

# Catégorisez automatiquement des questions

Projet 5 du parcours Machine Learning Engineer

## **Présentation**

Ce projet vise à développer un modèle de machine learning capable de prédire les tags associés à une question donnée sur StackOverflow.

### **Objectifs:**

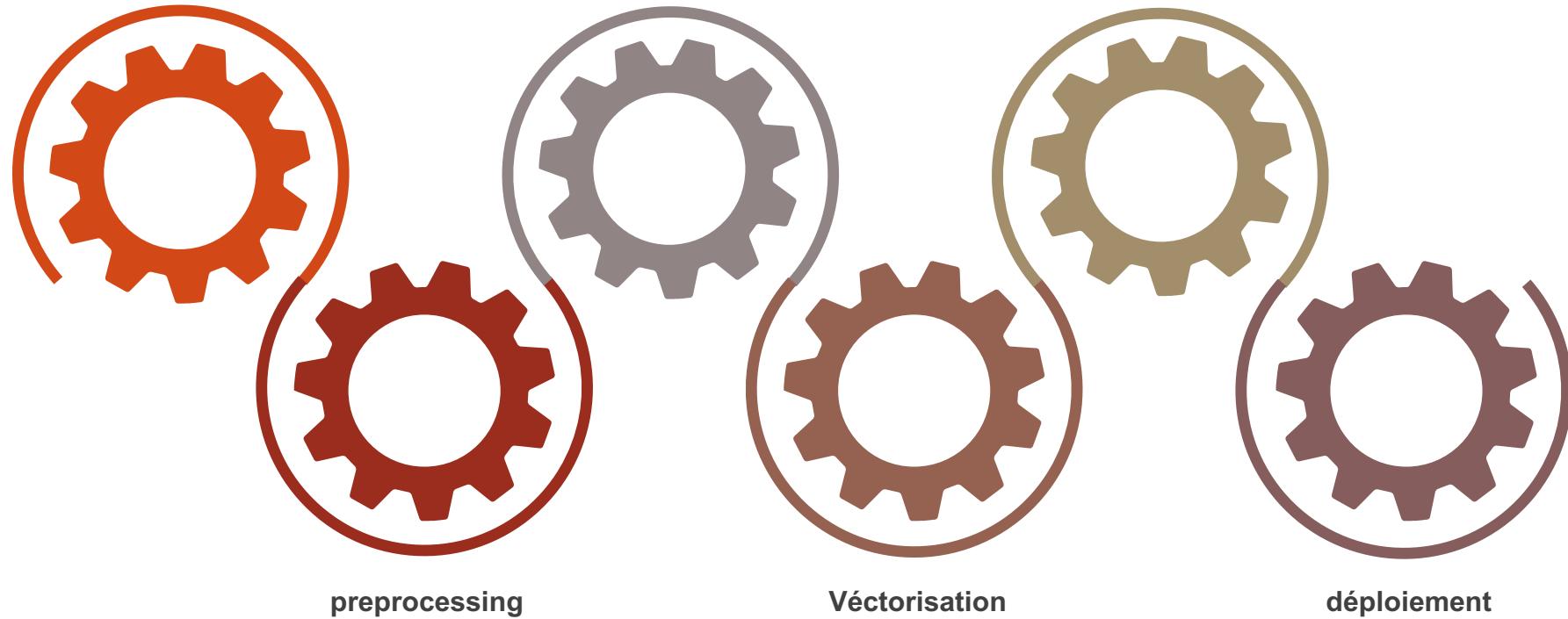
- Récupérer un jeu de données via une API
- Prétraiter des données de type texte
- Tester différentes approches de machine learning
- Développer un modèle de machine learning
- Déploiement sur le Cloud

# Sommaire

Téléchargement des données

Exploration

Prediction



# Le jeux de données

Le jeu de donnée a été recuperé grâce à StackExchange Data Explorer Avec la commande SQL suivante :

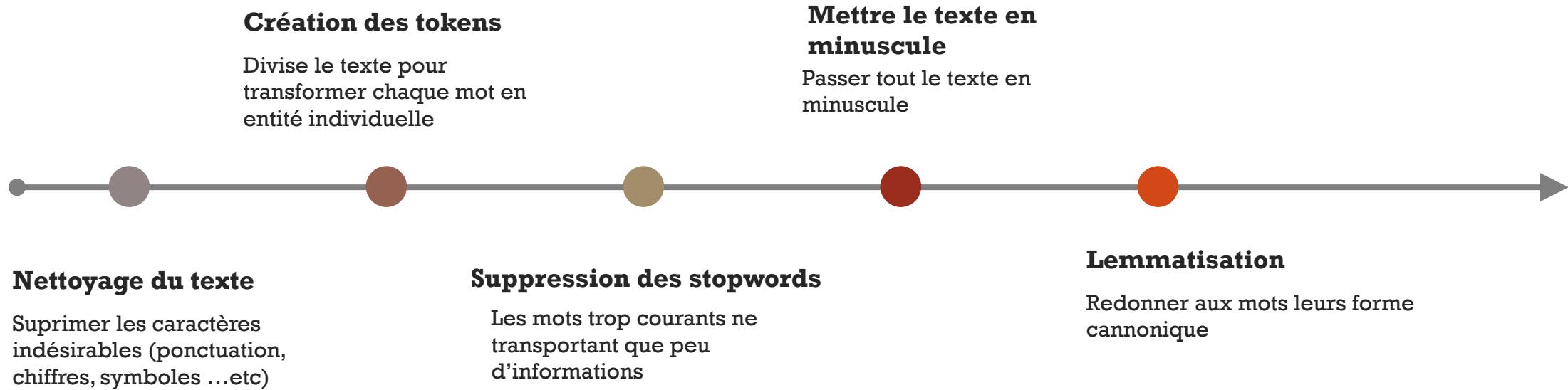
```
SELECT TOP 50000 Title, Body, Tags, Id, Score, ViewCount, AnswerCount
FROM Posts
WHERE PostTypeId = 1
AND ViewCount > 100
AND AnswerCount > 5
AND LEN(Tags) - LEN(REPLACE(Tags, '<', '')) >= 3
ORDER BY LEN(Tags) DESC;
```

## Résultats:

	Title	Body	Tags	Id	Score	ViewCount	AnswerCount
0	Android Jetpack Navigation, BottomNavigationView...	<p>Android Jetpack Navigation, BottomNavigationVi...	<android><android-architecture-components><bot...	50577356	81	67388	14
1	JetPack Compose Button with drawable	<p>How can we achieve this in jetpack compose<...	<android><android-jetpack-compose><android-com...	72336943	10	17903	6
2	How to handle back button when at the starting...	<p>I've started working with the new navigatio...	<android-architecture-components><android-arch...	50937116	17	23616	8
3	TopAppBar flashing when navigating with Compos...	<p>I have 2 screens which both have their own ...	<android><android-jetpack><android-jetpack-com...	68633717	19	4757	9
4	How to create recycler view in Compose Jetpack?	<p>Is there any special way to create recycler...	<android><android-recyclerview><android-jetpac...	58691725	17	12553	7

Shape of Df: (50000, 7)

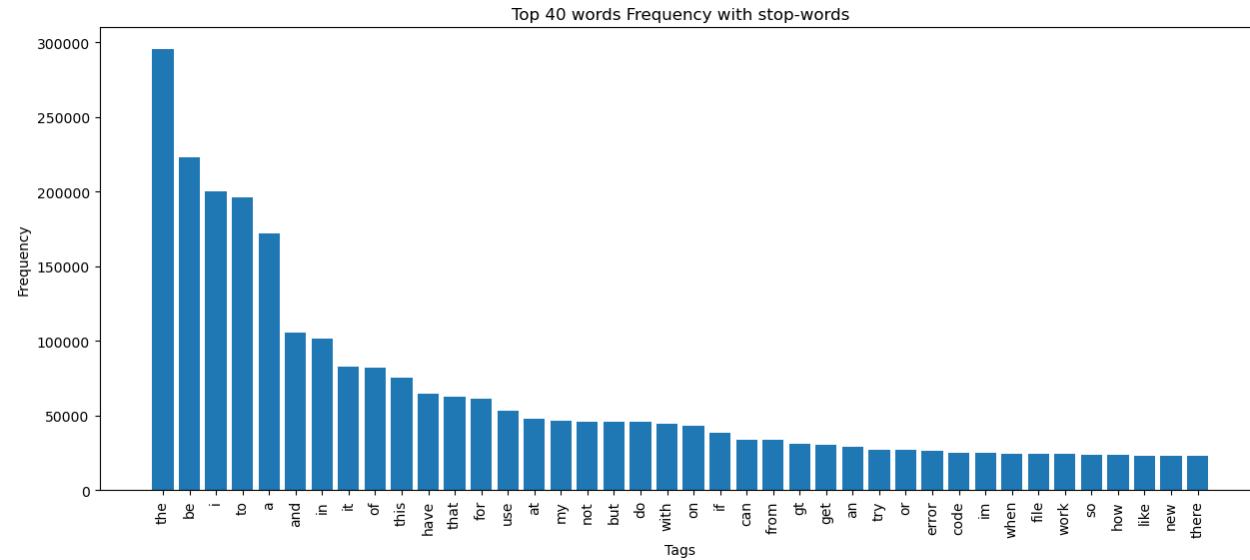
# Preprocessing



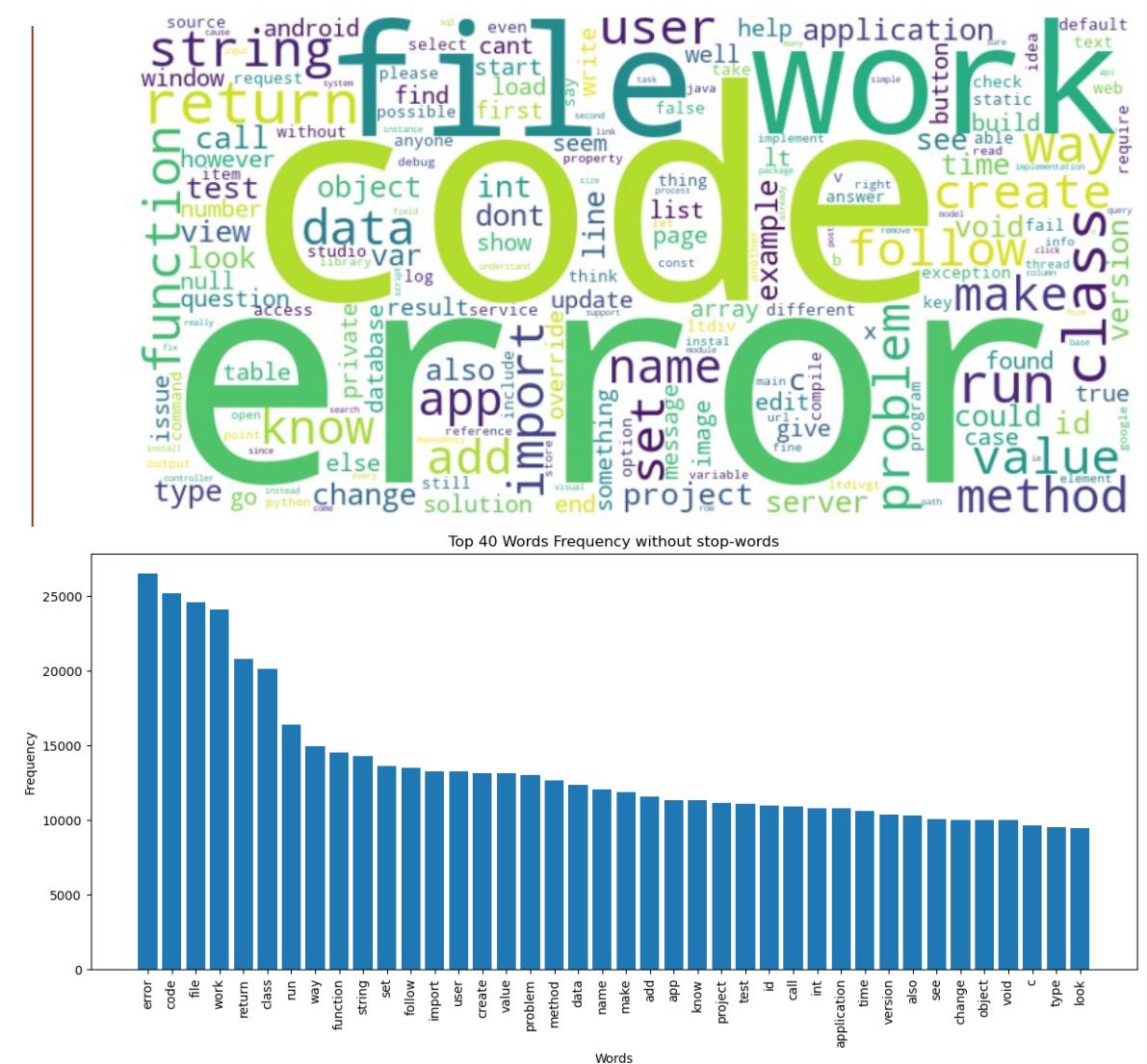
# Fréquence des mots (Body et Title combinés)

Les stop words sont des mots courants (comme "le", "et", "de") qui n'apportent pas de valeur significative à l'analyse du texte en NLP. Les supprimer permet de réduire la taille des données et d'améliorer l'efficacité et la précision des modèles.

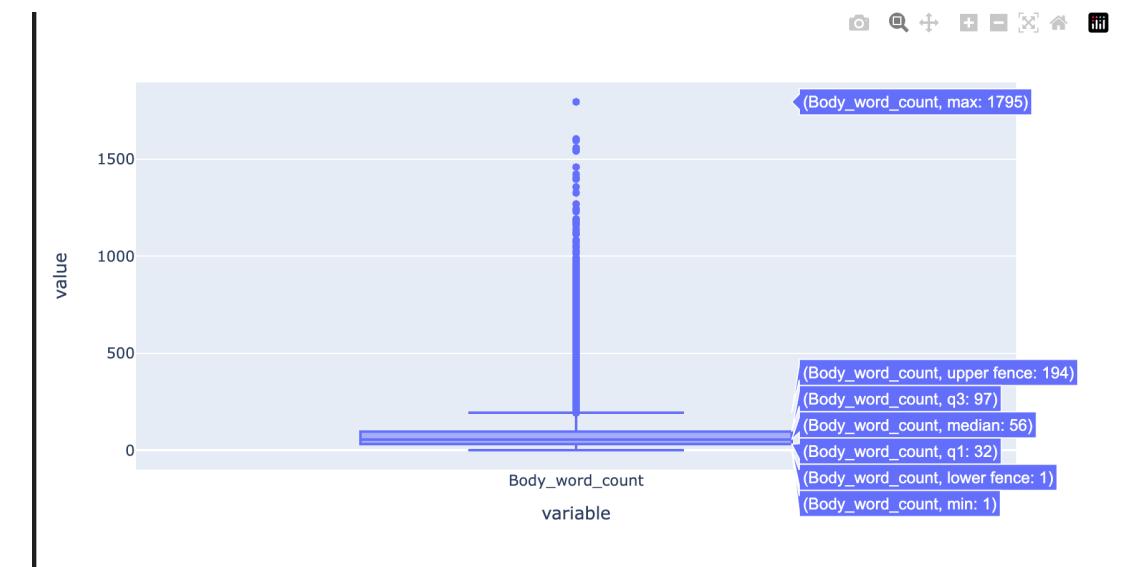
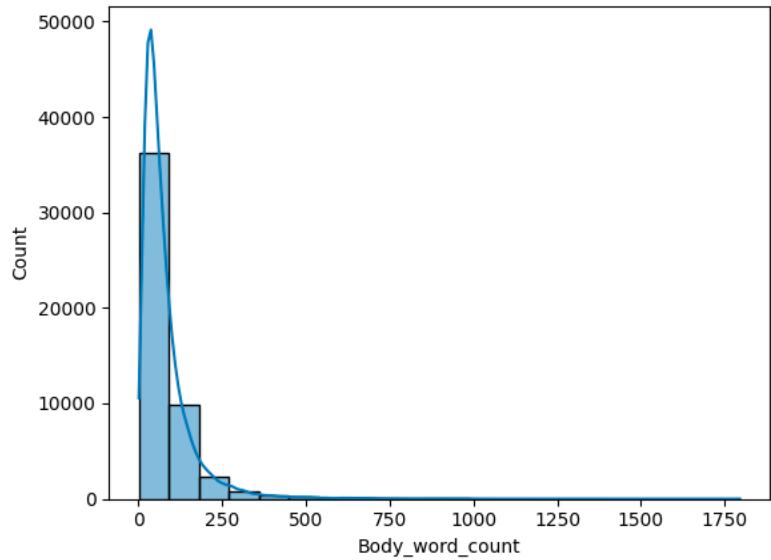
## Avec les stop-words



## Sans les stop-words

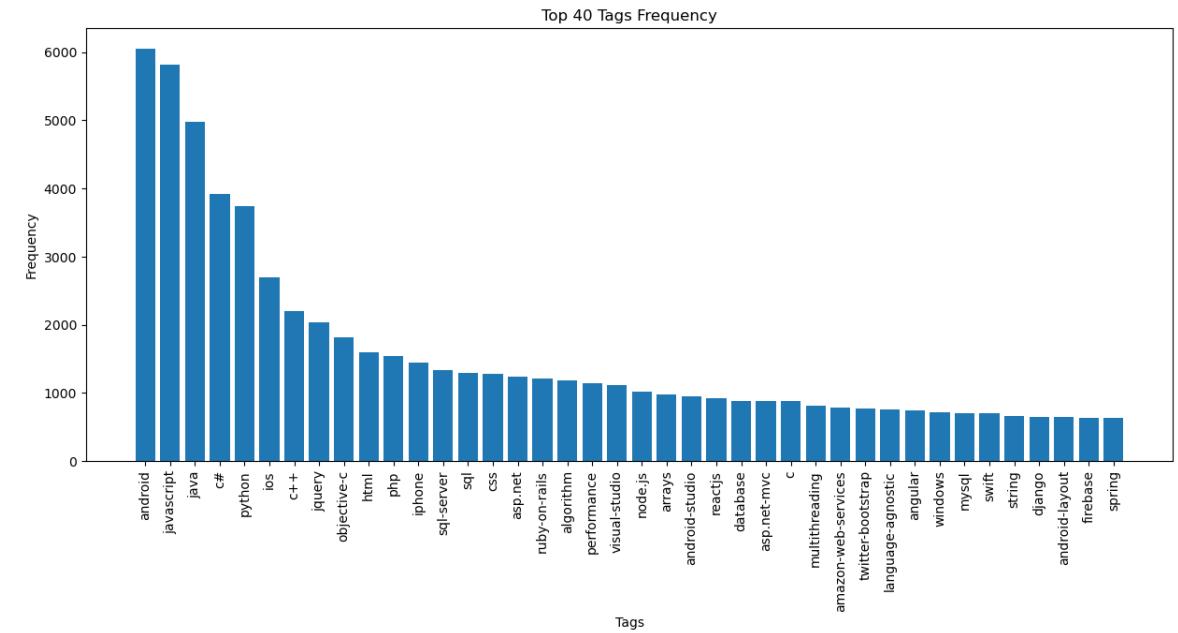


# Exploration Body



Les deux graphiques montrent la distribution de la variable Body. L'histogramme montre que la majorité des post possèdes moins de 100 mots ce qui est confirmé par le boxplot avec une mèdiane de 56 mots.

# Fréquence des tags



	Title	Body	Tags	Body_word_count
0	android jetpack navigation bottomnavigationvie...	android jetpack navigation bottomnavigationvie...	android android-architecture-components bottom...	131
1	jetpack compose button drawable	achieve jetpack compose something button eleva...	android android-jetpack-compose android-compos...	60
3	topappbar flash navigate compose navigation	screen scaffold topappbar navigate jetpack nav...	android android-jetpack android-jetpack-compos...	198
4	how create recycler view compose jetpack	special way create.recyclerview compose jetpac...	android android-recyclerview android-jetpack-c...	7
6	how navhostfragment	integrate android navigation architecture comp...	android android-fragments android-architecture...	48

# Évaluation des modèles

## Les métriques d'évaluation:

**Jaccard Score:** mesure la similarité entre les ensembles de prédictions et les ensembles réels.

**Precision:** Le ratio des vrais positifs (VP) par rapport au total des éléments classés comme positifs (vrais positifs + faux positifs).

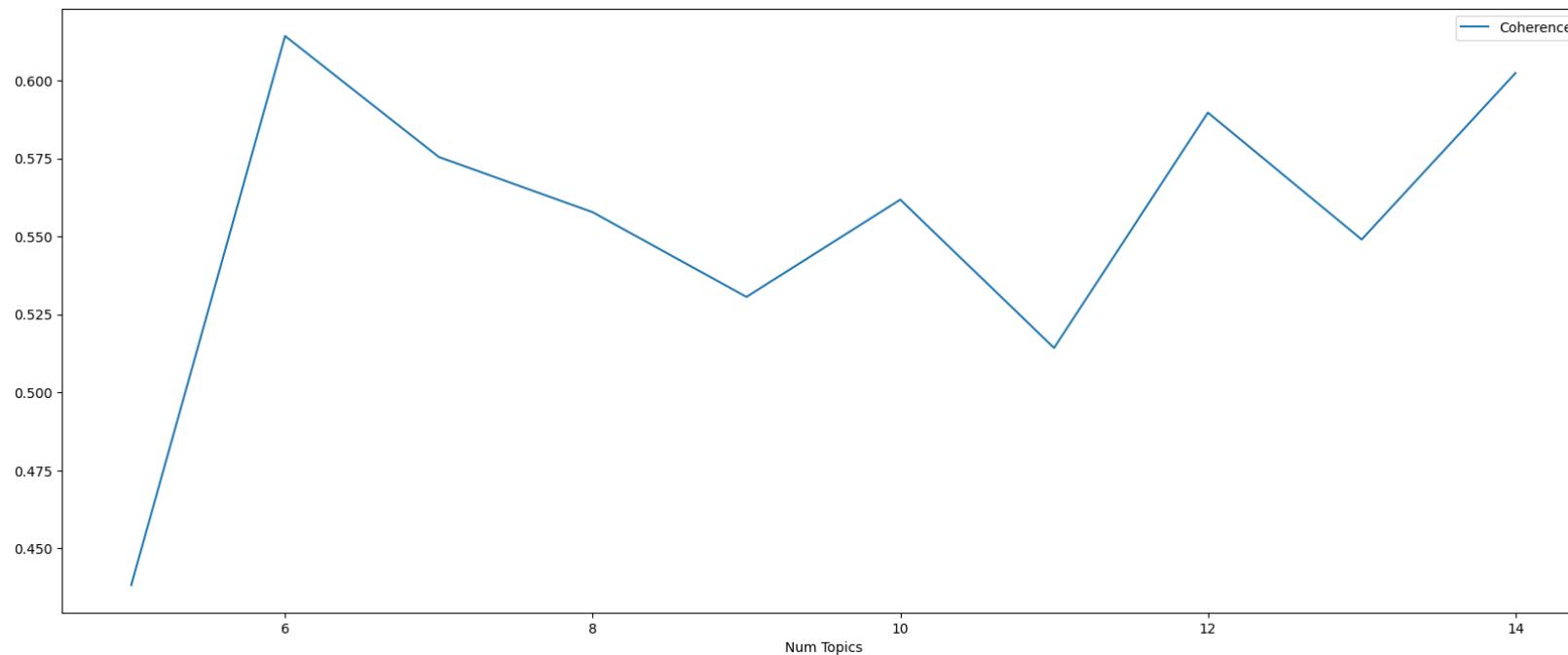
**Recall:** le ratio des vrais positifs par rapport au nombre total d'éléments qui sont réellement positifs (vrais positifs + faux négatifs).

**F1:** la moyenne harmonique de la précision et du recall. Il combine les deux en une seule métrique.

# Approche non supervisée

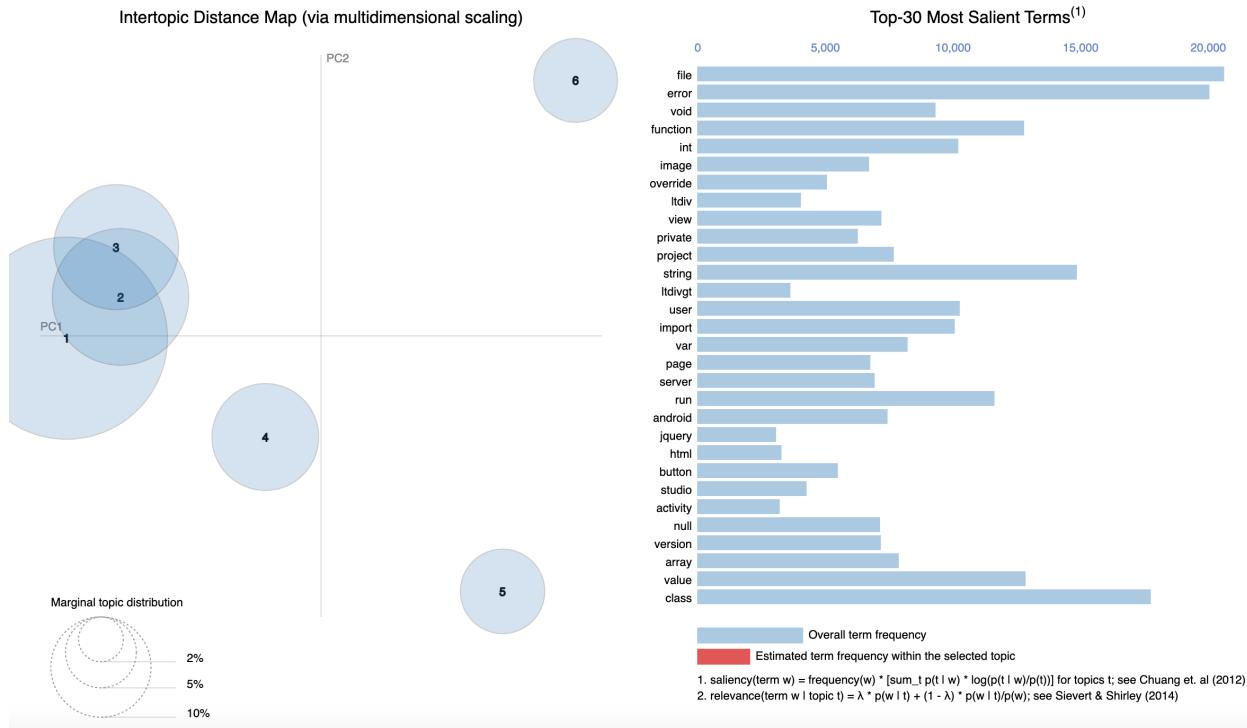
Latent Dirichlet Allocation (LDA) est une technique de modélisation de sujets qui découvre les thèmes cachés dans un ensemble de documents. Chaque document est vu comme une combinaison de sujets, et chaque sujet est une distribution de mots, permettant ainsi de comprendre et de classifier le contenu textuel.

Nombre de topics optimal



# Approche non supervisée

Une régression logistique est ensuite appliquée, pour en extraire les Tags, permettant d'utiliser les métriques adéquates



	Predicted Labels	True Labels
0	(c++, java, python)	(ecmascript-6, functional-programming, javascr...
1	(algorithm, c#, c++, java, python)	(binomial-coefficients, combinations, language...
2	(android, android-studio, python, visual-studio)	(android, android-gradle-plugin, android-studi...
3	(c#, python, sql, sql-server)	(activerecord, autocomplete, model, ruby-on-ra...
4	(javascript,)	(controller, php, symfony, symfony4)
...	...	...
7359	(c#, python, sql, sql-server)	(error-handling, sql-server, sql-server-2005, ...
7360	(android, javascript)	(frameworks, laravel, laravel-artisan, logging...
7361	(javascript,)	(authentication, beautifulsoup, mechanize, pyt...
7362	(android, android-fragments, java)	(android, android-actionbaractivity, android-f...
7363	(android, android-fragments, java)	(android, android-intent, android-layout, andr...

jaccard score: 0.115906

Precision: 0.157495

Recall: 0.284757

F1 Score: 0.186922

Score de cohérence: 0.553056

# Approche supervisée

## Les Différentes vectorisations:

**-TFIDF (Bag of Words):** Transforme les mots en vecteurs en fonction de leur fréquence dans un document et leur rareté dans le corpus, permettant d'évaluer l'importance des mots.

**Word2vec (Word Embedding):** Apprend des représentations de mots en tant que vecteurs en capturant les relations contextuelles entre les mots dans un espace vectoriel continu.

**Bert (Sentence Embedding):** Utilise des modèles de transformer pré-entraînés pour générer des vecteurs de mots contextuels, capables de comprendre les nuances du langage naturel.

**USE (Sentence Embedding):** Produit des vecteurs de phrases qui capturent le sens global des phrases, facilitant les tâches de similarité sémantique et de classification.

Les techniques **Bag of Words**, représentent les mots en fonction de leur fréquence d'apparition dans le texte, sans tenir compte du contexte. **Word Embedding**, capture les relations contextuelles entre les mots en les représentant dans un espace vectoriel continu. En revanche, **Sentence Embedding**, génère des vecteurs pour des phrases entières, capturant le sens global et les nuances contextuelles du texte.

# Approche supervisée

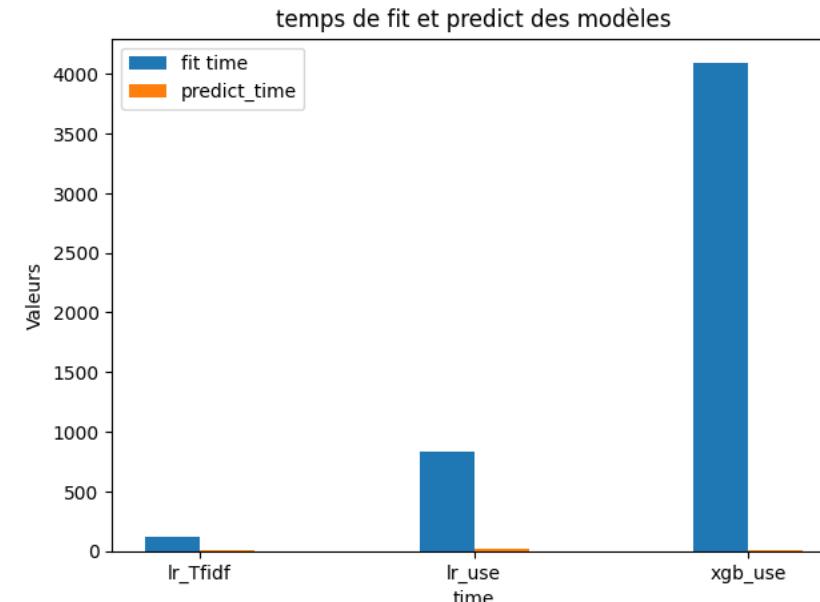
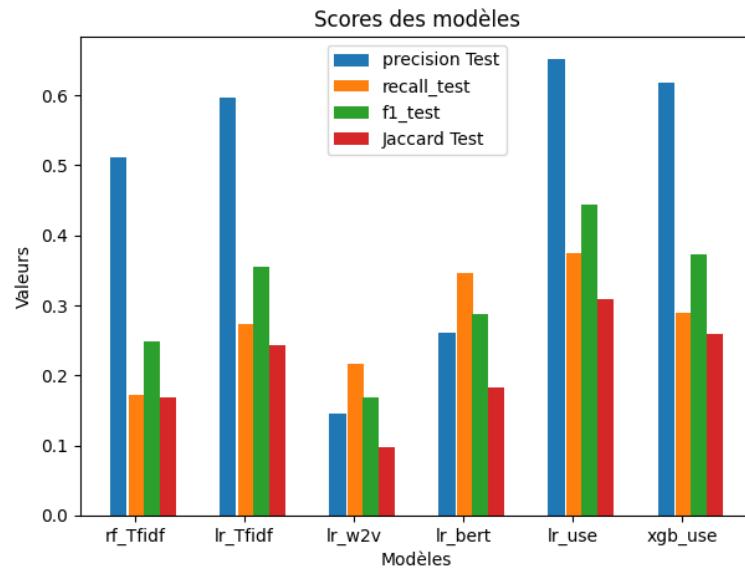
## Les algorithmes testés:

**Régression logistique OneVsRestClassifier :** est une méthode de classification multiclasse. Elle décompose un problème multiclasse en plusieurs problèmes binaires, où un modèle est entraîné pour chaque classe afin de distinguer cette classe de toutes les autres. Les prédictions finales sont faites en sélectionnant la classe avec la probabilité la plus élevée parmi tous les modèles binaires.

**Xgboost:** basé sur les arbres de décision, optimisé pour la performance et l'efficacité. Il est utilisé pour les tâches de classification et de régression, offrant une grande précision et une bonne rapidité d'exécution

**RandomForest:** Algorithme d'ensemble basé sur plusieurs arbres de décision, utilisé pour la classification et la régression. Il améliore la précision et la robustesse en combinant les prédictions de nombreux arbres indépendants, réduisant ainsi le risque de surapprentissage et les erreurs de biais.

# Comparaison des modèles



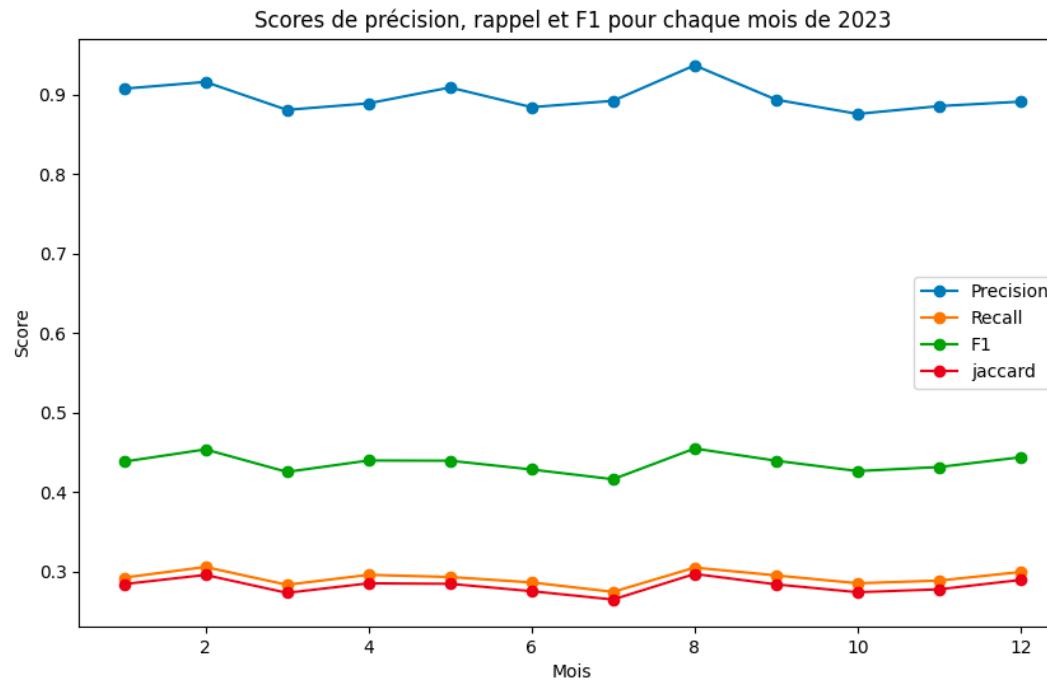
**Performance :** Le modèle lr\_use semble offrir les meilleures performances globales avec des scores élevés sur toutes les métriques.

**Efficacité :** Le modèle lr\_Tfidf est très efficace en termes de temps de fit et de prédiction, bien que ses scores plus faibles que pour lr\_use.

Le temps de fit et de prédiction du modèle lr\_use, bien que moins efficace que le modèle lr\_tfifd, semble néanmoins satisfaisant

**Modèle sélectionné:** Use et Regression Logistique

# Stabilité du modèle



**Nombre de données limité :** Le nombre de données pour chaque mois est réduit, avec une moyenne d'environ 300 instances par mois.

**Impact sur les scores :** Le faible nombre de données peut entraîner une variance élevée dans les scores de rappel et F1, car le modèle dispose de moins d'exemples pour apprendre et généraliser correctement.

Stabilité. On voit que le modèle est relativement stable dans le temps.

# tracking des modèles

Un tracking des différents modèles a été effectué grâce à mlflow, ce qui permet de stocker les informations importantes des différents modèles testés, tel que les différentes métriques, le fit et le predict time, la taille du set de test et de train, les graphiques, et les predictions faites sur le set de test

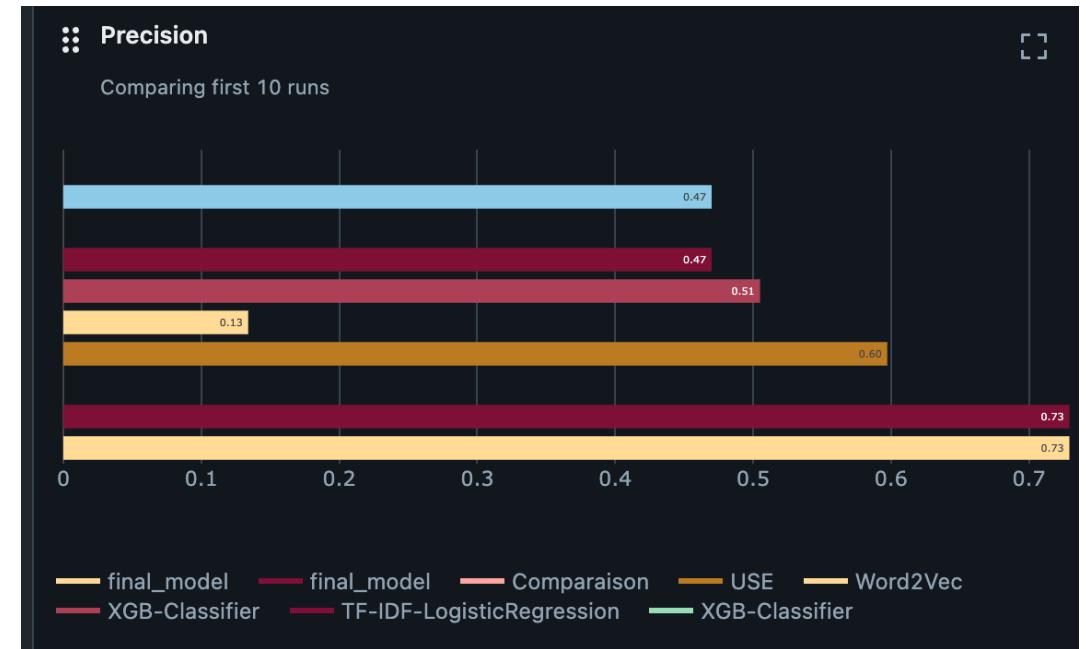
MLflow Quickstart [Provide Feedback](#) [Add Description](#) Share

Q metrics.rmse < 1 and params.model = "tree" Time created State: Active

Datasets Sort: Created Columns Group by + New run

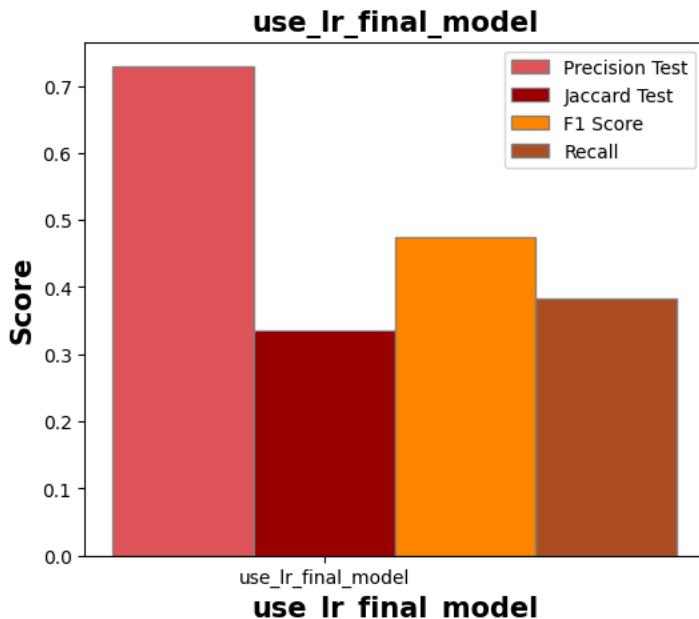
Table Chart Evaluation Experimental

	Run Name	Created	Dataset	Duration	Source
	final_model	10 hours ago	-	8.7min	ipykerne...
	final_model	10 hours ago	-	10.2min	ipykerne...
	Comparaison	10 hours ago	-	451ms	ipykerne...
	USE	10 hours ago	-	20.3s	ipykerne...
	Word2Vec	10 hours ago	-	15.3s	ipykerne...
	XGB-Classifier	10 hours ago	-	1.1min	ipykerne...
	TF-IDF-LogisticRegression	10 hours ago	-	11.9s	ipykerne...
	XGB-Classifier	10 hours ago	-	5.1min	ipykerne...
	TF-IDF-LogisticRegression	10 hours ago	-	10.4s	ipykerne...
	XGB-Classifier	10 hours ago	-	33.3s	ipykerne...
	TF-IDF-LogisticRegression	10 hours ago	-	10.6s	ipykerne...



# tracking des modeles

Metrics (7)	
Metric	Value
Jaccard Score	0.33491993438693607
fit_time	196.23536086082458
vec_time	192.41854310035706
F1 Score	0.47398856650621995
Recall	0.38385614702154625
Precision	0.7293457877447557
prediction_time	15.733880758285522



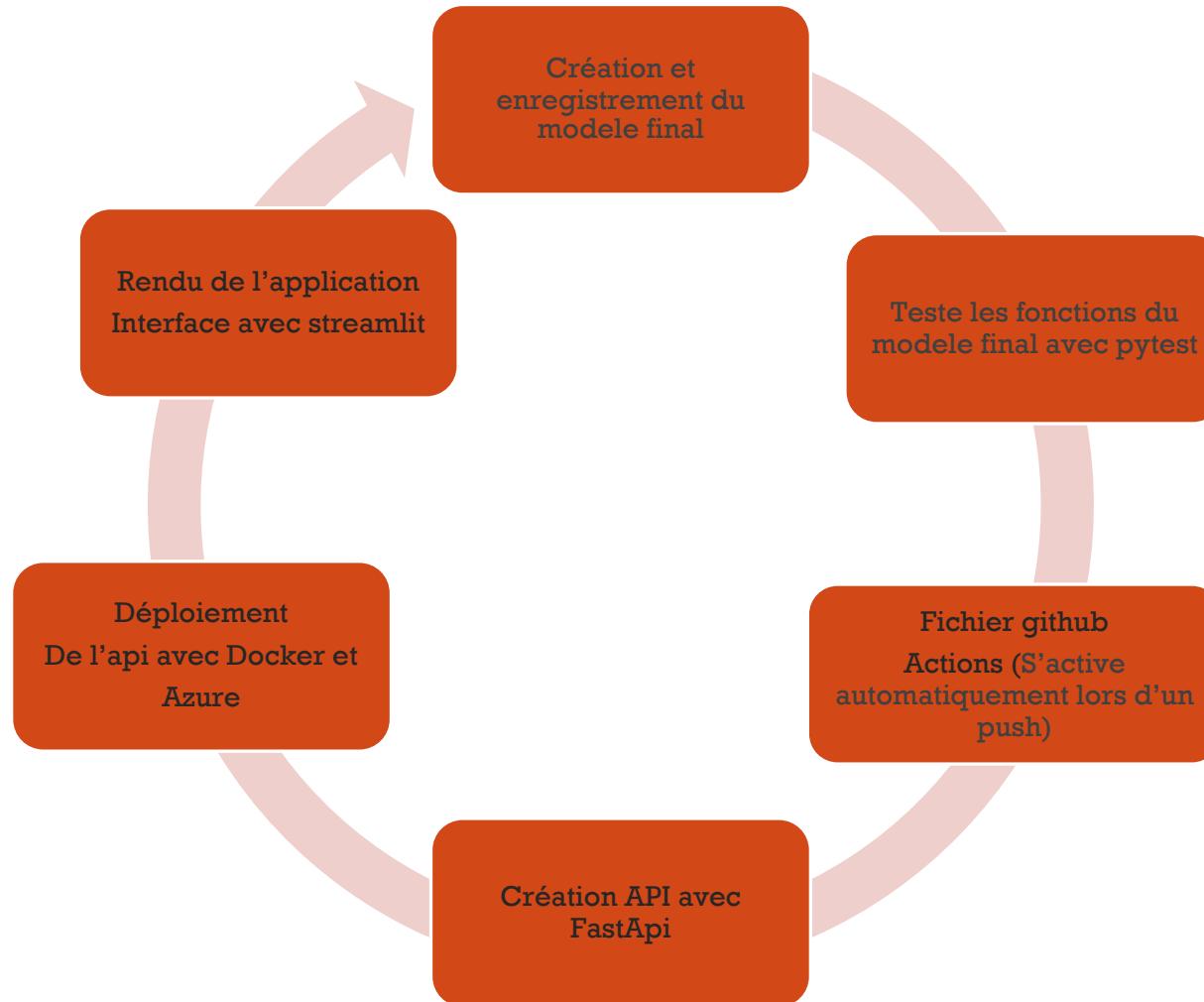
model

- metadata
- MLmodel
- conda.yaml
- model.pkl
- python\_env.yaml
- requirements.txt
- predictions\_labels.csv** 182.39KB

Path: file:///Users/bahia/Desktop/MLE-P5-main2/mlruns/656711072440611296/668bc5d826494860b64e300b76a590c9/artifacts/predictions\_labels.

0	1	2	3	4
java				
android	concurrency	java	multithreading	synchronization
javascript	jquery			
javascript	jquery	twitter-bootstrap		
json				
ios	iphone	objective-c		
java				
google-chrome	html	javascript	jquery	

# Création API et déploiement



# Github

l'historique des commits GitHub, illustrant les différentes étapes tout au long du développement du projet. Chaque commit représente une mise à jour importante ou une amélioration du code, assurant une traçabilité et une gestion efficace des versions. Lien vers le dossier : <https://github.com/BahiaB/MLE-P5-SOF>

Commits

main All users All time

Commits on Jul 4, 2024

Commit	Author	Date	Changes	Hash	Actions
Entry_point	BahiaB	15 hours ago	1 / 1	2ff2d36	Copy Diff
Merge remote-tracking branch 'MLE-P5_SOF/main'	BahiaB	15 hours ago	1 / 1	02957fa	Copy Diff
Delete wrong file	BahiaB	15 hours ago		5c4f11e	Copy Diff
Delete unsupervised2.ipynb	BahiaB	15 hours ago	1 / 1	25abd41	Copy Diff Verified
Merge remote-tracking branch 'MLE-P5_SOF/main'	BahiaB	15 hours ago	1 / 1	1f78af7	Copy Diff
LogisticRegression in unsupervised	BahiaB	15 hours ago		ac7540a	Copy Diff
Delete unsupervised.ipynb	BahiaB	15 hours ago	1 / 1	58fd283	Copy Diff Verified
commit new repo				b453287	Copy Diff

# Github Actions

l'utilisation de GitHub Actions pour l'intégration et le déploiement continu (CI/CD) de notre application. Grâce à un workflow automatisé, chaque push déclenche une série de tâches, incluant l'installation des dépendances, l'exécution des tests

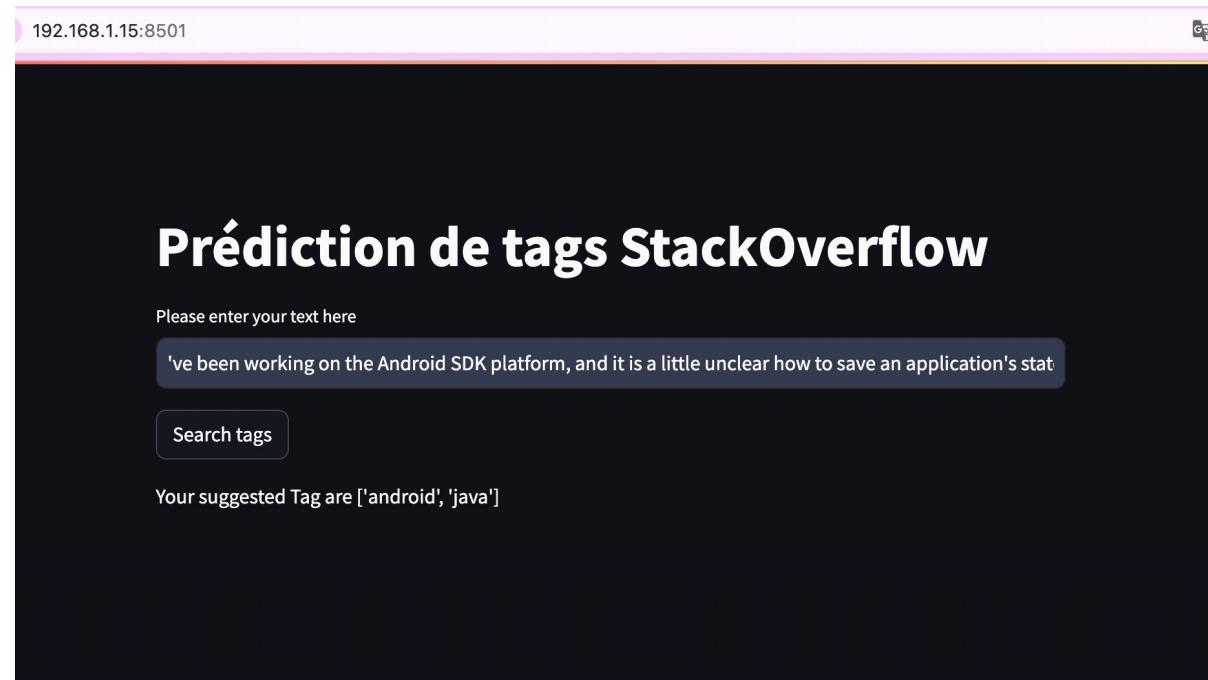
The screenshot shows a GitHub Actions build log for a workflow named 'build'. The build was successful, completed 9 hours ago in 9m 0s. The log details the execution of various steps:

Step	Description	Duration
> ✓ Set up job		1s
> ✓ Run actions/checkout@v4		7s
> ✓ Set up Python version		0s
> ✓ Create and start virtual environment		3s
> ✓ Install dependencies		2m 46s
> ✓ download nltk		2s
> ✓ pytest		5m 59s
> ✓ Post Run actions/checkout@v4		0s
> ✓ Complete job		0s

# Création de l'API

L'API a été créée grâce à FastAPI, un framework web moderne qui permet de développer rapidement des APIs performantes avec Python. Les performances sont optimisées pour répondre rapidement aux requêtes.

## Interface en local



# Déploiement dans Azure

- Crédit d'une image et d'un container avec docker
- Loger le container dans le groupe de ressource Azure
- Créer une instance de ce conteneur dans Azure et la déployer

The screenshot shows the Microsoft Azure portal interface. At the top, there is a navigation bar with the Microsoft Azure logo, a search bar, and various icons. On the right side of the header, the user's email (bahia.benali@gmail.com) and profile picture are displayed.

The main content area displays a container instance named "test2". The left sidebar shows navigation options: Accueil >, Instances de conteneur, Vue d'ensemble (which is selected), Journal d'activité, Contrôle d'accès (IAM), Étiquettes, Paramètres, Supervision, Automatisation, and Aide.

The central pane shows the following details for the container instance:

- Groupe de res... ([déplacer](#)) : Final\_ps
- Statut : En cours d'exécution
- Emplacement : West Europe
- Abonnement ([déplacer](#)) : Azure subscription 1
- ID d'abonnement : 2026f883-5731-4763-8e81-383410d32f29
- Référence SKU : Standard
- Type de système d'exploit... : Linux
- Adresse IP (Public) : 20.50.169.83
- FQDN : ---
- Nombre de conteneurs : 1

Below these details, there are two performance monitoring charts:

- Processeur**: A line chart showing CPU usage over time. The Y-axis ranges from 0 to 4, and the X-axis shows times 11:15, 11:30, 11:45, and UTC+02:00. The chart shows a sharp spike around 11:45.
- Mémoire**: A line chart showing memory usage over time. The Y-axis ranges from 0o to 1,8Go, and the X-axis shows the same time points. The chart shows a steady increase in memory usage over time.

# Résultat du déploiement

URL: <http://20.50.169.83/docs>

Responses

Curl

```
curl -X 'POST' \
  'http://20.93.223.113/predict/' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
    "text": "I've been working on the Android SDK platform, and it is a little unclear how to save an application's state. So given this minor re-tooling of the \"Hello, Android\"'
}'
```

Request URL

```
http://20.93.223.113/predict/
```

Server response

Code	Details
200	<p>Response body</p> <pre>{   "prediction": [     [       "android",       "java"     ]   ] }</pre> <p>Download</p> <p>Response headers</p> <pre>content-length: 35 content-type: application/json date: Sun, 30 Jun 2024 12:03:59 GMT server: uvicorn</pre>

# Conclusion

Ce projet a permis de développer une solution complète pour la prédiction de tags sur StackOverflow, en passant par la création, l'entraînement et l'évaluation des modèles, jusqu'à la mise en place d'une API et son déploiement sur Azure. Nous avons utilisé diverses techniques de vectorisation et des modèles de machine learning avancés pour atteindre des performances optimales. Grâce à des outils comme FastAPI, Streamlit, et GitHub Actions, nous avons assuré une intégration et un déploiement continu efficace.

Axes d'Amélioration:

- Suivi du Model Drift
- Optimisation des Hyperparamètres