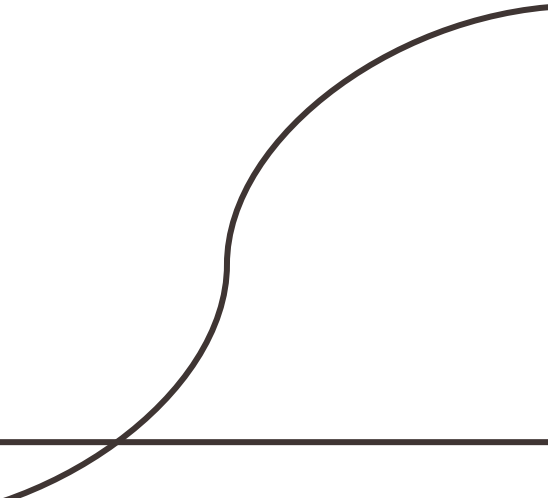


---

# Catégorisez automatiquement des questions

Soutenance de projet

Nicolas FAUCONNIER  
Parcours Ingénieur ML  
17/06/2022

A decorative curved line in a dark purple color, starting from the bottom right and curving upwards and to the left, ending near the center of the bottom edge of the slide.

# Plan

1.

## Problématique

Projet et missions

2.

## Data Cleaning

Données et prétraitements effectués

3.

## Analyse exploratoire

Analyse des données quantitatives et des questions

4.

## LDA

Approche non supervisée et interprétation des résultats

5.

## Modélisation supervisée

Extraction de features et entraînement des modèles

6.

## API

Démonstration de l'API REST



1.

# Problématique

Projet et missions

# 1. Problématique

## Contexte

StackOverflow est un site de questions-réponses liées au développement informatique. Chaque auteur d'une question lui attribue des « tags » afin qu'elle soit facilement retrouvable et mieux référencée. Trouver les bons tags pour une question n'est pas une tâche intuitive, notamment pour les utilisateurs novices.

## Missions

- Créer un système de suggestion de tags pour les nouvelles questions StackOverflow
- Rendre accessible ces suggestions via une API REST
- Analyser les données utilisées

# 2.

# Data Cleaning

Données et prétraitements effectués

# Dataset



Le dataset est la concaténation des résultats de deux requêtes SQL effectuées sur l'outil StackExchange Data Explorer. Il s'agit des questions StackOverflow ayant:

- ayant été mises en **favoris par au moins 6 utilisateurs**
- ayant obtenues un **score d'au moins 7**
- ayant reçu **au moins une réponse**
- ayant **au moins 5 tags**

2 .csv pour un total de 89Mo

Outils utilisés:



# Preprocessings

- Concaténation Titre et Body
- Retrait des cellules de code
- Retrait des balises html
- Retrait des accents
- Retrait des caractères spéciaux
- Expansion des formes contractées
- Lemmatization
- Retrait des stopwords
- Tokenization

```
# Check du texte preprocessed
df["cleaned_Text"][27024]
```

```
'database remove uninstall android applicationi two major question database delete uninstall
app downloaded file delete unstable app database android application create java specific pat
h define code database code download file specific path allow create folder android datum com
myapp well please help sorry bad english'
```

```
# Check Titre de la question
df["Title"][27024]

'Database won't remove when uninstall the Android Application'

# Check corps de la question
print(df["Body"][27024])

<p>I have two major questions. </p>

<ol>
<li>Database won't delete when uninstall app.</li>
<li>Downloaded files won't delete while unstable the app.</li>
</ol>

<p>There is a database in my android application. I create it by java</p>

<pre><code>class as follows.

public DataBaseHelper(Context context) {
    super(context, DATABASE_NAME, null, DATABASE_VERSION);
}

public DataBaseHelper(Context context, String name, SQLiteDatabase.CursorFactory factory, int version, DatabaseErrorHandler errorHandler) {
    super(context, name, factory, version, errorHandler);
}

@Override
public void onCreate(SQLiteDatabase db) {
    // creating required tables
    db.execSQL(CREATE_TABLE_QUOTES);
    db.execSQL(CREATE_TABLE_FILTERS);
}

@Override
public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
    // on upgrade drop older tables
    db.execSQL("DROP TABLE IF EXISTS " + TABLE_QUOTES);
    db.execSQL("DROP TABLE IF EXISTS " + TABLE_FILTERS);
    // create new tables
    onCreate(db);
}
}</code></pre>

<p>There is no specific path defined at the code for database.</p>

<p>This is the code how I download files. And there is specific path, But it is not allowed to create folder in Android>data>com.myapp as well

<pre><code>public String downloadImage(String img_url, int i) {
    File sdCard = Environment.getExternalStorageDirectory();
    File dir = new File (sdCard.getAbsolutePath() + "/fog/Images/filters");
    // Make sure the Pictures directory exists.
    dir.mkdirs();
    File destinationFile = new File(dir, "filter"+i);
    String filepath = null;
    try{
        URL url = new URL("http://fog.wrapper.io/uploads/category/"+img_url+".png");
```



3.

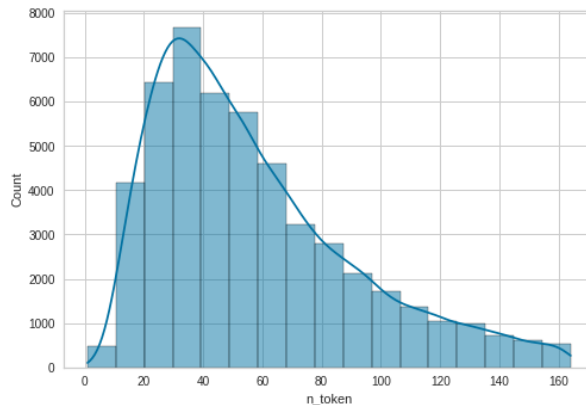
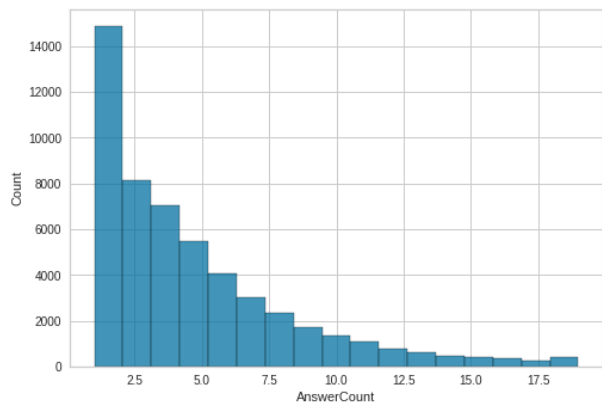
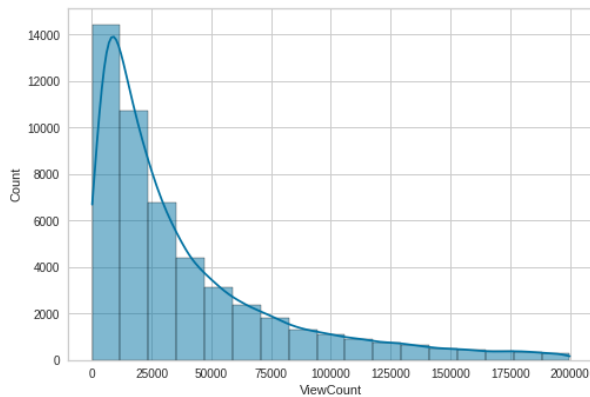
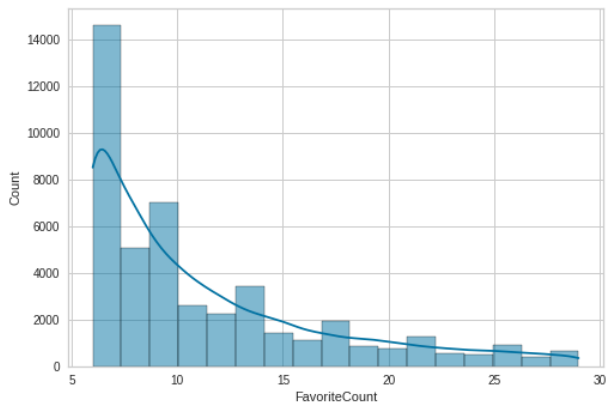
# Analyse exploratoire

Analyse des données quantitatives et des questions





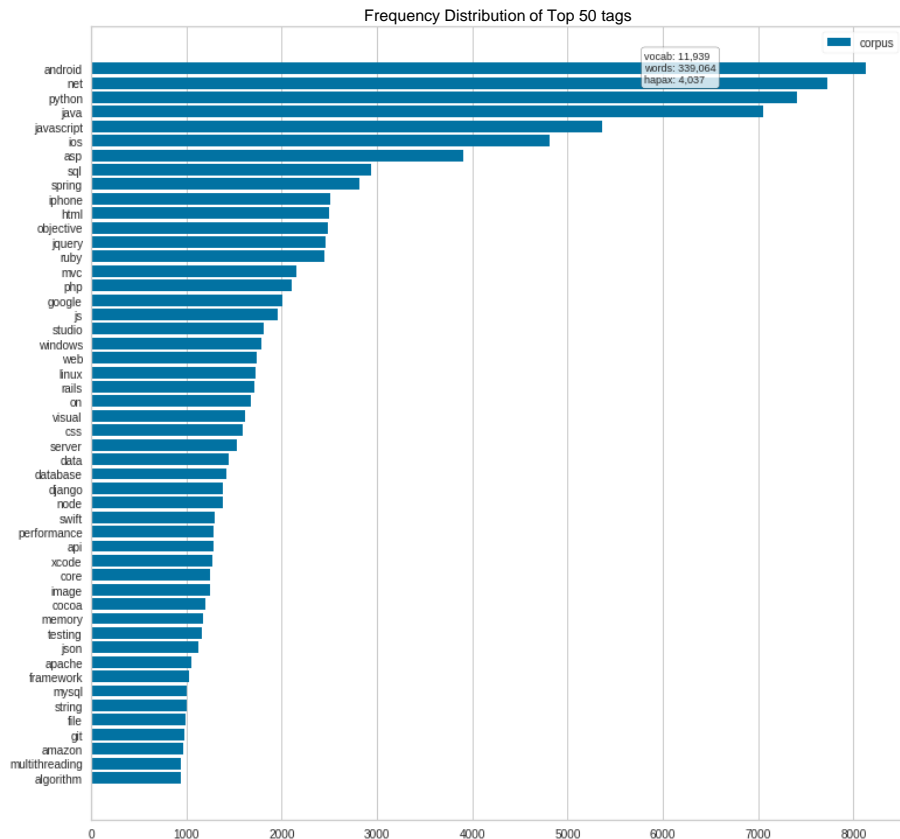
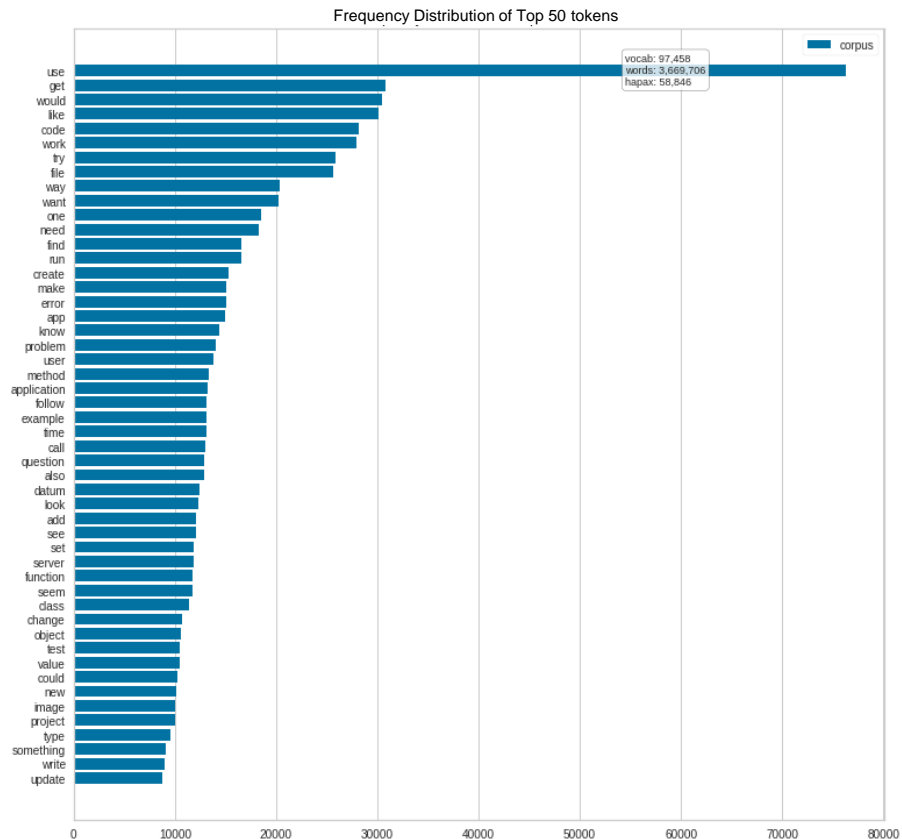
# Variables annexes



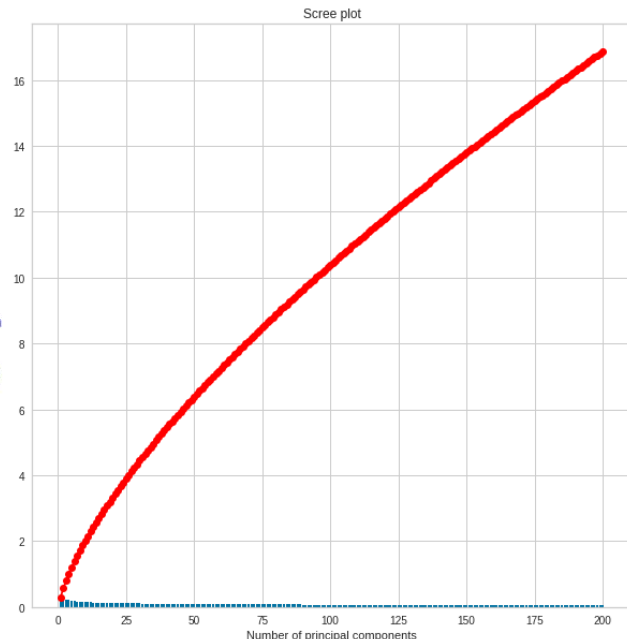
	ViewCount	FavoriteCount	AnswerCount	n_token
<b>count</b>	53750	53750	53750	53750
<b>mean</b>	67231	25	5	69
<b>std</b>	175777	109	5	63
<b>min</b>	141	6	1	1
<b>25%</b>	11055	7	2	33
<b>50%</b>	26280	11	4	52
<b>75%</b>	62908	19	7	84
<b>max</b>	9771619	11552	125	1643

- Distribution log-normales
- Corrélations positives (0,3 à 0,5) sauf pour le nombre de tokens

# Distribution des tags et des tokens

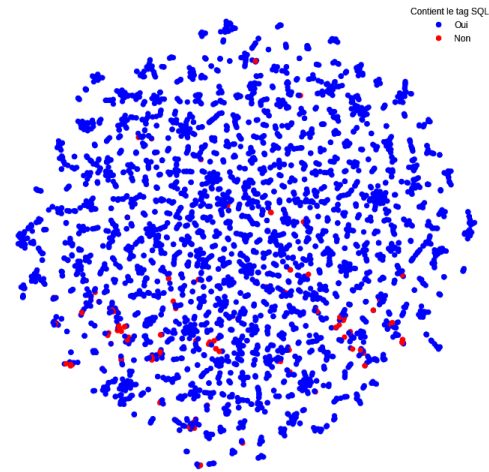


ACP:



- **Aucune des deux méthode n'offre de résultats satisfaisants.**  
(nb. overlapping des tags et sujets)

t-SNE sur échantillon de 4000 commentaires StackOverflow



# 4. LDA

Approche non supervisée et interprétation des résultats

---

# Latent Dirichlet Allocation

Effectuée sur 6600 questions

CountVectorizer, puis TF-IDF (inclus dans *sklearn.LatentDirichletAllocation*)

GridSearch, et surtout expérimentation manuelle (interprétabilité) pour trouver le bon nombre de topics

Librairie pyLDavis pour les visualisation, et Gensim pour le calcul du score de « Topic Coherence » :

```
print("Coherence Model - LDA: ", coherence_lda)
```

```
Coherence Model - LDA: 0.31660799608139767
```

- 1: Système d'exploitation, bash/shell, paths, versions, etc
- 2: algorithmes, classes et Programmation Orientée Objets
- 3: API
- 4: strings/textes
- 5: Java
- 6: JavaScript et développement web
- 7: développement mobile (Apple/iOs)
- 8: SQL et bases de données, par exemple:

```
<p>I have a WCF service from which I want to return a DataTable. I know that this is often a highly-debated topic.
```

```
<p>When I create a DataTable from scratch, as below, there are no problems whatsoever. The table is created, popu.
```

```
<pre><code>[DataContract]
public DataTable GetTbl()
{
    DataTable tbl = new DataTable("testTbl");
    for(int i=0;i<100;i++)
    {
        tbl.Columns.Add(i);
        tbl.Rows.Add(new string[]{"testValue"});
    }
    return tbl;
}
</code></pre>
```

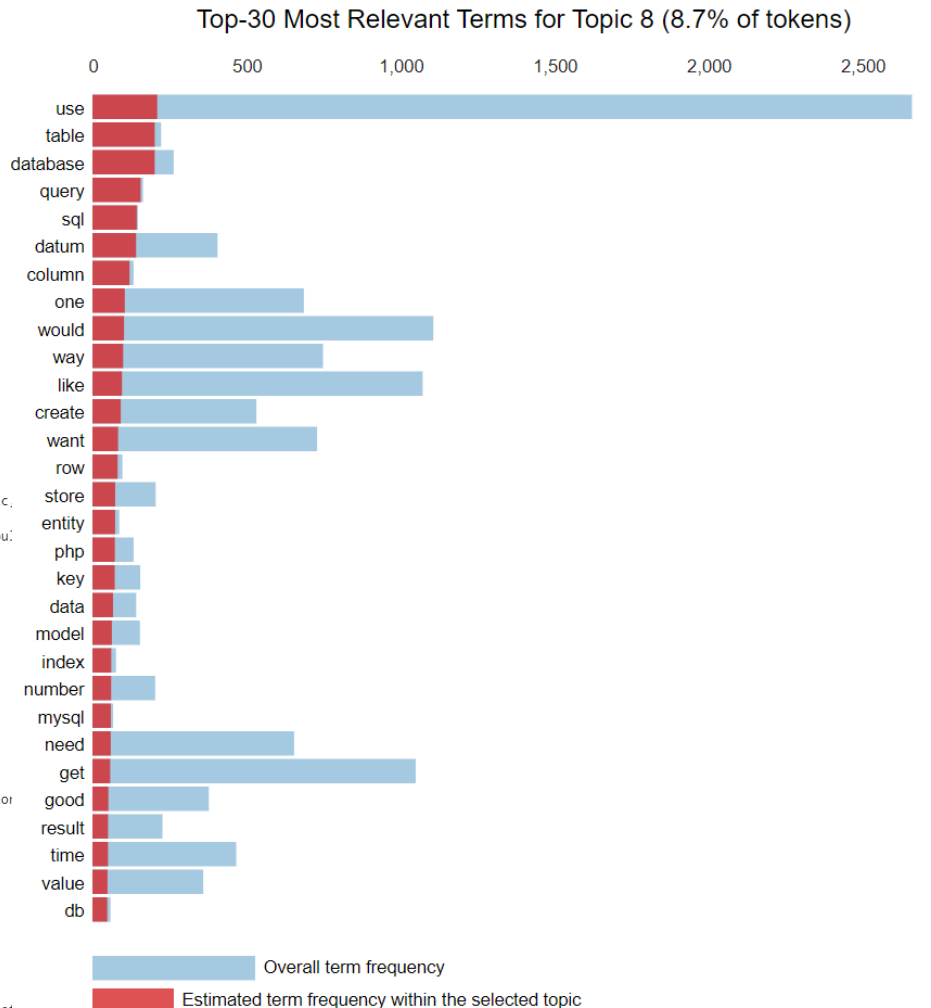
```
<p>However, as soon as I go out and hit the database to create the table, as below, I get a CommunicationException
```

```
<pre><code>[DataContract]
public DataTable GetTbl()
{
    DataTable tbl = new DataTable("testTbl");
    //Populate table with SQL query

    return tbl;
}
</code></pre>
```

```
<p>The table is being populated correctly on the server side. It is significantly smaller than the test table that
```

```
<p>Why would the way that the table is being populated have any bearing on the table returning successfully?</p>
```



1. saliency(term w) = frequency(w) \* [sum\_t p(t | w) \* log(p(t | w)/p(t))]] for topics t; see Chuang et. al (2012)
2. relevance(term w | topic t) =  $\lambda * p(w | t) + (1 - \lambda) * p(w | t)/p(w)$ ; see Sievert & Shirley (2014)



# 5.

# Modélisation supervisée

Extraction de features et entraînement des modèles

Classification MultiLabels des 50 tags les plus communs

# Extractions de feature

TF-IDF	Word2Vec	Bert	USE
<code>max_features=2000</code>	<code>w2v_size=300</code> <code>w2v_window=5</code> <code>w2v_min_count=1</code> <code>w2v_epochs=50</code> <code>maxlen = 150</code>	<code>max_length = 150</code> <code>batch_size = 25</code>  <u>Small Bert:</u> -4 Layers (L) -128 Hidden Layer size (H) -2 Attention Heads (A)	<code>batch_size = 10</code>
2000 dimensions	300 dimensions	128 dimensions	512 dimensions

<https://radimrehurek.com/gensim/models/word2vec.html>

[https://tfhub.dev/tensorflow/small\\_bert/bert\\_en\\_uncased\\_L-4\\_H-128\\_A-2/2](https://tfhub.dev/tensorflow/small_bert/bert_en_uncased_L-4_H-128_A-2/2)

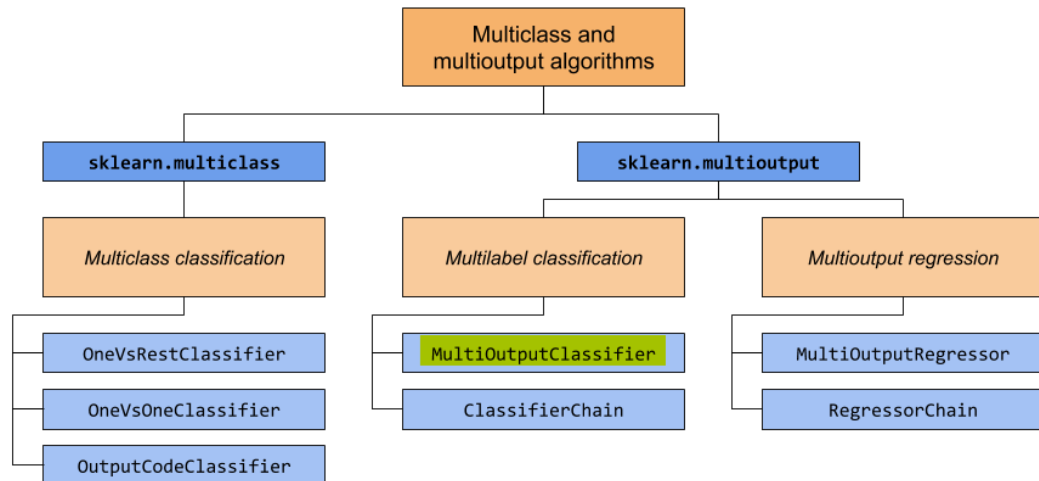
<https://tfhub.dev/google/universal-sentence-encoder/4>



# Modélisation

Test set: 1/3 du dataset

Modélisation: **MultiOutputClassifier**  
avec **Régression Logistique** sur les  
4 extractions de feature



On retient TF-IDF:

- Performances relativement bonnes
- Moins de ressources nécessaires (RAM limitée sur Heroku pour le déploiement de l'API)

	Accuracy	Score Jaccard	Hamming Loss
TF-IDF	0.2936	0.3063	0.0223
Word2Vec	0.2778	0.2913	0.0236
Bert	0.1386	0.0830	0.0298
USE	0.3326	0.3546	0.0205

# 6. API

Démonstration de l'API REST

---

# API REST

Développée avec FastAPI

Déploiement sur une instance Heroku

Repository :

[https://github.com/fauconnier-n/stackoverflow\\_app](https://github.com/fauconnier-n/stackoverflow_app)

2 endpoints (POST) :

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

Swagger (UI & doc) :

<https://stackoverflow-tags-pred.herokuapp.com/docs>

---

---

# Pistes d'amélioration

- Entraînement sur plus d'observations (besoin de plus de ressources)
  - Entraîner des modèles plus complexes
  - Traiter et exploiter les cellules de code (eg. extraire les commentaires)
  - Regrouper certains tags (eg. sql et mysql, langages et leurs librairies)
-

---

# Merci

fauconnier.n@gmail.com

---