## README :

### 1) Instructions pour répliquer les résultats

Pour reproduire les résultats, il est important d'exécuter les notebooks dans l'ordre indiqué dans leurs titres. Avant de les exécuter**, il faut lire attentivement les notebooks 9 et 10, que j'ai déjà exécutés sur Kaggle**, car ils utilisent un LLM pour la génération. Il y a quelques manipulations spécifiques à respecter :

- Télécharger correctement les fichiers CSV d'entrée dans Kaggle.
- La génération de résumés étant chronophage, le Notebook 9 a été exécuté en plusieurs fois pour annoter quelques lignes de mon CSV. Les manipulations à faire pour exécuter le notebook en plusieurs fois sont détaillées dedans.

Une fois les notebooks exécutés, il faut créer une base de données avec PostgreSQL et y mettre les données procedures_2025.csv, puis la lier avec Flask dans le fichier app.py, qui crée le site. Si on exécute app.py dans le terminal (en lançant la commande flask run), on peut ouvrir le site en local. Pour la publication, j'ai rendu le site totalement statique avec le fichier freeze_website.py (le rendre statique le rends gratuit à la publication avec Github, puisqu'on n'a pas besoin de passer par Render).

### 2) Quel fichier correspond à quoi ?

**Dossier docs :** comprend les fichiers utilisés par Github pour publier mon site
**Dossier static :** comprend les fichiers permettant le style de mon site (js, css, images)
**Dossier templates :** contient les codes pour les pages html de mon site

**Notebooks (les flèches indiquent quels CSV sont produits par l'exécution du Notebook) :**

1_url_scraper : scrape les urls des procédures législatives répertoriées dans l'Observatoire législatif.

- list_urls_2025.csv

2_scraper : scrape les informations directement disponibles sur une page décrivant une procédure législative ordinaire

- urls_not_found_2025.csv
- urls_no_title_2025.csv
- urls_missing_title_span_2025.csv
- final_scrape_2025.csv

3_data_manip_technical_info : manipulation de données sur la colonne technical_info pour en extraire les données et les stocker dans de nouvelles variables

- final_sample_cod_manip_2025.csv

4_legislative_proposal_scraper : scrape les textes des propositions législatives

- lp_url_not_found_2025.csv
- final_sample_cod_legislative_scrape_2025.csv

5_lp_scraped_verifier : identifie quelles procédures n'ont pas permis de scraper le texte de la proposition législative

6_text_adopted_scraper : scrape les textes adoptés des procédures terminées

- final_sample_cod_TA_2025.csv

7_final_act_scraper : scrape les actes finaux des procédures terminées

- final_sample_cod_TA_final_act_2025.csv

8_splitting_datasets : sépare la base de données en 4 bases de données qui regroupent chacune les procédures relavant d'une de ces catégories : procédures complétées, procédures rejetées, procédures en cours, procédures abandonnées ou retirées

- sample_cod_completed_2025.csv
- sample_cod_rejected_2025.csv
- sample_cod_lapsed_2025.csv
- sample_cod_ongoing_2025.csv

9_summarization_proposals : résume quelques propositions législatives d'avril à décembre 2025

- cod_completed_proposal_general_summary_0_5.csv
- cod_completed_proposal_general_summary_5_10.csv
- cod_completed_proposal_general_summary_10_15.csv
- cod_completed_proposal_general_summary_15_17.csv
- cod_proposal_general_summary_0_5.csv
- cod_proposal_general_summary_5_10.csv
- cod_proposal_general_summary_10_20.csv
- cod_completed_proposal_general_summary_20_25.csv

10_summarization_final_acts : résume les actes finaux adoptés par le Parlement Européen en 2025

- cod_final_act_general_summary.csv

11_final_manipulation_for_sql : agrège les procédures en cours et complétées et les nettoie pour les importer sur PostgreSQL

- completed_proc_clean_2025.csv
- ongoing_proc_clean_2025.csv
- procedures_2025.csv