# Rapport sur le projet final

L’application “Presse-Sentiment” récupère depuis le web les articles de presse en français.

Après un traitement analysant les textes, toutes ces données sont enregistrées dans une base de données.

Un site internet permet de consulter ces données sous différentes formes (graphiques, statistiques, positivité, articles similaires, nuage de mots, ...).

## Démarche

Scraping

Le script en charge du scraping récupère les articles de [Google Actu](https://news.google.com/u/1/topstories?hl=fr&gl=FR&ceid=FR%3Afr), en extrait le titre, l’éditeur, le lien vers l’article, ...

Puis scrape le lien de chaque article pour récupérer le texte complet des articles en question.

Sur la base des infos textuelles, (titre + chapeau + texte), différents traitements sont réalisé :

* Concaténation pour obtenir une chaîne de caractère par article
* Calcul de la positivité
* Calcul de la subjectivité
* Tokenistion, Stemmatisation et Lemmatisation
* Nettoyage du texte à l’aide d’expressions régulières
* Création d’une matrice de termes

L’exécution du script est automatisée et se lance régulièrement.

Les logs sont enregistrés dans un fichier.

Base de **données**

L’article ainsi traité est comparé à la base de données existante pour éviter les doublons.

Puis il est enregistré dans une base de données non-relationnelle (Mongodb).

Application Flask

L’application web consulte la base de données pour retourner les données utiles à divers traitements :

- **Graphiques**

Ils représentent l’évolution de la moyenne des notes de positivité des articles à travers le temps.

- **Classement par positivité**

Liste les articles d’une période choisie et les affiches par ordre de positivité.

- **Présentation des articles similaires**

Sur la base d’une matrice [TF-IDF](https://fr.wikipedia.org/wiki/TF-IDF), renvoie les articles les plus semblables à celui choisi.

- **Nuage de mots**

Affiche les termes les plus utilisés d’une période choisie sous la forme d’un nuage de mots.

- **Statistiques**

Présente quelques statistique générales sur les données.

Docker

Le tout s’exécute en conteneur depuis Docker-Compose.

## Notions abordées

* Scraping avec **BeatifulSoup**
* Base de données non-relationnelle avec **Mongodb** (Écriture, Lecture et modifications)
* Écriture de logs dans un fichier
* Web app avec **Flask**, **HTML** et **CSS**
* Rafraîchissement des images malgré le cache du navigateur

- (En créant un code aléatoire concaténé au nom)

* Programmation Orienté Objet
* Import de packages personnels avec le module **sys** (Sys.path)
* Exécution de script automatisée avec **Crontab**
* Conteneurisation avec **Docker** et **Docker-Compose**
* Import grâce au fichier **requirements.txt**
* Nettoyage de texte à l’aide d’**expressions régulières**
* Calcul de positivité et de subjectivité avec **Textblob**
* Lemmatisation et Stemmatisation avec **SpaCy**
* Matrice de termes avec **CountVectorizer** de **Scikit Learn**
* Matrice TF-IDF avec **TfidfVectorizer**
* Extraction de noms propres avec **SpaCy**
* Nuage de mots avec **WordCloud**
* Manipulation de DataFrame avec **Pandas**
* Création de graphiques avec **Matplotlib**
* Utilisation des dates avec **Time** et **Datetime**
* Manipulation de fichier avec **OS** et **Glob**
* Traitement séquentiel pour effectuer des modifications assez lourdes sur la base de données

## Remarques

Pour simplifier la démarche, j’ai pris le parti de scraper tous les articles depuis Google Actu.

* Problème d’exhaustivité (l’algorithme de Google présélectionne les articles qu’il présente)
* Manque de transparence (on ignore sur quelles bases)

Certains sites bloquent le scraping et d’autres limitent le nombres de requètes.

## Idées d’améliorations

* Mise en place de JavaScript pour une actualisation des pages dynamique
* Statistiques plus détaillées
* Statistiques propres aux données affichées dans la partie classement
* Mise en place d’un système de pagination
* Clusterisation des articles pour essayer de faire apparaître des catégories
* Extraction des Noms propres, classement par positivité, possibilité d’afficher des statistiques et de lier les articles liés
* Compte utilisateur et système de recommandations
* CSS et retouches esthétiques
* ...