**République du Sénégal**



*Un peuple-Un but-Une foi*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Ministère de l’économie du plan et de la coopération

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Agence Nationale de la Statistique et de la Démographie (ANSD)



\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Ecole Nationale de la Statistique et de l’Analyse Economique-Pierre NDIAYE



\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Web Scraping :**

**Résumé de projet**

**Rédigé par :**

Ahmed Firhoun OUMAROU SOULEYE

Mamadou Saïdou DIALLO

*Elèves en AS3*

**Juin 2025**

**Sous la supervision de :**

1. Baye Demba DIACK

*Chef du Bureau des Données et des Solutions informatique (BDSI)*

# Introduction

Ce projet a été réalisé dans le cadre du cours de Webscraping de la classe AS3 – Option Data Science. L’objectif est de collecter des données de sites d’actualités afin d’analyser les tendances médiatiques. Les sites SeneWeb et Senego seront retenus pour la collecte et nous utiliserons le Natural Language Processing (NLP) et la modélisation de sujets avec Latent Dirichlet Allocation (LDA) pour détecter les mots-clés fréquents et les tendances thématiques. Cela ofrira une vue d’ensemble des priorités médiatiques au Sénégal.

# Méthodologie

## Collecte des données

Un script *scraper.py*, basé sur BeautifulSoup, scrape quotidiennement SeneWeb et Senego, et harmonisant les rubriques (ex. "Politique", "Economie", "Sport", …), les catégories de chaque site pouvant différer, il est nécessaire d’effectuer une harmonisation. L’automatisation est assurée par le workflow GitHub Action, le fichier *scrape.yml*, exécuté de manière journalière collecte les informations sur les différents sites, notamment la date de publication, l’auteur, le titre, le contenu et l’URL. Les données sont continuellement sauvegardées dans *articles\_scraped.csv* sur le repository GitHub.

## Préparation et analyse des données

Les contenus sont prétraités avec la bibliothèque NLTK, les stopwords ou mots-vides (termes comme "le", "la", "un","selon", "est"… ) sont retirés, et les articles sont vectorisés avec CountVectorizer. Nous avons ensuite pu passer à la modélisation et à l’optimisation des hyperparamètres, qui a été réalisée avec la bibliothèque Optuna dans le fichier *lda.py*. Le modèle a été ainsi entraîné et réentraîné de manière hebdomadaire via *lda.yml* pour intégrer les nouvelles données. Le modèles et le vectorizer sont sauvegardés dans *models/*.

## Développement de l’application

L’application Dash (*app.py*) permet de visualiser les résultats avec des graphiques dynamiques et filtres par source, thème ou date. Elle récupère directement le modèle et la base de données depuis le repository GitHub. L’interface est fonctionnelle et déployée via *Render.com.*

Elle est composée de trois pages en dehors de la page d’accueil :

* **Analyse exploratoire :** Page présentant quelques statistiques descriptives sur les articles et la fréquence de publication ;
* **Recherche d’article :** Page permettant de rechercher un article selon son titre, sa catégorie ou la date de publication et d’y accéder ;
* **Topic modeling :** Présentant les résultats du NLP (nuage des mots, diagramme des mots les plus fréquents) ainsi que les résultats du topic modeling par LDA.

## Conclusion

Ce projet illustre l’efficacité du scraping automatisé, du NLP, et de Dash pour analyser les tendances médiatiques. Les livrables incluent ce résumé, les codes Python et YAML (scraper.py, lda.py,scraper.yaml, lda.yaml, Notebook\_NLP.ipynb), le document d’architecture, les liens renvoyant à l’application Dash et au répertoire GitHub.

## Liens :

**Application Dash déployée sur Render :** [**https://sene-scraper.onrender.com/**](https://sene-scraper.onrender.com/)

**Code source de l’application Dash :** [**https://github.com/zimbo-hur/app\_sene\_scraper**](https://github.com/zimbo-hur/app_sene_scraper)

**Code source pour le Web Scraping et le Topic Modeling :** [**https://github.com/zimbo-hur/sene-scraper**](https://github.com/zimbo-hur/sene-scraper)