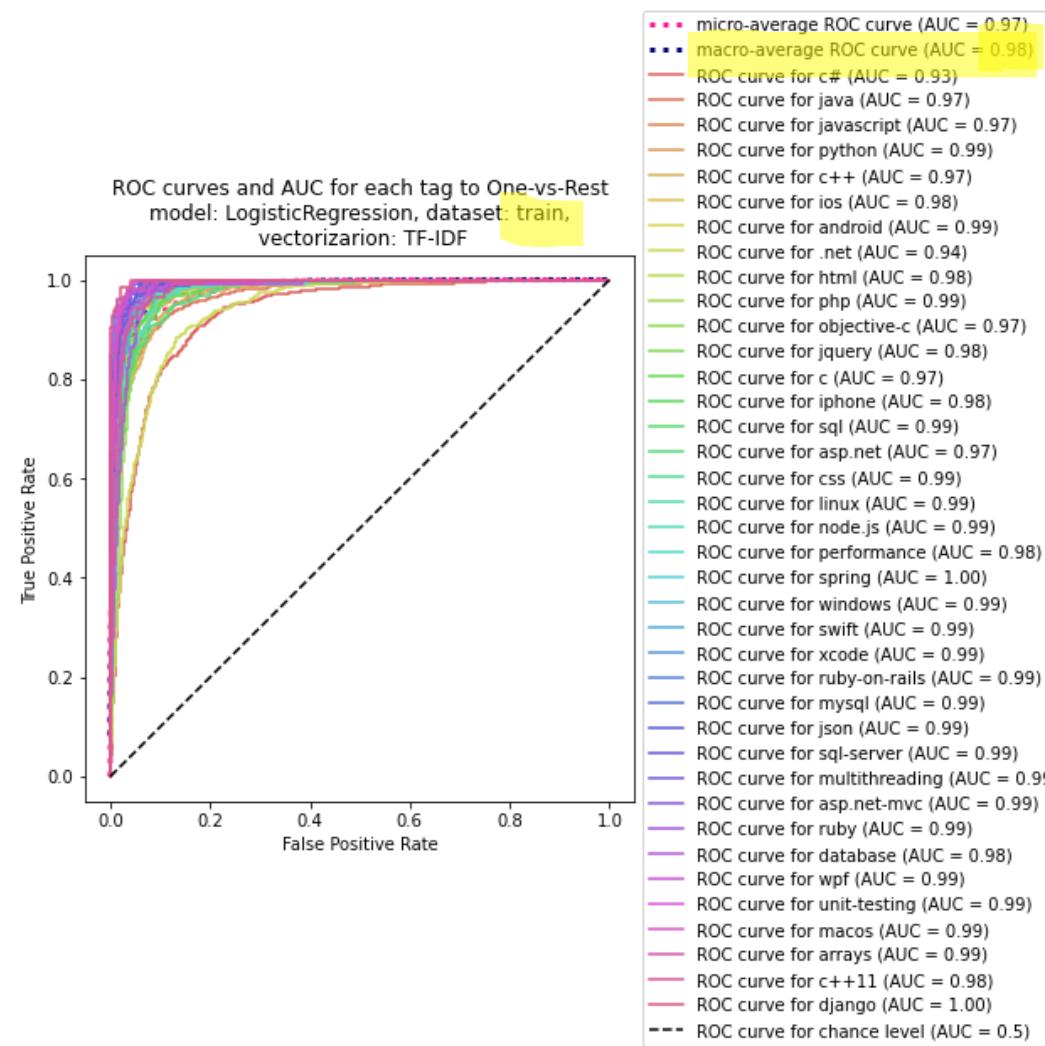
Modélisation (supervisée)

Bag of words: TF-IDF + LogisticRegression --> Évaluation



Modélisation (supervisée)

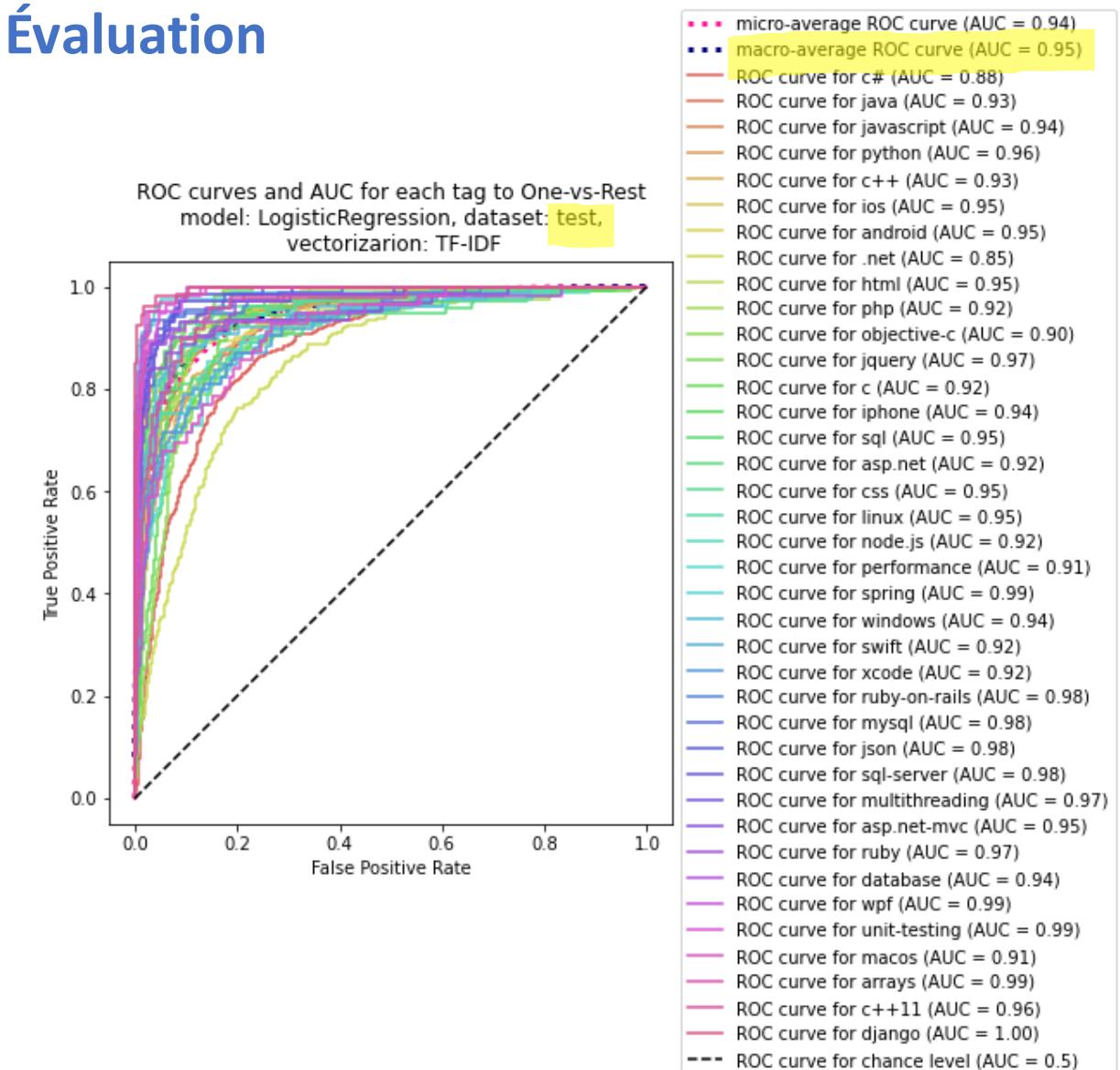
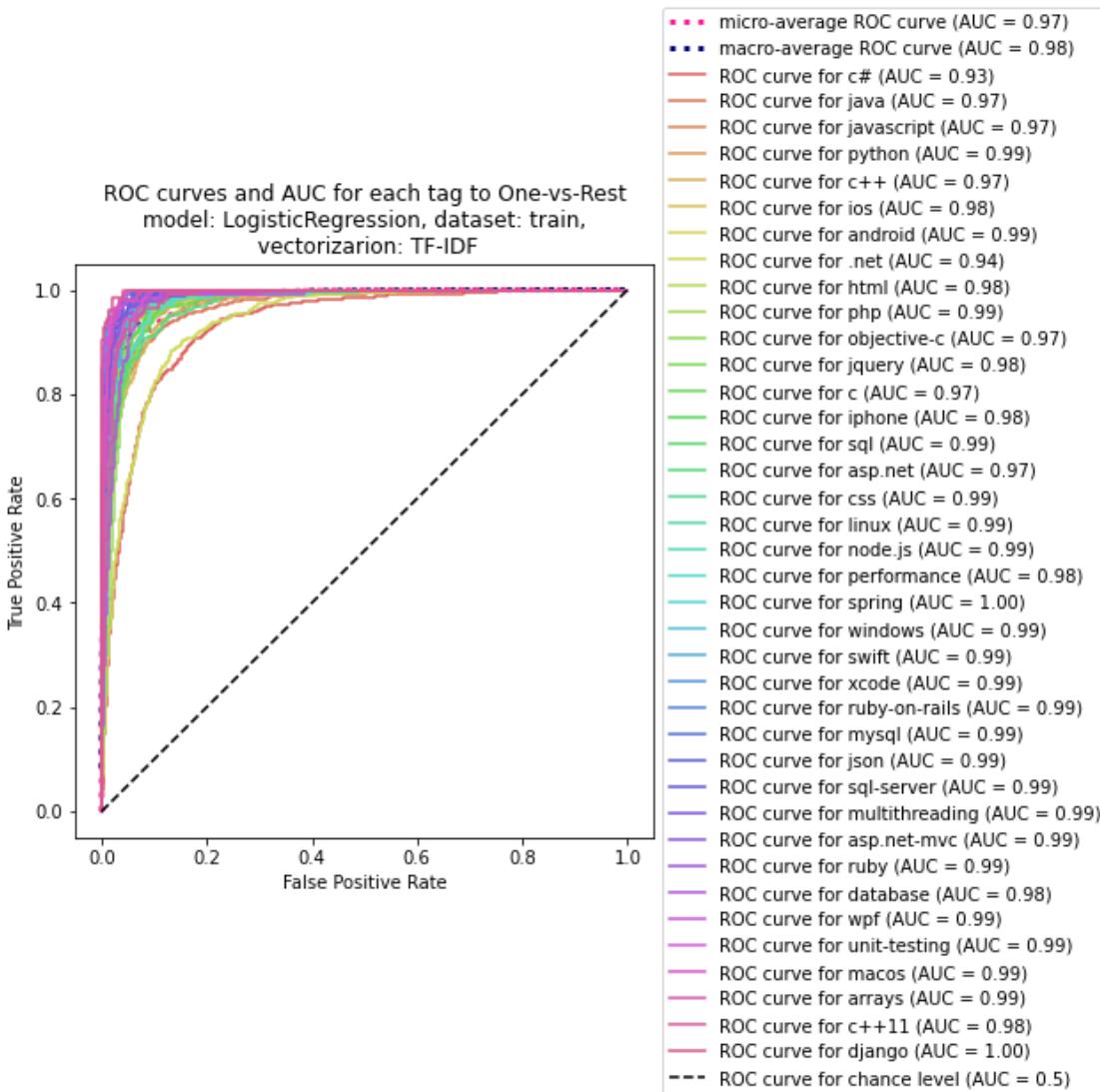
Bag of words: TF-IDF + LogisticRegression --> Évaluation



ROC pour chacun des classifiants

Modélisation (supervisée)

Bag of words: TF-IDF + LogisticRegression --> Évaluation



Modélisation (supervisée)

3

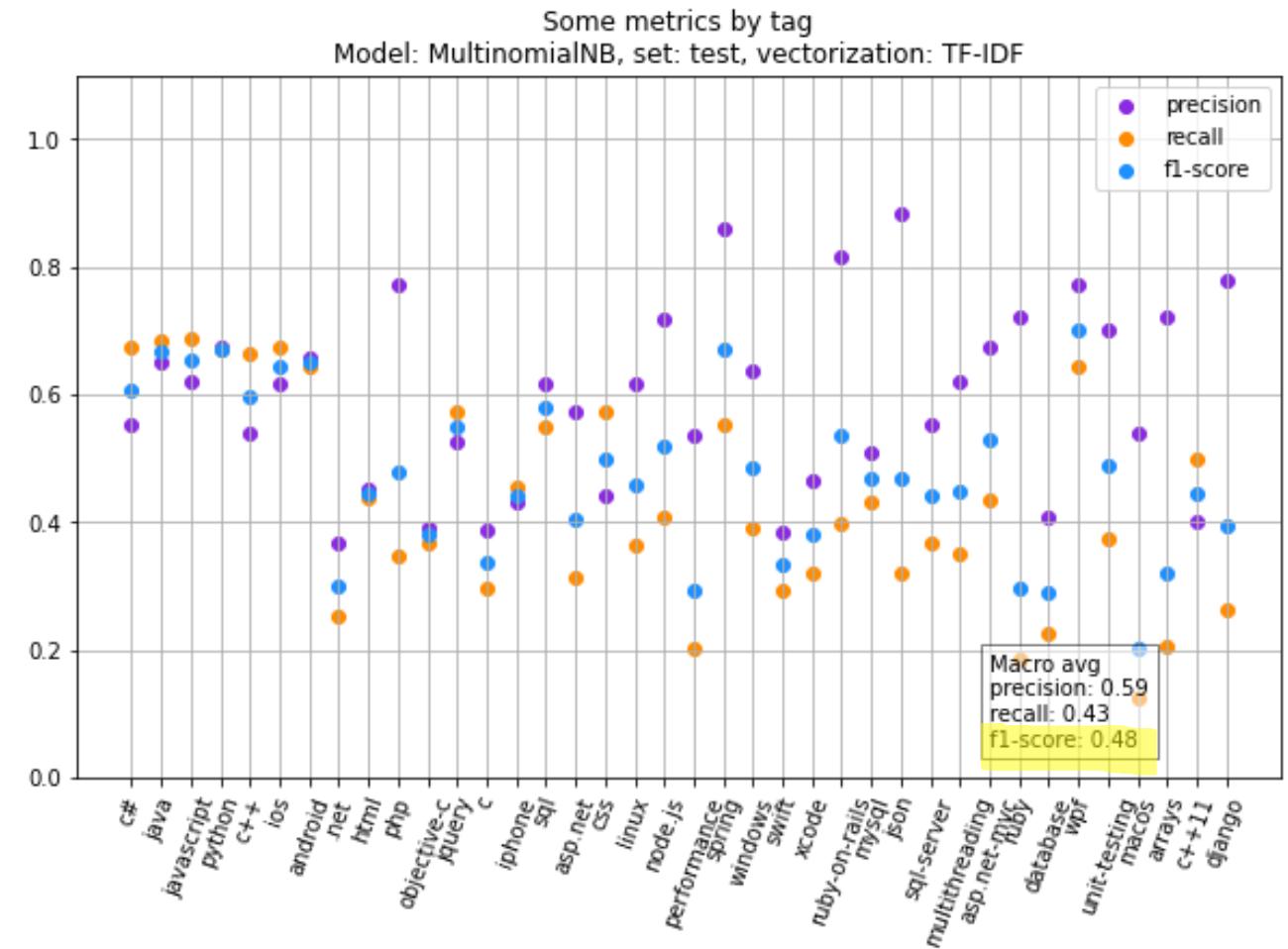
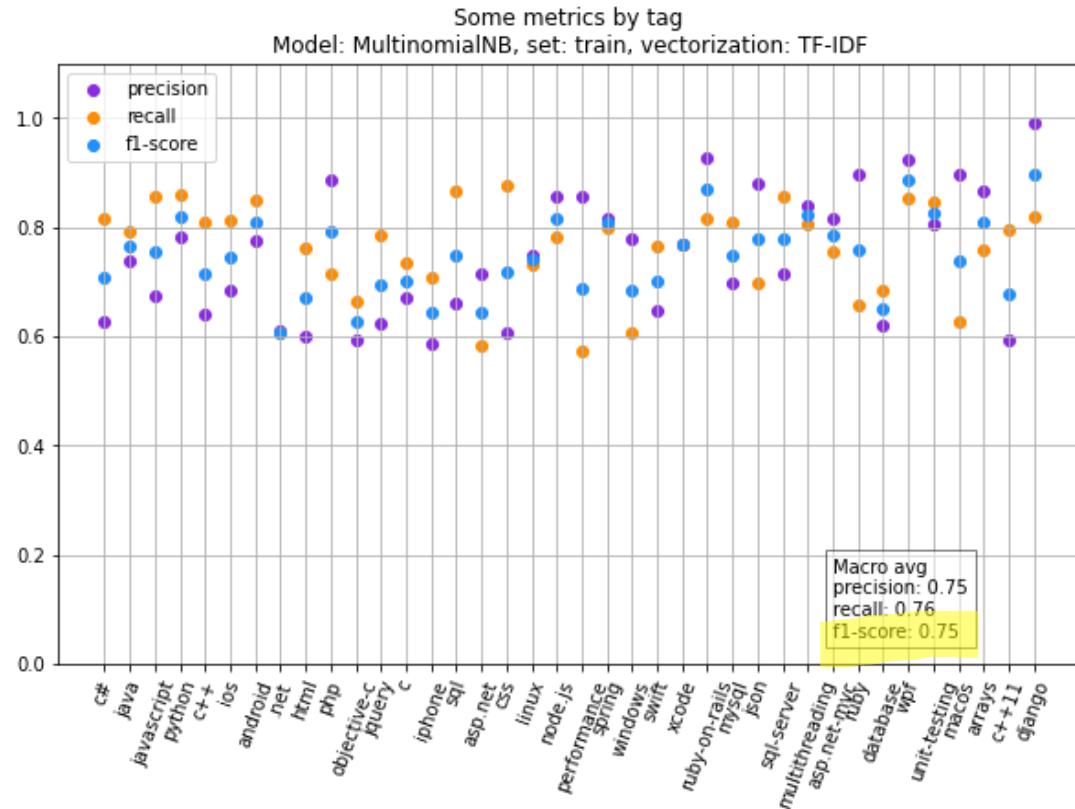
Bag of words: TF-IDF + Naïve-Bayes classifier

```
1 parameters = {  
2     'estimator_alpha' : [0.05, 0.2, 0.5, 1],  
3     'estimator_fit_prior' : [True, False]  
4 }  
5  
6 # Instantiating OneVsRest Classifier  
7 cl = OneVsRestClassifier(MultinomialNB())  
8  
9 # Grid search for best hyperparameters  
10 grid_search = GridSearchCV(  
11     estimator = cl,  
12     param_grid= parameters,  
13     scoring='roc_auc_ovr',  
14     verbose=2,  
15     n_jobs=-1  
16 )
```

Validation croisée
pour trouver
hyperparamètres
adaptés

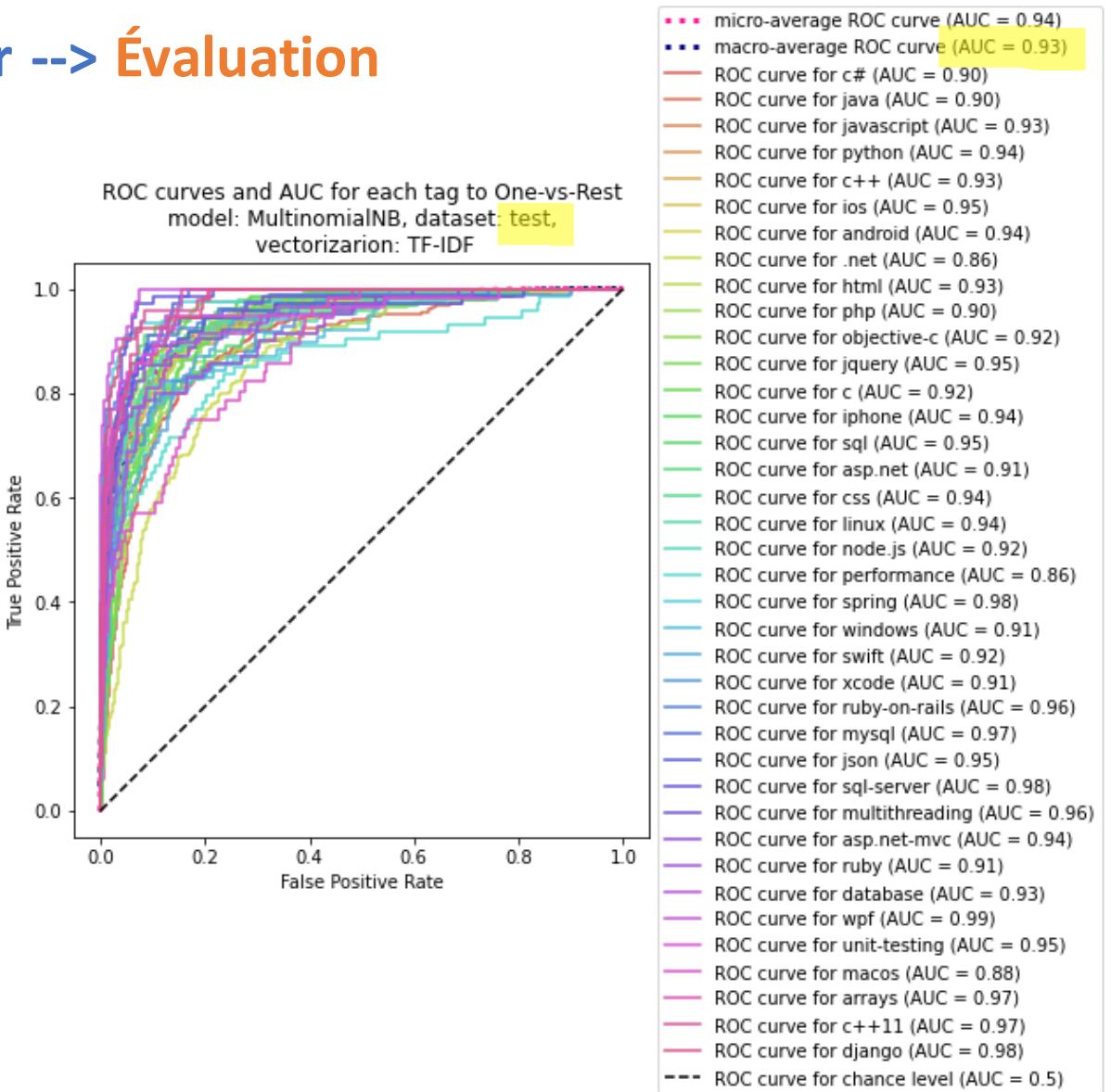
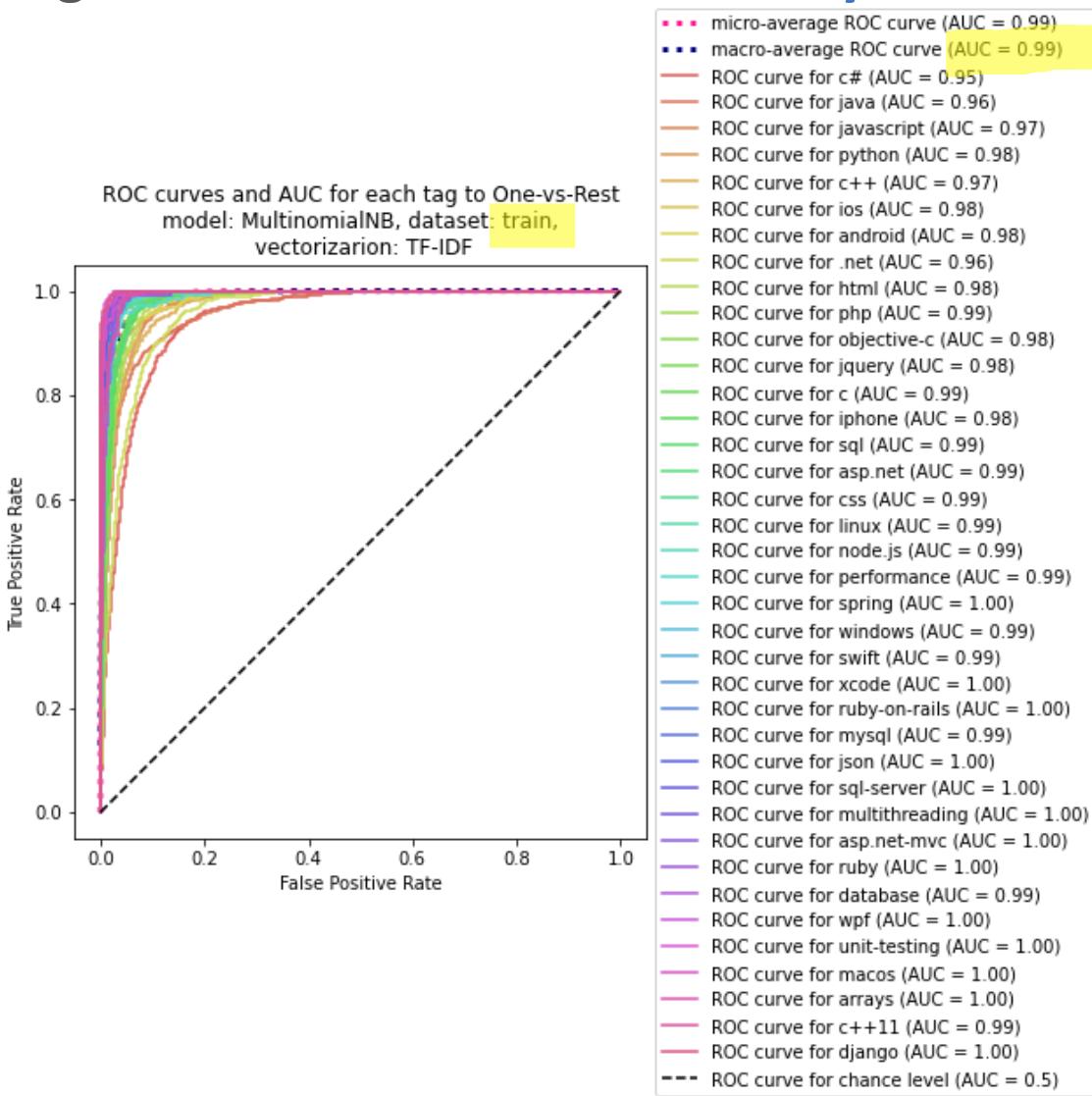
Modélisation (supervisée)

Bag of words: TF-IDF + Naïve-Bayes classifier --> Évaluation



Modélisation (supervisée)

Bag of words: TF-IDF + Naïve-Bayes classifier --> Évaluation



Modélisation (supervisée)

Bag of words: TF-IDF + **RandomForestClassifier**

```

1 # Hyperparameter tuning for RandomForest
2
3 # parameters = {
4 #     'estimator__n_estimators' : [50, 100, 150, 200, 250], 100
5 #     'estimator__max_depth' : [int(x) for x in np.linspace(10, 110, num =
11)], None
6 #     'estimator__min_samples_split' : [2,5,10], 2
7 #     'estimator__min_samples_leaf' : [1,2,4] 2
8 #     'estimator__bootstrap' : [True, False],
9 #     'estimator__max_features' : ['sqrt', 'log2', None] sqrt
10# }
11
12# # Instantiating OneVsRest Classifier
13# cl = OneVsRestClassifier(RandomForestClassifier())
14
15# # Random search for best hyperparameters
16# random_search = RandomizedSearchCV(
17#     estimator = cl,
18#     param_distributions= parameters,
19#     n_iter=30,
20#     scoring='roc_auc_ovr',
21#     verbose=2,
22#     random_state=5,
23#     n_jobs=-1
24# )

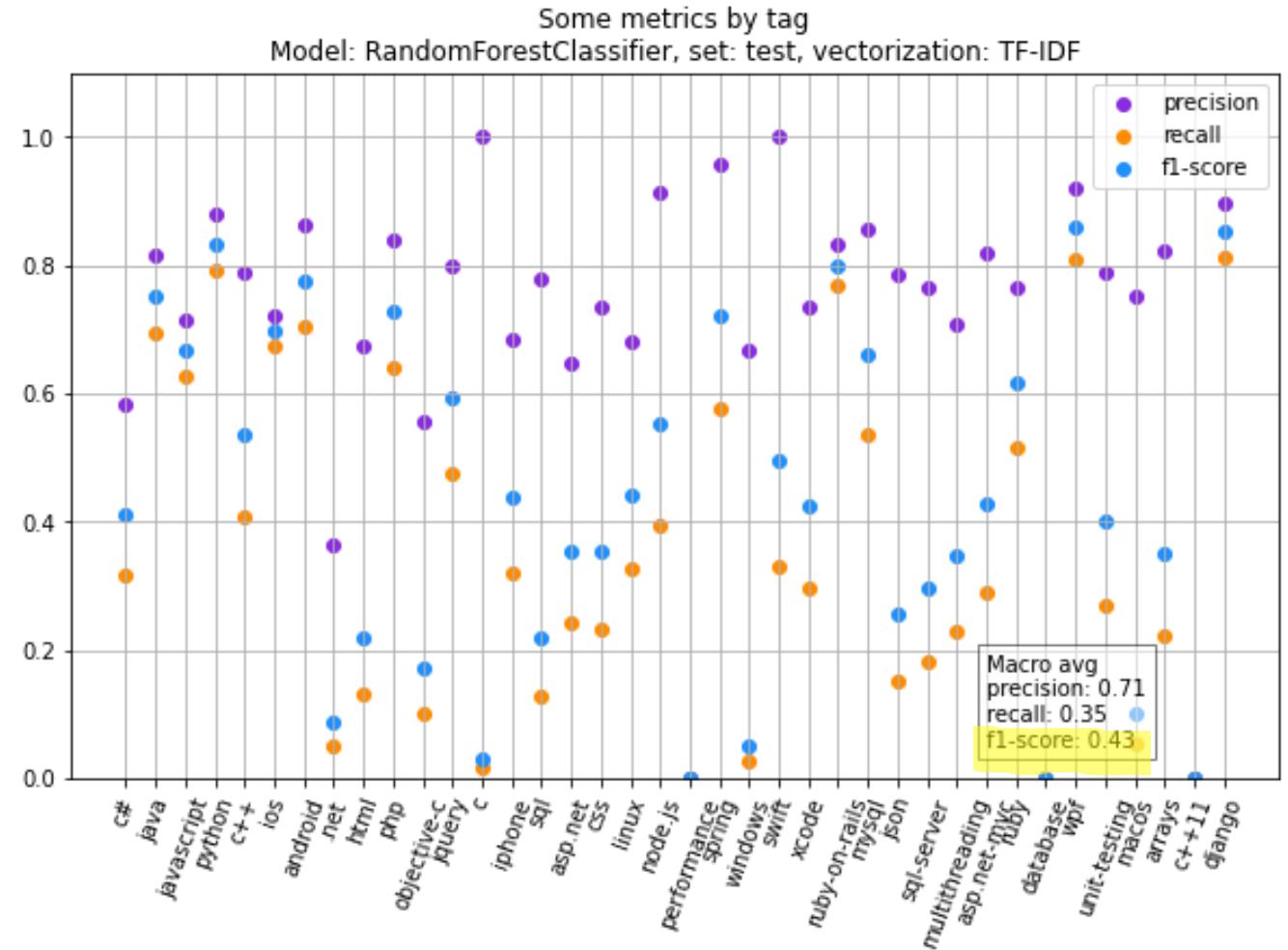
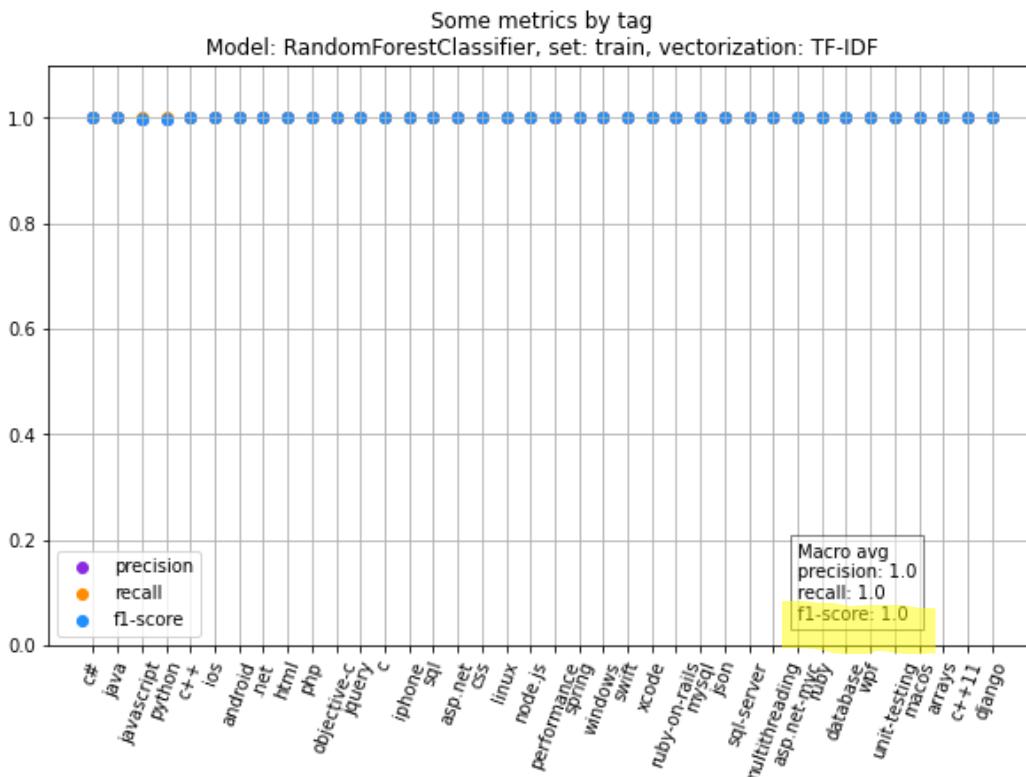
```

Nous ressources de calcul
n'ont pas permis de faire une validation croisée pour RandomForest.

Solution: entraînement (très long) avec paramètres par défaut.

Modélisation (supervisée)

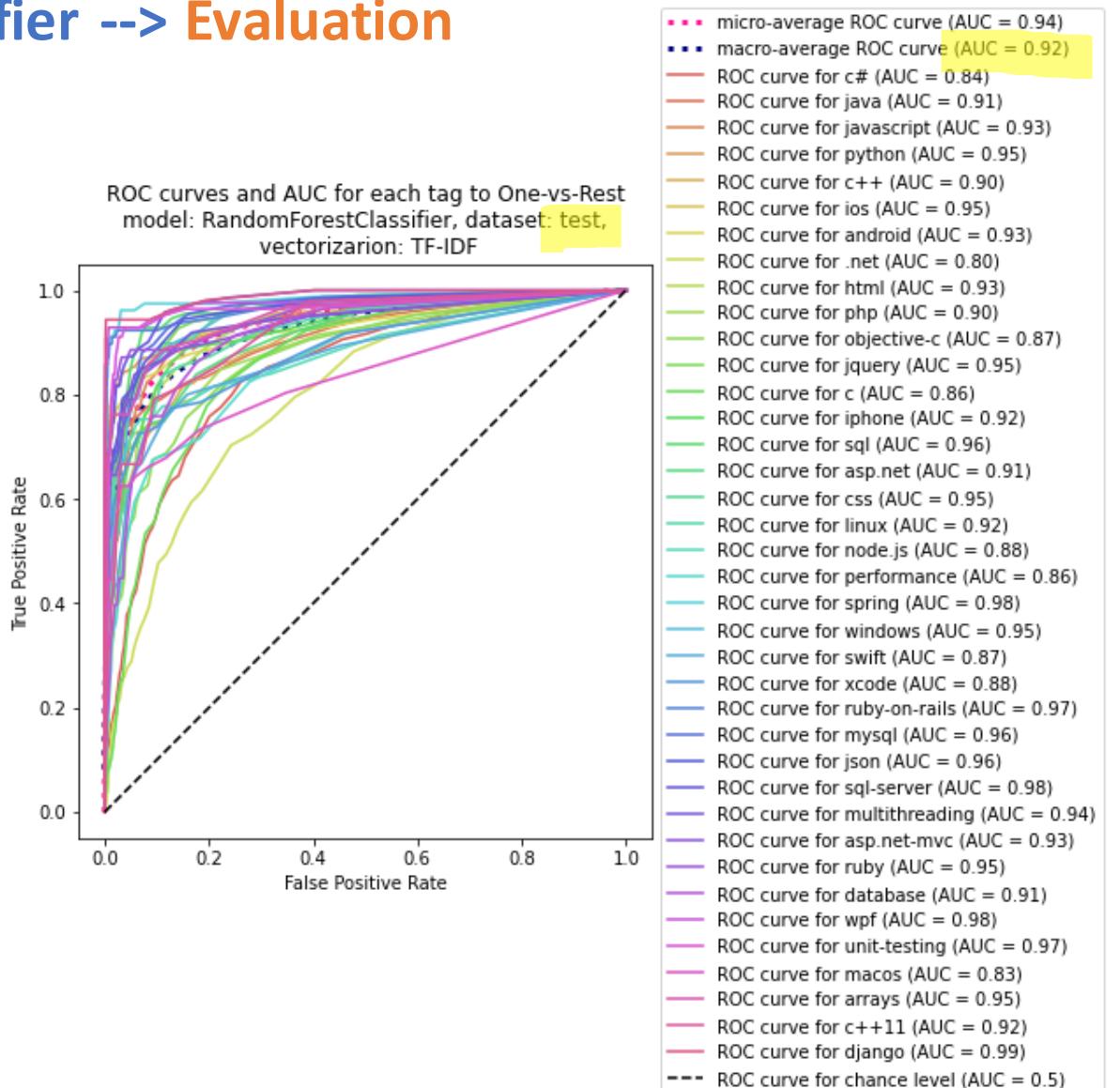
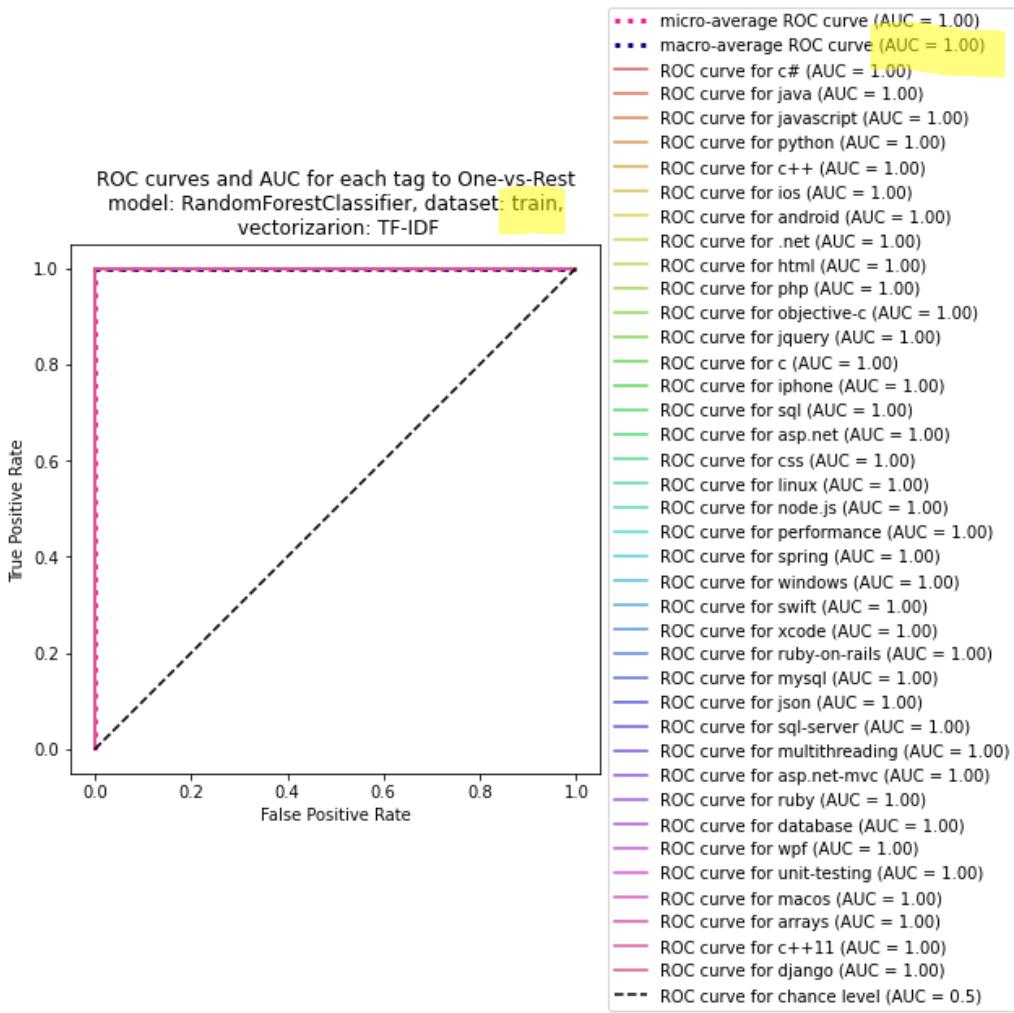
Bag of words: TF-IDF + **RandomForestClassifier** --> **Évaluation**



Modélisation (supervisée)

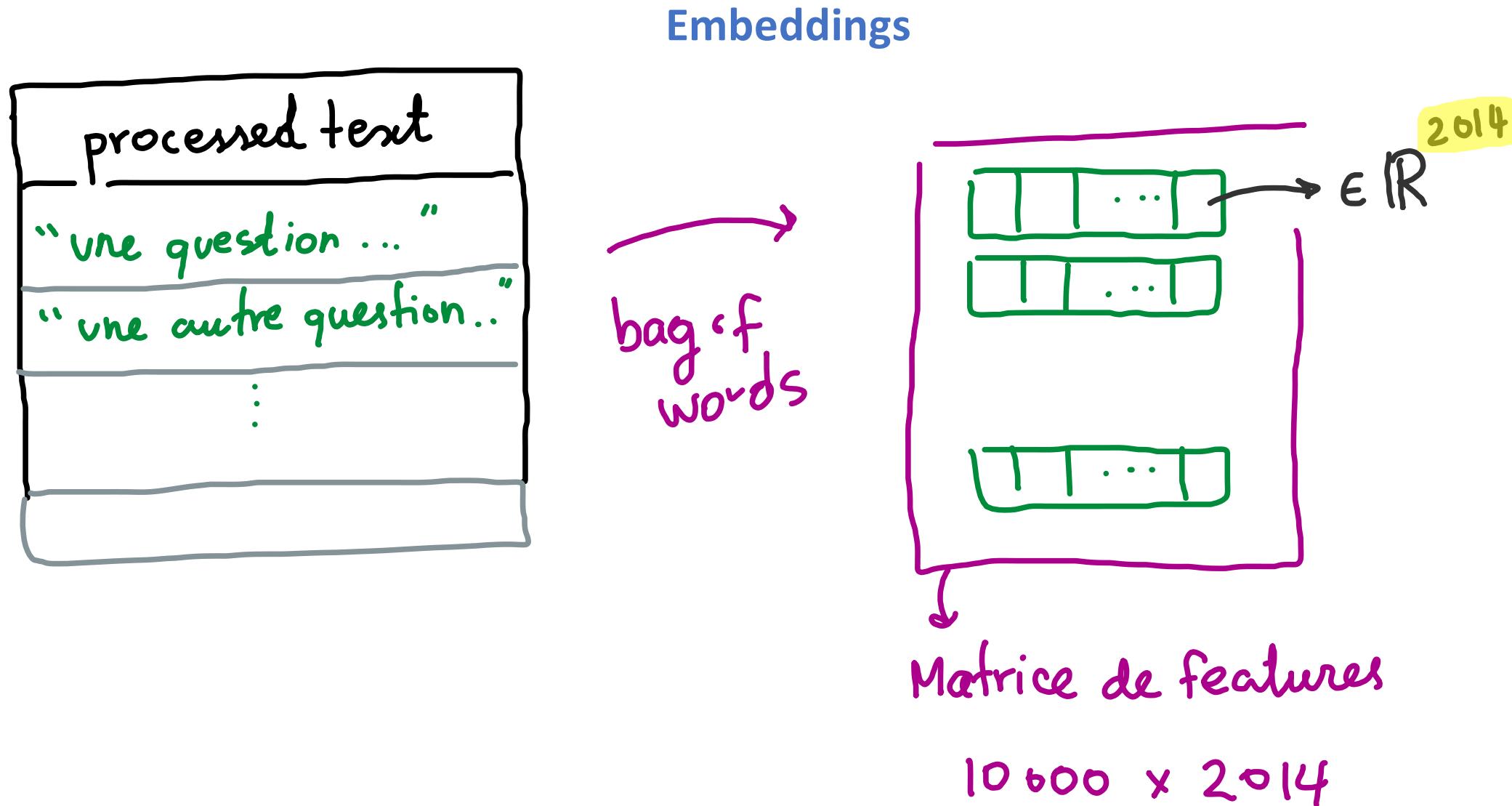
3

Bag of words: TF-IDF + **RandomForestClassifier** --> **Évaluation**



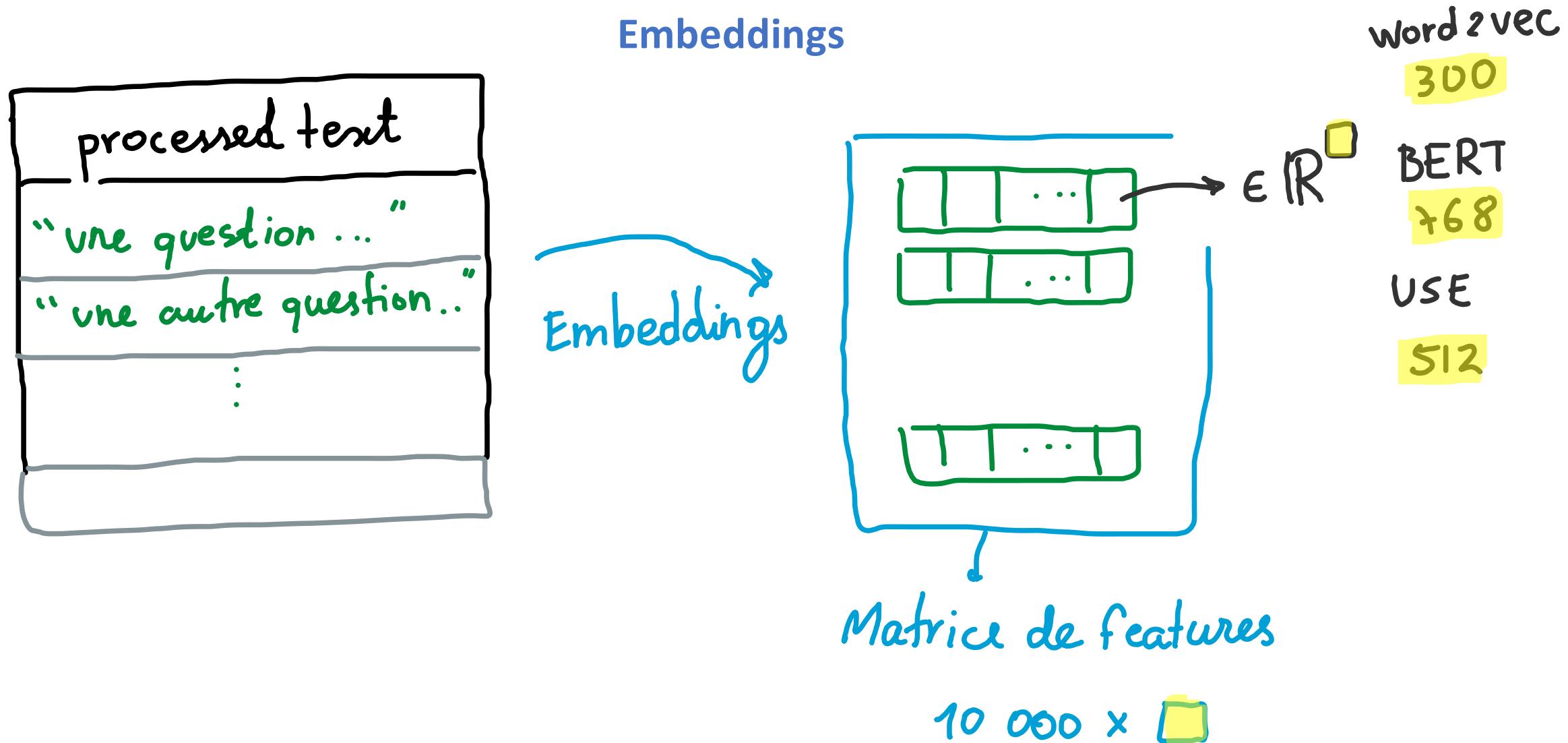
Modélisation (supervisée)

3



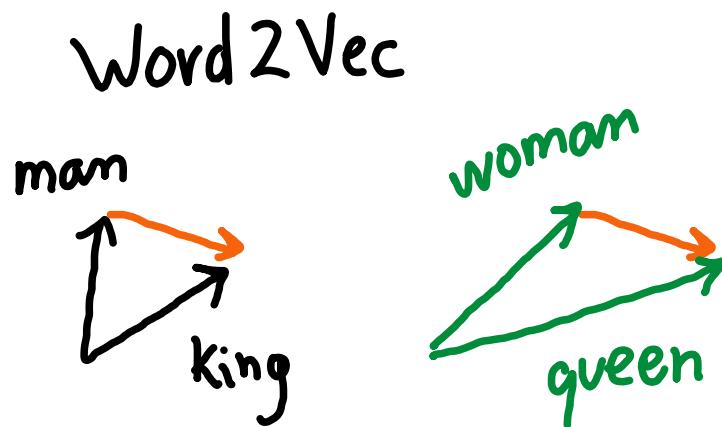
Modélisation (supervisée)

3



Modélisation (supervisée)

3



- basée : réseaux de neurones

- CBOW, skipgram

Embeddings

BERT

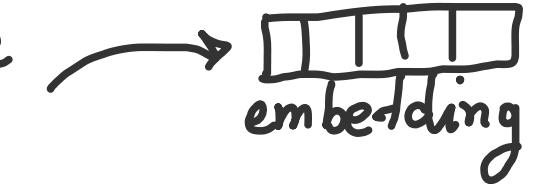
Bidirectional Encoder Representations
from Transformers

- pré-entraînement en ~~masquant~~ des mots

USE

Universal Sentence Encoder

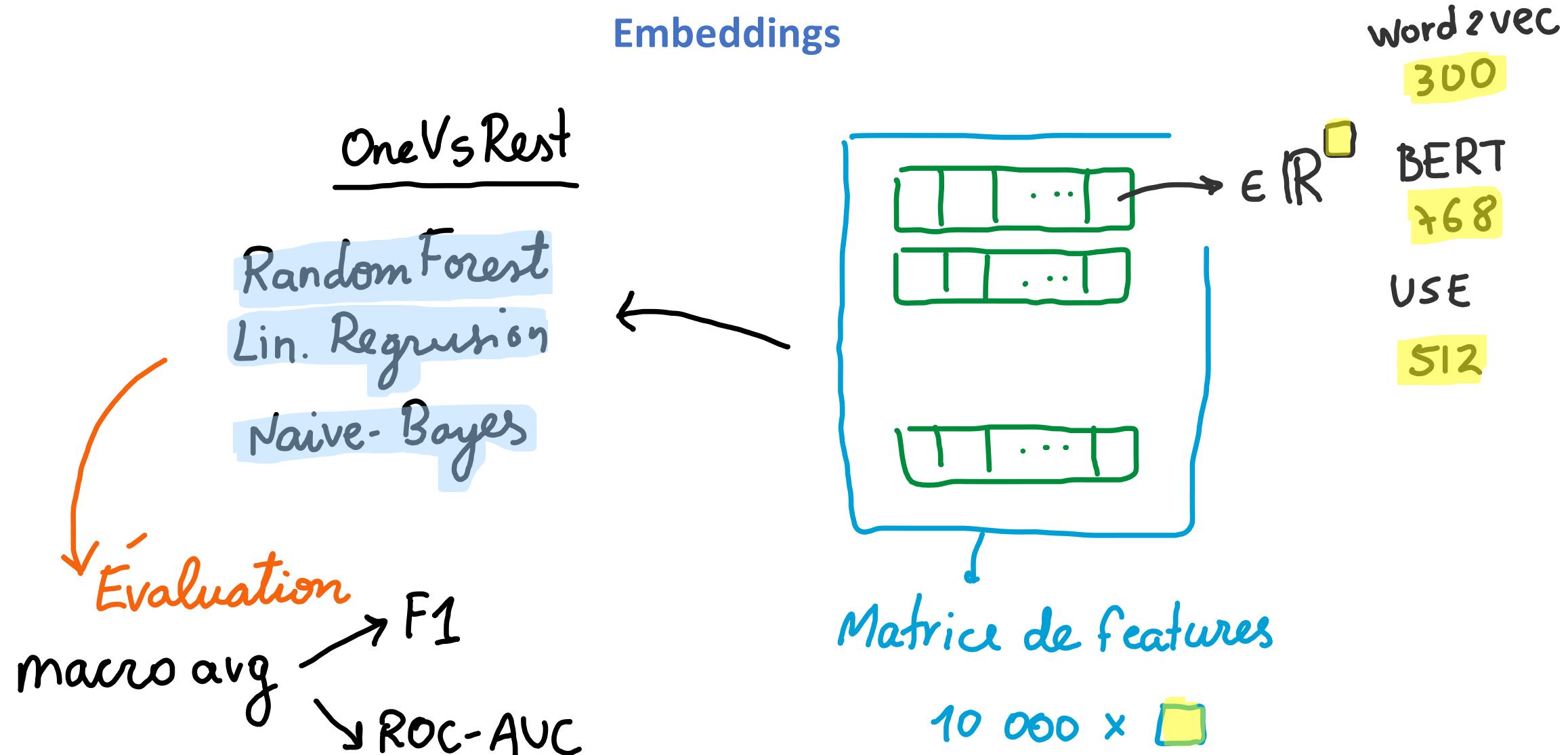
phrase
entière



basée : *Transformers*

Modélisation (supervisée)

3



Modélisation (supervisée)

3

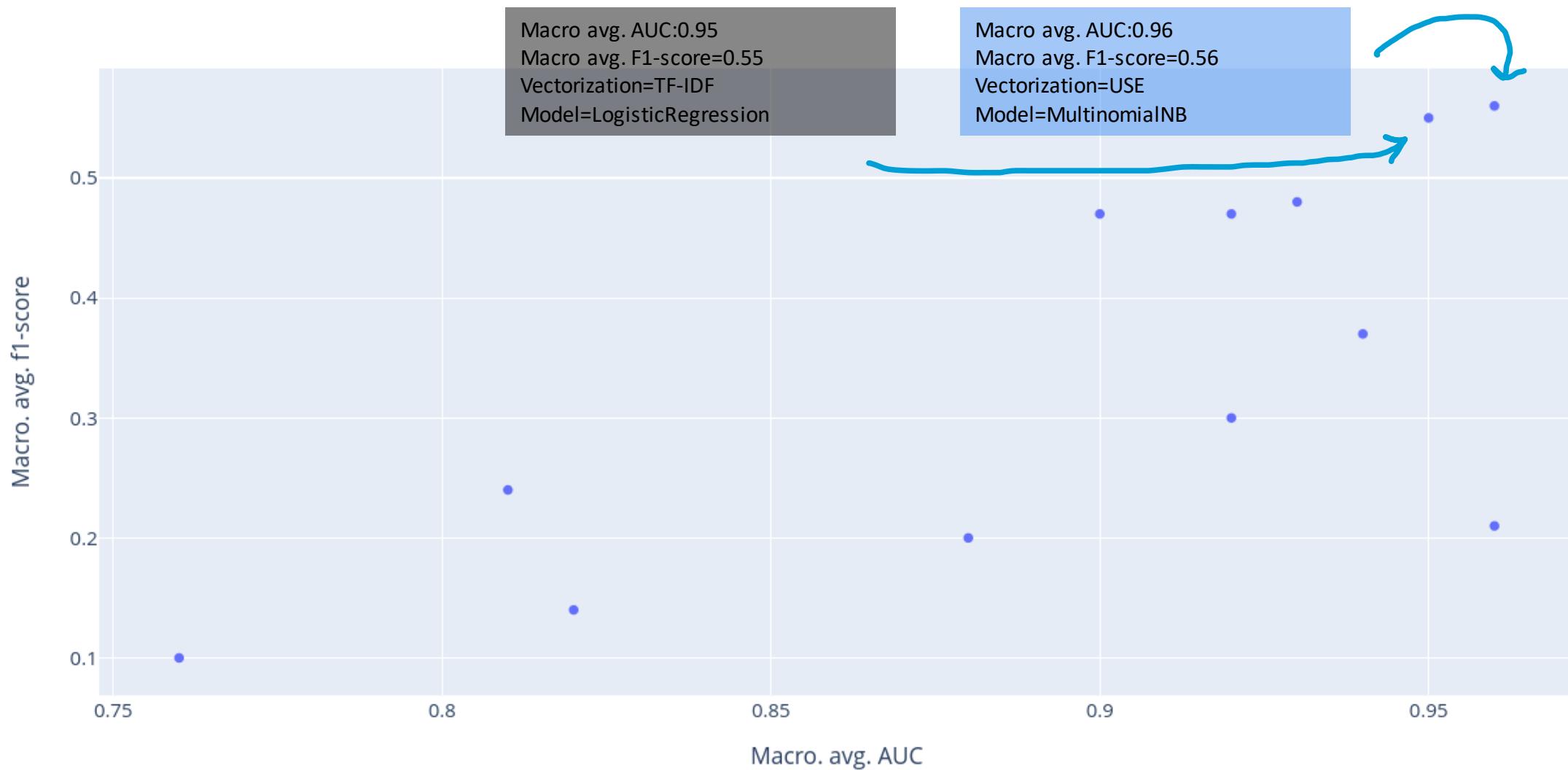
Comparaison performances tous les modèles supervisés

	Vectorization	Model	Macro. avg. f1-score	Macro. avg. AUC	Fit time
0	TF-IDF	LogisticRegression	0.55	0.95	3.87
1	TF-IDF	MultinomialNB	0.48	0.93	3.87
2	TF-IDF	RandomForestCl	0.43	0.92	86.42
3	Word2Vec	LogisticRegression	0.47	0.92	25.08
4	Word2Vec	RandomForest	0.15	0.82	401.76
5	Word2Vec	MultinomialNB	0.20	0.88	401.76
6	BERT	LogisticRegression	0.47	0.90	259.71
7	BERT	RandomForest	0.10	0.76	984.06
8	BERT	MultinomialNB	0.24	0.81	1.91
9	USE	LogisticRegression	0.21	0.96	19.21
10	USE	RandomForest	0.37	0.94	443.52
11	USE	MultinomialNB	0.56	0.96	0.62



Modélisation (supervisée)

Comparaison performances tous les modèles supervisés



Modélisation (supervisée)

3

Comparaison performances tous les modèles supervisés

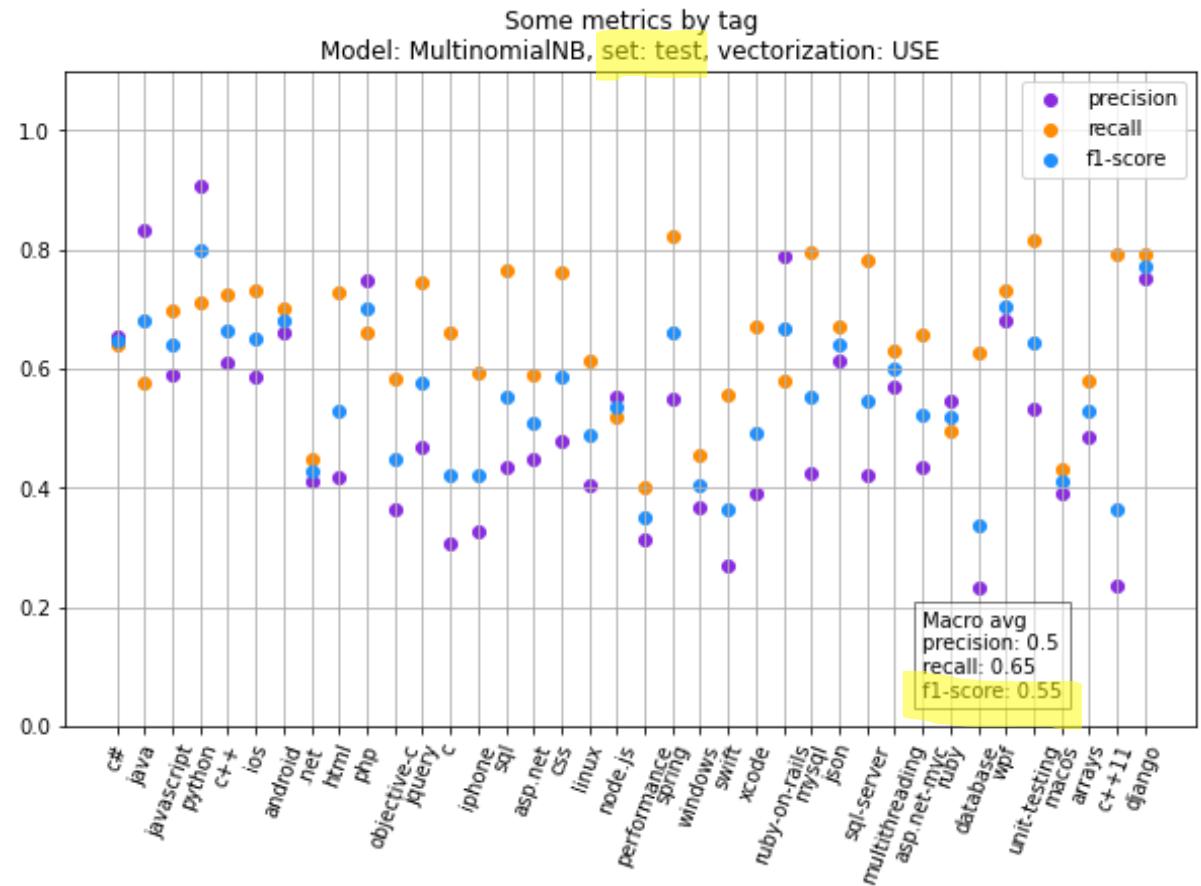
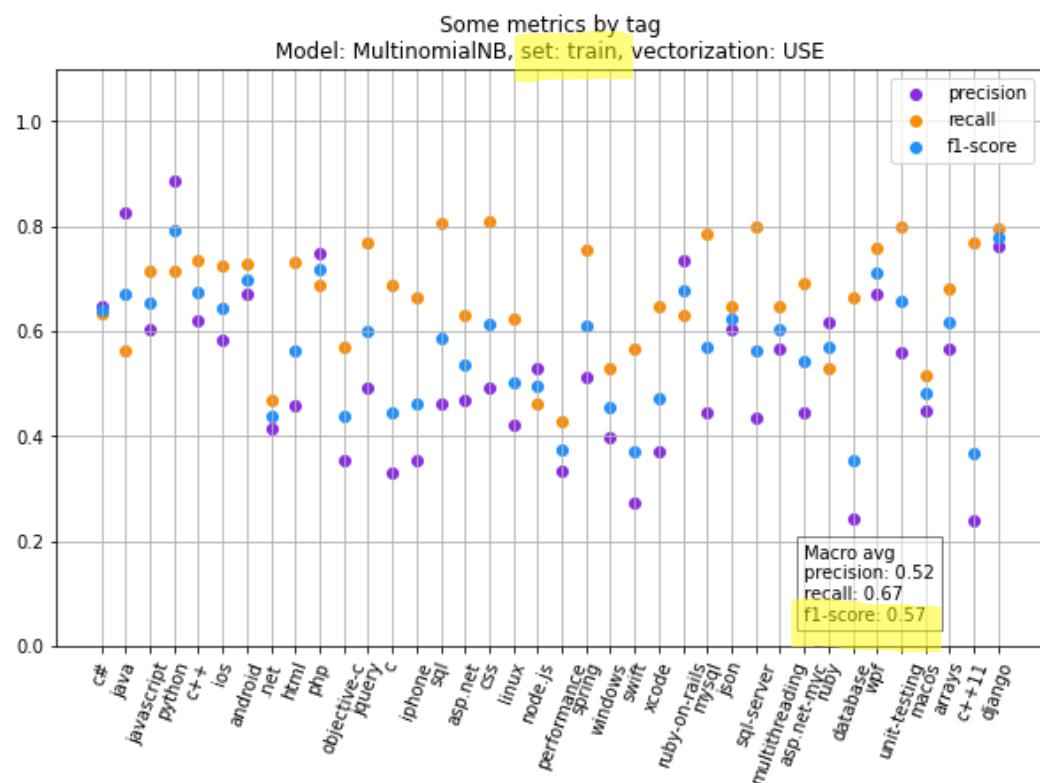
	Vectorization	Model	Macro. avg. f1-score	Macro. avg. AUC	Fit time
0	TF-IDF	LogisticRegression	0.55	0.95	3.87
1	TF-IDF	MultinomialNB	0.48	0.93	3.87
2	TF-IDF	RandomForestCl	0.43	0.92	86.42
3	Word2Vec	LogisticRegression	0.47	0.92	25.08
4	Word2Vec	RandomForest	0.15	0.82	401.76
5	Word2Vec	MultinomialNB	0.20	0.88	401.76
6	BERT	LogisticRegression	0.47	0.90	259.71
7	BERT	RandomForest	0.10	0.76	984.06
8	BERT	MultinomialNB	0.24	0.81	1.91
9	USE	LogisticRegression	0.21	0.96	19.21
10	USE	RandomForest	0.37	0.94	443.52
11	USE	MultinomialNB	0.56	0.96	0.62



Modélisation (supervisée)

3

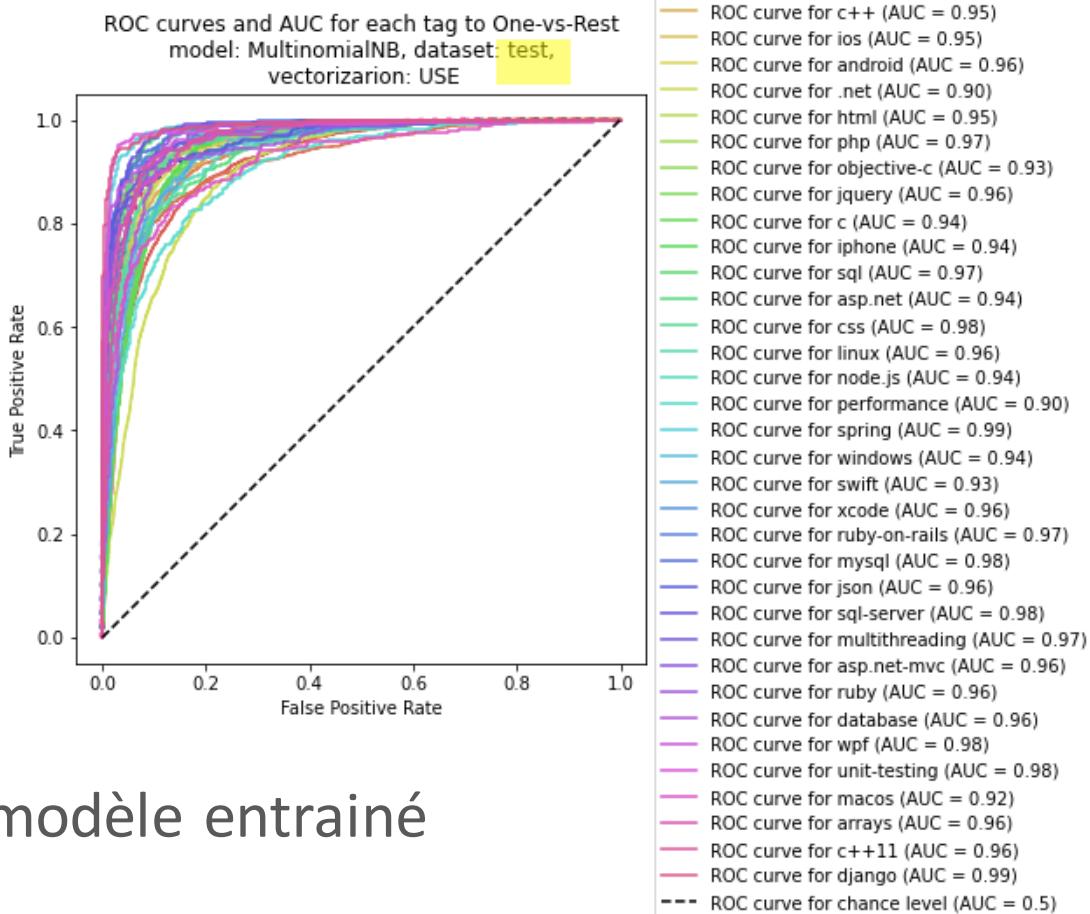
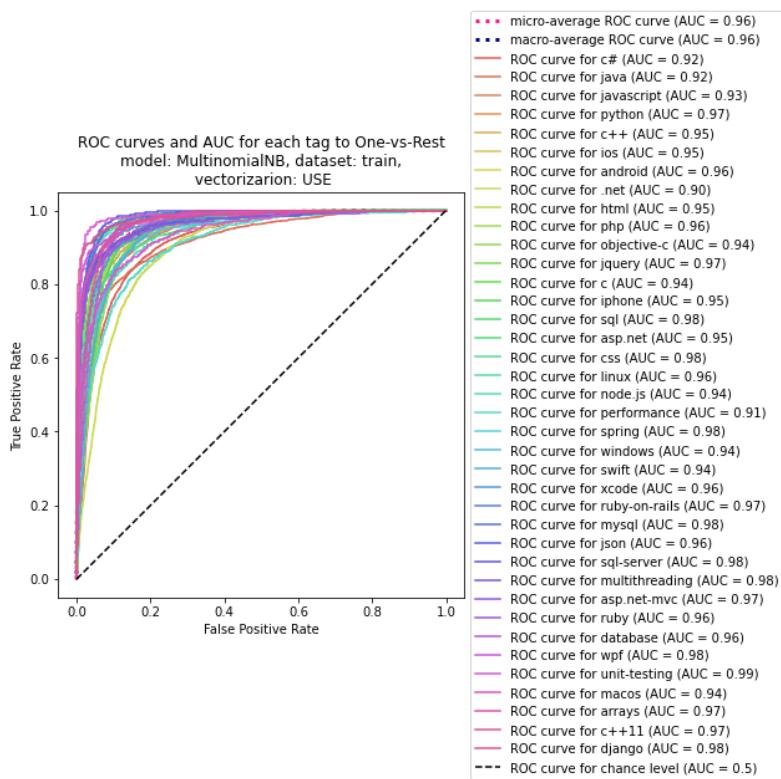
Entraînement OneVsRest(Naïve Bayes Multinomial) avec toutes les données



Exporter modèle entraîné

Modélisation (supervisée)

Entraînement OneVsRest(Naïve Bayes Multinomial) avec toutes les données

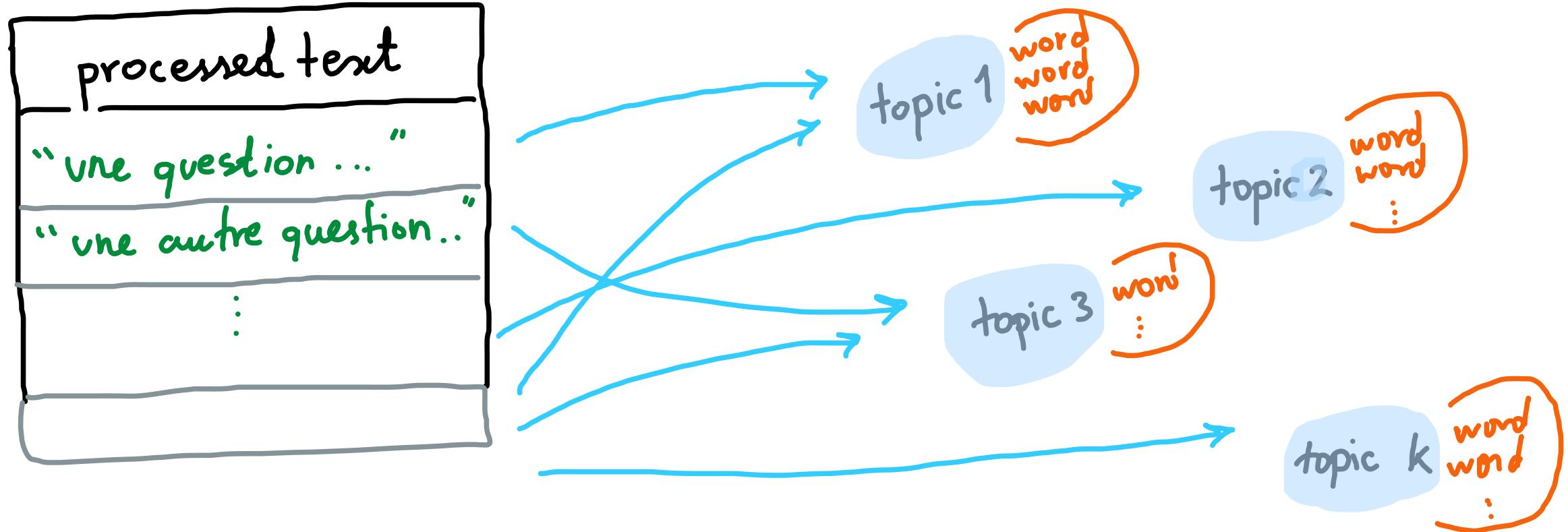


Exporter modèle entraîné

Modélisation (non-supervisée)

3

LDA : Latent Dirichlet Allocation



Modélisation (non-supervisée)

3

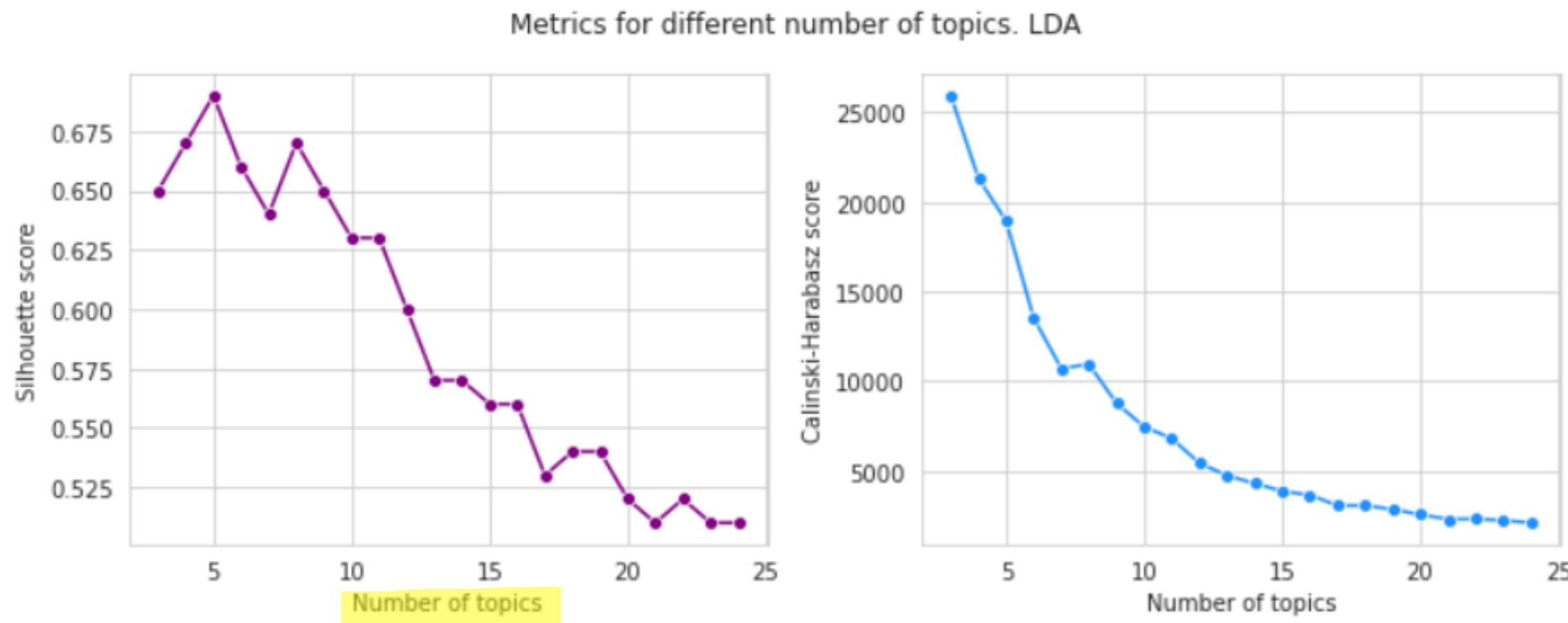
LDA : Latent Dirichlet Allocation

Modélisation (non-supervisée)

3

LDA : Latent Dirichlet Allocation. --> Pas très adapté à notre but !

Mes approches :



hyperparamètre →

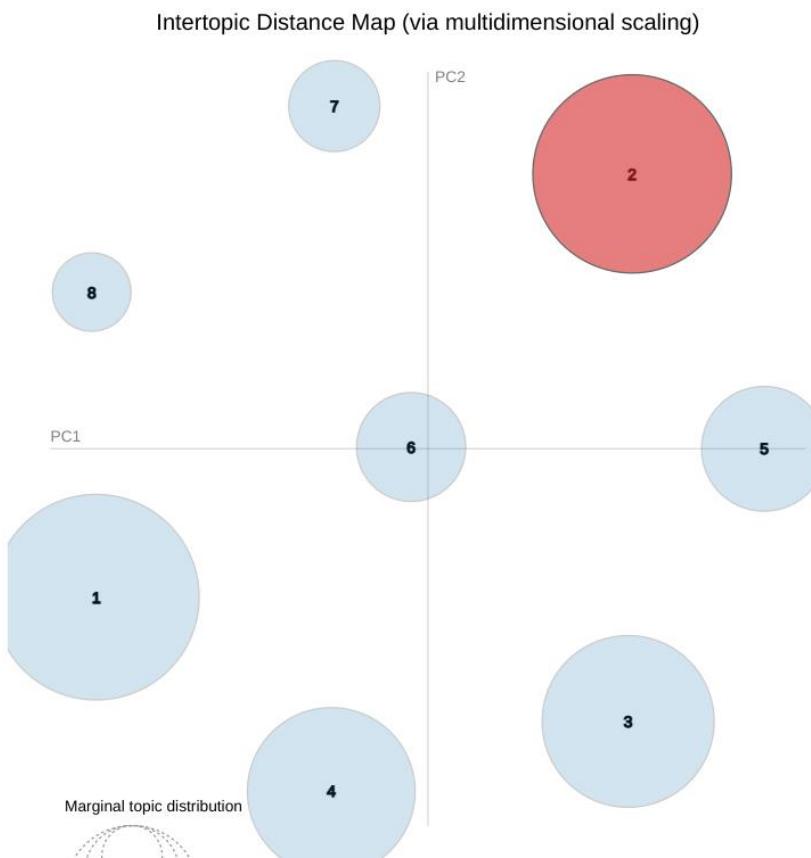
Modélisation (non-supervisée)

3

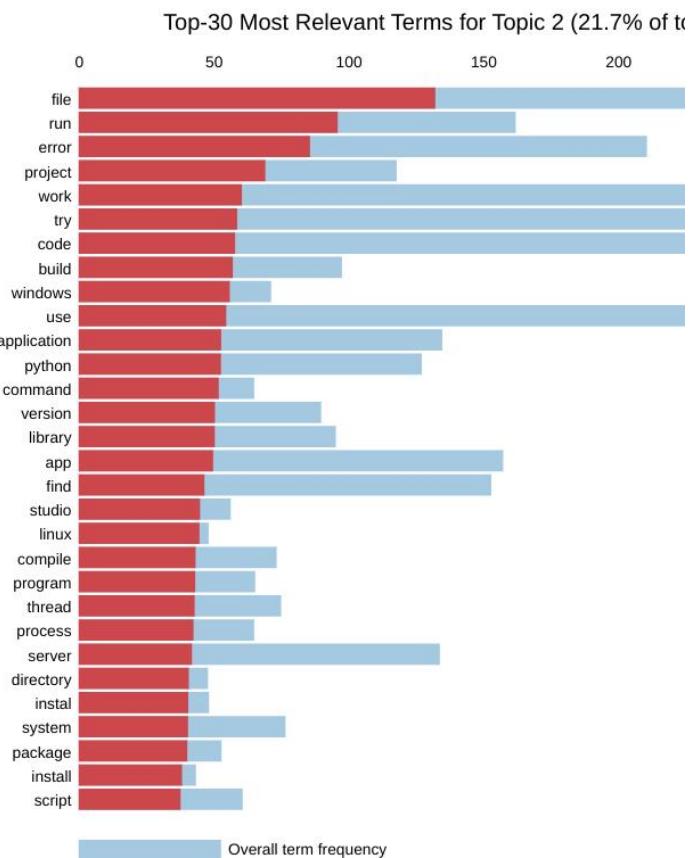
LDA : Latent Dirichlet Allocation. --> Pas très adapté à notre but !

Mes approches :

Selected Topic: 2 Previous Topic Next Topic Clear Topic



Slide to adjust relevance metric:⁽²⁾
 $\lambda = 1$ 0.0 0.2 0.4 0.6



Distribution de mots par sujet
↓
ideé
↓
intersection avec mes tags

Modélisation (non-supervisée)

3

LDA : Latent Dirichlet Allocation. --> Pas très adapté à notre but !

Mes approches :

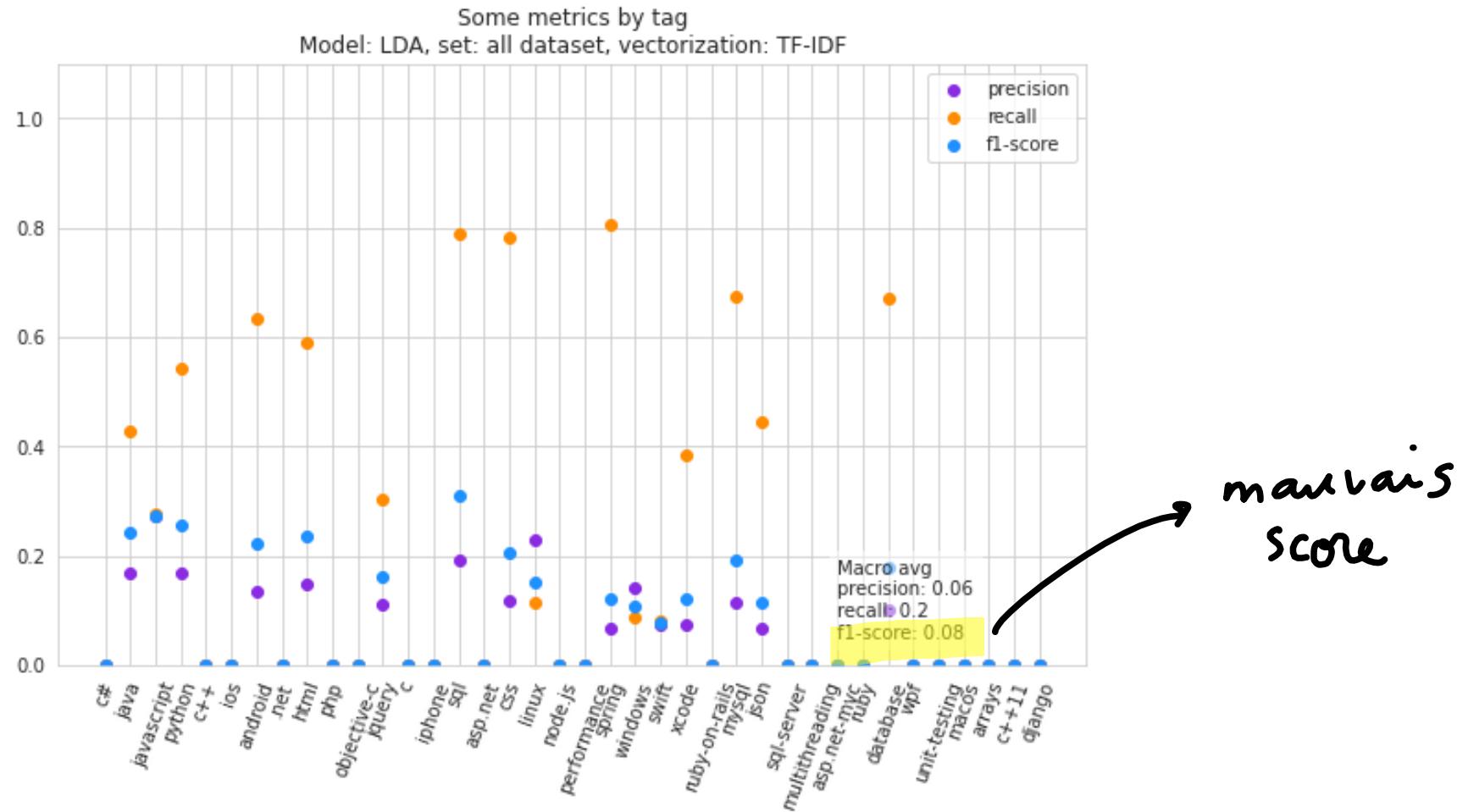


Modélisation (non-supervisée)

3

LDA : Latent Dirichlet Allocation. --> Pas très adapté à notre but !

Mes approches :



4

API

Avec 

```
1 def tag_suggestion(raw_text):
2     """
3         Returns a list of tags suggested for the question raw_text.
4     """
5     # Clean text first
6     clean_text = clean(raw_text)
7     document = [clean_text]
8
9     # Find an embedding of the text with USE
10    X = embed(document)
11
12    # Predict a tag set with our classification model
13    pred = my_pred(X)
14
15    return binary_to_tag_list(pred)
16
17
18
19 # API ##
20
21 demo = gr.Interface(fn=tag_suggestion,
22                      inputs="text",
23                      outputs=["text"],
24                      examples=examples)
25
26
27 if __name__ == "__main__":
28     demo.launch()
29
```

1 Nettoyer texte avec notre fonction

↓ embed avec USE



↓ prediction
OneVsRest
Naive Bayes



↓ traduction

javascript

php

Hébergé sur spaces de



Hugging Face

[aller à l'API](#)

The screenshot shows a web interface for the Hugging Face StackOverflowTagSuggestion space. At the top, there's a navigation bar with links for Models, Datasets, Spaces, Docs, Solutions, Pricing, and a user profile icon. A search bar is also present. A yellow banner at the top of the main content area says "Hugging Face is way more fun with friends and colleagues! 🤗" with a "Join an organization" button and a "Dismiss this message" button. Below the banner, the space title is "ana-bernal/StackOverflowTagSuggestion" with a "Running" status indicator and a "Open logs" link. The main content area has tabs for App, Files and versions, Community, and Settings. On the left, a red box highlights the "raw_text" section, which contains several paragraphs of text from a Stack Overflow post about jQuery opacity animations. A blue arrow points from a yellow box labeled "Entrée texte" to this section. On the right, another red box highlights the "output" section, which displays the predicted tags: ["javascript", "html", "jquery", "css"]. A blue arrow points from a yellow box labeled "sortie : tags prédictes" to this section. The entire interface is framed by a red border.

Entrée texte

raw_text

Jquery/Javascript Opacity animation with scroll <p>I'm looking to change the opacity on an object (and have the transition be animated) based on a users scroll.
example(<http://davegamahe.com/>)</p>

<p>I've searched everywhere
like here, but it ends up pointing me to the waypoints plugin
(<http://stackoverflow.com/questions/6316757/opacity-based-on-scroll-position>)</p>

<p>I've implemented the [waypoints][1] plugin and have the object fading once it's higher than 100px. [Using the offset attribute] but would like to basically control the opacity of an object and have the animation be visible like the above example.</p>

<p>I've searched all over- this is my last resort.
Any help is greatly appreciated.</p>

sortie : tags prédictes

["javascript", "html", "jquery", "css"]

5

Conclusions

Suggestions des tags + Projet OC

- Fonctionnement **satisfaisant** avec le meilleur modèle.
- Beaucoup de **nouveaux apprentissages**: NLP, version control, API avec Gradio et HuggingFace.
- Traitement de texte : cas **complexe** pour un premier approche à NLP --> du code dans le texte, HTML, etc.

Pour aller + loin

- Inclure plus de tags: il faut plus de ressources de calcul. (nous: 40 sur 18 000)
- Inclure plus de questions (nous: 50 000 sur ∞)
- Améliorer visualisation API, rajouter des options.
- Prédictions avec des réseaux de neurones.
- Analyse exploratoire + profond.