



École Nationale Supérieure d’Informatique pour l’Industrie et l’Entreprise

Rapport de stage

« Développement d’une application pour le
Web-Scraping et l’analyse des données textuelles »

Étudiant : Adib HABBOU

Tuteur académique à l’ENSIIE : Nicolas BRUNEL

المملكة المغربية



المندوبية السامية للتخطيط

+٢٠٣٤٥٤٦ | +٢٠٣٤٥٣٥

HAUT-COMMISSARIAT AU PLAN

Établissement : Haut-Commissariat au Plan (Rabat, Maroc)

Maître de stage : Oumaima HOURRANE

Stage de première année du lundi 23 mai 2022 au vendredi 15 juillet 2022

Préambule

Remerciements

Tout d'abord, je voulais remercier Oumaima Hourrane, qui a été mon maître de stage, et qui m'a accompagné tout au long de celui-ci en me fournissant de précieux conseils.

Ensuite, un grand merci à Mohammed Ramdani, pour m'avoir mis en contact avec le Haut-Commissariat au Plan, ce qui m'a permis d'obtenir ce stage.

Enfin, je remercie Hamza Lotf pour m'avoir pris en tant que stagiaire et pour m'avoir accueilli au sein du Haut-Commissariat au Plan.

Résumé

Au cours de ce stage, j'ai pu réaliser des tâches de **Web Scraping** sur des sites de presse en ligne marocaine pour récupérer des données textuelles.

J'ai ensuite appliqué des modèles de machine learning **NLP** et plus précisément de **Topic Modeling** pour faire ressortir les sujets principaux des articles. J'ai également réalisé une tâche de **Text Classification** des articles par catégories.

Enfin, j'ai développé un **DashBoard** (tableau de bord interactif) pour pouvoir présenter visuellement les résultats obtenus lors des tâches précédentes.

Mots clés

Web Scraping - Natural Language Processing - Topic Modeling - Text Classification

Abréviations

- **HCP** : Haut-Commissariat au Plan
- **IA** : Intelligence Artificielle
- **ML** : Machine Learning
- **NLP** : Natural Language Processing
- **BDS** : Bases de Données Statistiques
- **LDA** : Latent Dirichlet Allocation
- **PLSA** : Probabilistic Latent Semantic Analysis
- **NLTK** : Natural Language Toolkit
- **SVM** : Support Vector Machine
- **SGD** : Stochastic Gradient Descent
- **MLP** : Multi-Layer Perceptron

Table des matières

1	Introduction	4
1.1	Structure d'accueil	4
1.2	Sujet et objectifs	4
2	Environnement	5
2.1	Locaux	5
2.2	Encadrement	5
2.3	Suivi du stage	5
2.4	Développement	5
3	Problématique	6
3.1	Contexte	6
3.2	Enjeux et perspectives	6
3.3	Activités du HCP	6
3.4	Méthodologie du HCP	7
3.5	Existant en interne	7
3.6	Missions du stage	7
4	Outils et concepts	8
4.1	Outils	8
4.2	Concepts	8
5	Phase de recherche	9
5.1	Web Scraping	9
5.1.1	Définition et applications	9
5.1.2	Implémentation en Python	9
5.2	Textual Data Visualisation	9
5.2.1	Définition et applications	9
5.2.2	Implémentation en Python	9
5.3	Topic Modeling	10
5.3.1	Définition NLP	10
5.3.2	Définition Topic Modeling	10
5.3.3	Explication LDA	11
5.3.4	Distribution de Dirichlet	11
5.3.5	Étapes du LDA	12
5.3.6	Visualisation du LDA	12
6	Web Scraping	13
6.1	Prérequis	13
6.2	Méthodologie générale	13
6.3	Récupération des liens	13
6.4	Récupération des informations	15
7	Web Scraping Massif	16
7.1	Définition avec le Web Scraping	16
7.2	Résultats obtenus	16
7.3	Perspectives	16

8 Textual Data Visualisation	17
8.1 WordCloud Anglais	17
8.2 WordCloud Français	18
9 Topic Modeling	19
9.1 Modèle non supervisé	19
9.2 Preprocessing	19
9.3 Conversion	19
9.4 Application	19
9.5 Cohérence	19
9.6 Résultats	20
10 Text Classification	21
10.1 Modèle supervisé	21
10.2 Preprocessing	21
10.3 Conversion	21
10.4 K Nearest Neighbour	21
10.5 Logistic Regression	22
10.6 Decision Tree	22
10.7 Random Forest	23
10.8 Support Vector Machine	24
10.9 Stochastic Gradient Descent	25
10.10 Multi-Layer Perceptron	26
10.10.1 Réseaux neuronaux	26
10.10.2 Rétro-propagation	27
10.11 Évaluation	27
10.12 Résultats Classification Report	28
10.13 Interprétation	30
10.14 Résultats Confusion Matrix	31
11 DashBoard	34
11.1 Streamlit	34
11.2 Résultat	34
11.3 Configuration	37
12 Conclusion	38
13 Bibliographie	39
14 Glossaire	41
15 Annexes	42
15.1 Développement Durable et Responsabilité Sociétale	42
15.2 Annexe A : Extrait Code Web Scraping	44
15.3 Annexe B : Extrait Code WordCloud	47
15.4 Annexe C : Extrait Code Topic Modeling	48
15.5 Annexe D : Extrait Code Text Classification	52
15.6 Annexe E : Extrait Code DashBoard	62

1 Introduction

1.1 Structure d'accueil

Le Haut-Commissariat au Plan est une structure ministérielle marocaine érigée en septembre 2003 en une administration de mission, sous l'autorité d'un haut commissaire au plan nommé, avec rang de Ministre, par Sa Majesté le Roi.



FIGURE 1 – Logo du Haut Commissariat au Plan

C'est le principal producteur de l'information statistique économique, démographique et sociale au Maroc. Il est également chargé de l'établissement des comptes de la nation. Pour ce faire il élaborer des études dans les domaines de la conjoncture, du cadrage macroéconomique et de la prospective.

Il dispose d'un observatoire des conditions de vie des ménages et d'un centre d'études et de recherches démographiques. Le HCP se conforme dans ses statistiques et ses études aux normes internationales et est admis depuis 2005 à la Norme Spéciale de la Diffusion des Données du Fond Monétaire Internationale.

1.2 Sujet et objectifs

Au cours du stage, l'objectif principal était de collecter des données textuelles à partir des sites de presse en ligne marocains en différentes langues (Anglais, Français et Arabe) en utilisant des techniques de Web Scraping et d'ensuite d'appliquer des modèles de machine learning NLP pour extraire les sujets principaux de ses articles puis enfin d'appliquer des modèles de classification pour labéliser les articles en fonction de leur catégorie.

Ce travail rentre en compte dans les missions du HCP puisqu'il permet de récupérer des données économiques et sociales à partir d'articles de presse et ensuite de les trier et de les classer en vue de servir plus tard pour une analyse plus poussée et plus large.

Mon travail au HCP ayant été mené dans une logique de tâches, mon rapport sera divisé entre les principales tâches que mon stage a constituées :

1. Recherches sur les sujets du stage (Web Scraping, Text Data Vis, Topic Model)
2. Web Scraping sur trois sites de presse en ligne en anglais, français et arabe
3. Web Scraping massif sur deux sites de presse en ligne en anglais et français
4. Application du Topic Modeling sur les données textuelles collectées
5. Application de techniques de classification de texte sur les articles
6. Réalisation d'un dashboard pour présenter les résultats obtenus

2 Environnement

2.1 Locaux

Le Haut-Commissariat au Plan admet une Direction des Systèmes d'Information Statistique qui a pour mission de collecter des données statistiques économiques et sociales, les traiter et les analyser ainsi que gérer la banque de données statistiques. C'est au sein de cette direction que j'ai eu la chance d'effectuer mon stage. Ses locaux se trouvent dans le quartier de Hay Riad à Rabat au Maroc.



FIGURE 2 – Bâtiment HCP à Hay Riad, Rabat, Maroc

2.2 Encadrement

J'ai eu la chance lors de mon stage d'être encadré par Oumaima Hourrane, doctorante en IA, ML et NLP qui occupe un poste d'ingénieur d'état au sein de la Direction des Systèmes d'Information Statistique du HCP à Rabat au Maroc.

2.3 Suivi du stage

Il a été établit avec ma maître de stage de faire une réunion par semaine afin de la tenir au courant de l'avancement de mes tâches, de mes éventuels problèmes mais aussi pour pouvoir discuter des futures tâches qui allait m'être attribué. Je pouvais également lui poser des questions ou de lui demander de l'aide sur les problèmes que je rencontrais tout au long de la semaine.

2.4 Développement

Concernant l'environnement de développement j'ai travaillé avec des Notebook Python sur l'outil Google Colaboratory ce qui permettait d'inclure en plus du code en Python du texte pour commenter ou expliquer le code. Cet outil c'est aussi avérer être très utile pour pouvoir travailler simultanément avec ma maître de stage sur le même Notebook.

3 Problématique

3.1 Contexte

L'objectif principal de ce stage réside dans la récupération et la classification des données textuelles à partir de sites de presse en ligne. Cet objectif rentre en compte dans un travail plus global réalisé dans un contexte d'administration publique qui souhaite recueillir un maximum d'informations donnant la réalité économique, sociale, culturelle du pays et son évolution, et cela d'une manière précise, fondée sur des données scientifiques.



FIGURE 3 – Capture d'écran du site Web HCP

3.2 Enjeux et perspectives

Ces informations vont ensuite servir à mener des études d'importance capitale puisqu'elles sont les bases sur lesquelles les différents ministères du royaume ainsi que de nombreuses institutions publiques établissent leurs décisions qui sont ensuite directement appliquées sur le terrain au service de la population.

De plus, à travers sa politique d'Open Data, le HCP fournit des données anonymisées en libre accès sur son site web et qui peuvent donc être utile à divers organismes indépendants nationaux ou internationaux que ce soit pour faire des études de marché pour le secteur privé ou encore pour mener des études académiques et universitaires.

Les domaines étudiés par le HCP sont très nombreux, on peut citer parmi eux l'emploi, le chômage, la démographie, la consommation, la santé, la pauvreté, les inégalités, le développement durable, l'habitat, l'instruction, l'éducation ou encore le budget.

3.3 Activités du HCP

Afin de remplir ces objectifs cruciaux qui relèvent d'une importance nationale, le HCP se donne les moyens de son ambition en s'entourant de spécialistes en mathématiques appliquées et en mettant la statistique au cœur de son mode de fonctionnement comme l'a si bien dit le Haut-Commissaire au Plan, Monsieur Ahmed Alami Lahlimi dans une interview donnée à l'occasion de la journée mondiale de statistique¹.

3.4 Méthodologie du HCP

Les différentes étapes du processus s'articulent donc comme suit :

1. Récupération des données à travers différentes sources
2. Nettoyage des données afin qu'elle soit exploitable
3. Classification et tri des données pour qu'elle soit plus facilement exploitable
4. Stockage des données dans les BDS du Haut-Commissariat au Plan
5. Analyse des données pour extraire des indicateurs et des informations clés
6. Publication de rapports (trimestriel et annuel) sur les différents domaines étudiés

3.5 Existant en interne

Actuellement au sein du HCP, le Web Scraping n'est pas encore utilisé à grande échelle de manière automatisée et sur un grand nombre de site web. Son utilisation est encore en phase de test à l'heure actuelle. Ainsi mon sujet de stage relève de l'envie du HCP d'explorer des nouvelles technologies comme le Big Data et l'IA et de les exploiter pour renforcer le processus classique actuel de collecte et analyse de données et d'informations.

De même pour le topic modeling et la classification de texte qui sont en phase de test actuellement sur différents types de données en vue d'être déployé à grande échelle dans les mois qui viennent pour pouvoir devenir à long terme des outils fondamentaux pour la réalisation d'études.

3.6 Missions du stage

Ma mission principale tout au long de ce stage a été la récupération et la classification des données textuelles à partir de sites de presse en ligne marocaine. Ma mission principale a été subdivisé en plusieurs missions suivantes :

1. Recherches sur les sujets du stage, réalisation de Notebook Python pour tester les techniques et préparation d'une présentation pour ma maître de stage ²
2. Entraînement au Web Scraping sur trois sites de presse en ligne Morocco World News (en anglais), Le Matin (en français) et Hespress (en arabe) afin de récupérer quelques centaines d'articles
3. Web Scraping massif sur deux sites de presse en ligne Morocco World News (en anglais) et Le Matin (en français) pour récupérer des milliers d'articles
4. Application des techniques Topic Modeling sur les données textuelles collectées à partir de Morocco World News et Le Matin pour obtenir les topics des articles
5. Application de techniques de classification sur les articles par catégorie afin d'entraîner un modèle à déterminer la catégorie d'un article de presse
6. Réalisation d'un dashboard pour présenter les différents résultats obtenus

4 Outils et concepts

4.1 Outils

Les principaux outils à disposition du HCP pour la collecte des données à l'heure actuelle sont des enquêtes statistiques officielles qui ont directement lieu sur le terrain auprès de la population. La plus grande enquête ayant eu lieu récemment est le Recensement Général de la Population et de l'Habitat qui a permis de récolter beaucoup de données disponibles actuellement sur le site du HCP.

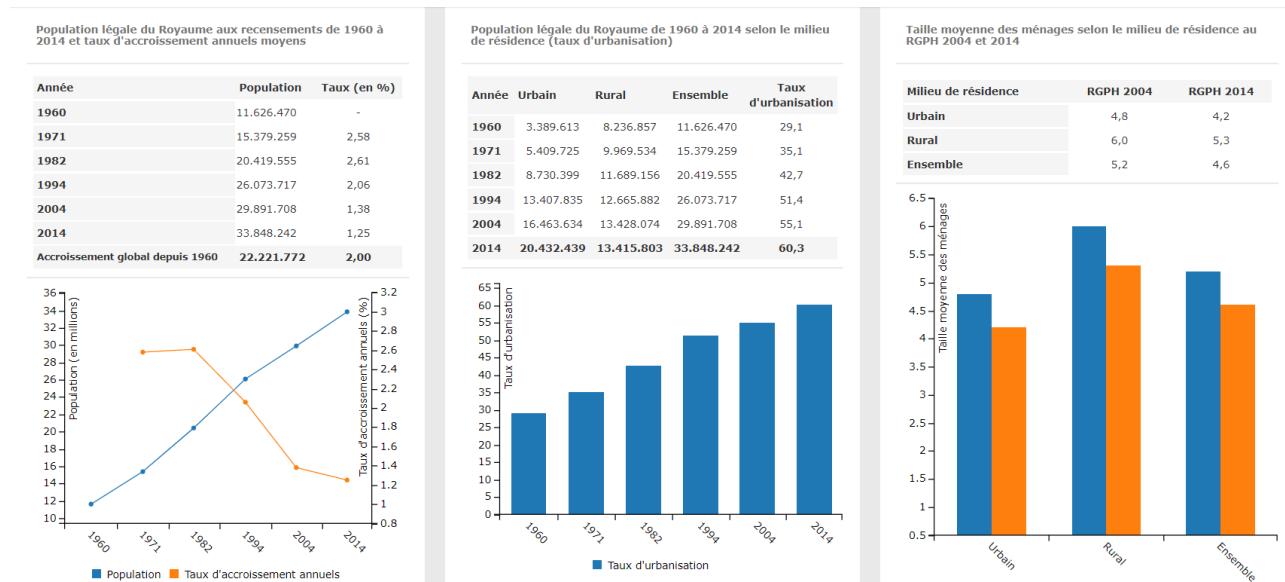


FIGURE 4 – Capture d'écran du site web 'RGPH en tableaux'

Concernant l'analyse des données les outils dont dispose le HCP à l'heure actuelle sont les indicateurs et des agrégats statistiques comme l'indice des prix à la consommation, l'indice du coût de vie ou encore l'indice du commerce extérieur.

L'infrastructure de stockage des données est ensemble des bases de données statistiques organisées de manière régionale et mise en relation avec une liste de métadonnées et une nomenclature nationale prédéfinie par le HCP.

4.2 Concepts

Les processus informatiques du HCP dépendent actuellement d'approches Rule-Based (moteur de règles en français) qui appliquent des règles créées par l'homme pour stocker, trier et manipuler des données sous forme de If Statements.

La manipulation de concept comme le Big Data, l'Intelligence Artificielle et le Machine Learning est actuellement explorée afin d'étudier la faisabilité et la mise en place d'infrastructures tangibles, capable d'élargir les applications de ces méthodes à grande échelle pour qu'à terme le Haut-Commissariat au Plan fasse partie des acteurs principaux et novateurs du numérique au Maroc.

5 Phase de recherche

5.1 Web Scraping

5.1.1 Définition et applications

Le Web Scraping est une technique qui permet de récupérer de manière automatisée des données provenant de diverses pages web et de les transformer en d'autres formats plus exploitables.

Elle admet de nombreuses applications parmi lesquelles on peut citer l'étude de marché, la comparaison des prix, l'analyse des sentiments ou encore l'email marketing...

5.1.2 Implémentation en Python

La principale bibliothèque Python utilisé pour le Web Scraping de site web dynamique est **Selenium** qui est compatible avec plusieurs navigateurs. Pour des sites web statiques on peut également utiliser **BeautifulSoup** avec **requests** ou **urllib**.

5.2 Textual Data Visualisation

5.2.1 Définition et applications

Représenter visuellement le contenu d'un document texte est l'une des tâches les plus importantes dans le domaine du **Text Mining**. Le **Textual Data Visualization** permet de faire apparaître différentes données clés à partir de données textuelles.

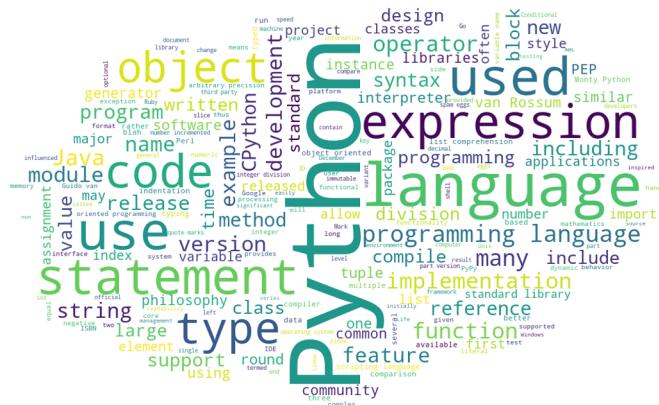


FIGURE 5 – Nuages de mots

5.2.2 Implémentation en Python

Pour ce faire en Python on peut utiliser différentes libraires comme WordCloud permet de synthétiser les notions les plus importantes d'un texte, plus un mot est présent dans le texte, plus il apparaît en gros dans le wordcloud.

D'autres librairies d'utilisation plus large comme Seaborn, Matplotlib ou encore Plotly peuvent également être utilisé pour visualiser certaines informations.

5.3 Topic Modeling

5.3.1 Définition NLP

Le **NLP** est un domaine de l'informatique qui vise à créer des outils de traitement de la langue naturelle pour diverses applications. Comme par exemple l'analyse de données de textuelles notamment le **Text Mining**, la traduction automatique, l'analyse de sentiment ou encore la correction automatique.

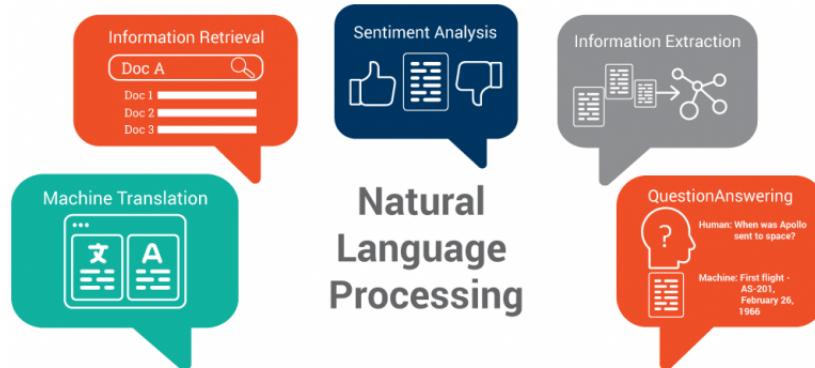


FIGURE 6 – Applications NLP

5.3.2 Définition Topic Modeling

Le **Topic Modeling** consiste lui à utiliser l'apprentissage non supervisé pour extraire les principaux sujets, représentés par un ensemble de mots, qui apparaissent dans une collection de documents. Le modèle le plus utilisé actuellement est celui du **Latent Dirichlet Allocation (LDA)** qui est une généralisation du **Probabilistic Latent Semantic Analysis (PLSA)**.

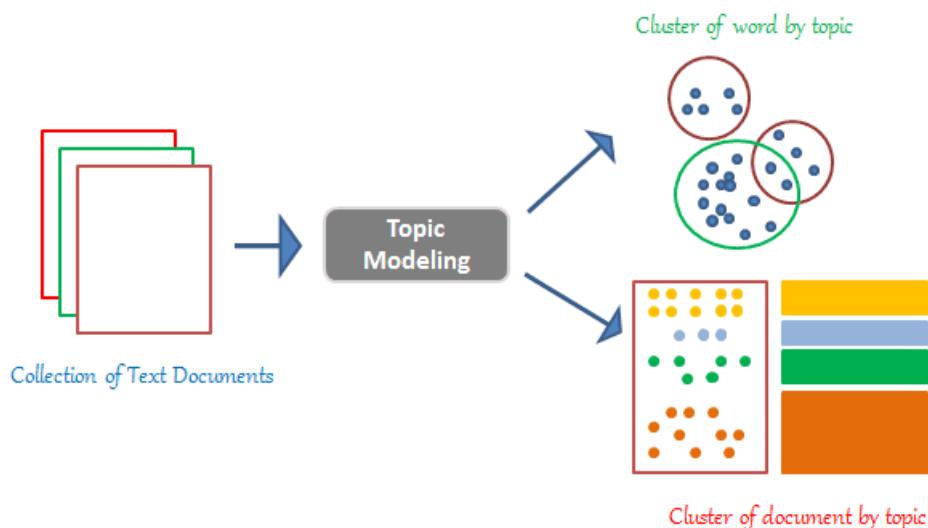


FIGURE 7 – Schéma Topic Model

5.3.3 Explication LDA

Le modèle **LDA** suppose que chaque document noté M est un mélange θ d'un petit nombre de sujets α et que la génération de chaque occurrence d'un mot w est attribuable à l'un des thèmes t du document. En pratique, on attribue un thème à chaque mot selon une distribution de Dirichlet tel que :

$$\theta_i \sim Dir(\alpha) \text{ pour } 1 \leq i \leq M \text{ avec } \alpha < 1$$

On obtient donc un premier **topic model**. Pour générer le suivant, on prend chaque mot et on met à jour le thème auquel il est lié. Ce nouveau thème est celui qui aurait la plus forte probabilité de le générer dans ce document : $\mathcal{P}(t|d) * \mathcal{P}(w|t)$. En répétant ce processus un grand nombre de fois on arrive à obtenir un **topic model** satisfaisant.

$\mathcal{P}(t|d)$: la probabilité que le document d soit assigné au thème t

$\mathcal{P}(w|t)$: la probabilité que le thème t soit assigné au mot w

5.3.4 Distribution de Dirichlet

La densité de probabilité d'une densité de Dirichlet s'écrit :

$$f(x_1, \dots, x_K, \alpha_1, \dots \alpha_K) = \frac{1}{B(\alpha)} \prod_{i=1}^K x_i^{\alpha_i-1}$$

Avec la fonction bêta suivante :

$$B(\alpha) = \frac{\prod_{i=1}^K \Gamma(\alpha_i)}{\Gamma(\sum_{i=1}^K \alpha_i)}$$

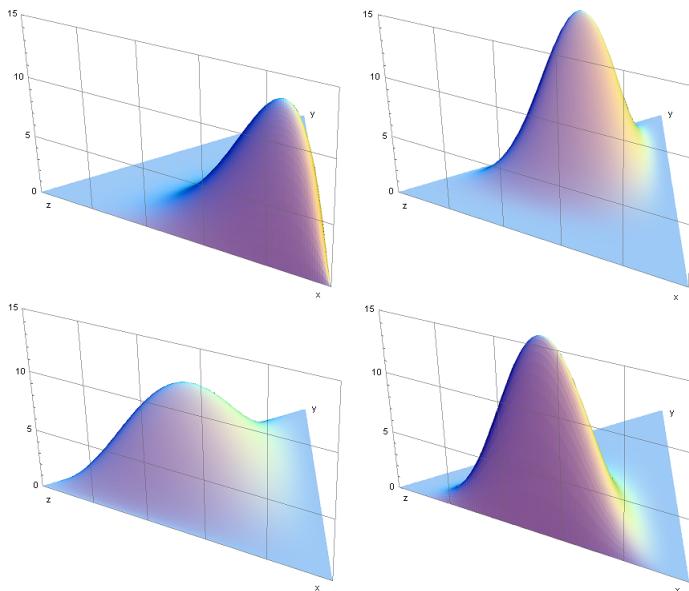


FIGURE 8 – Densité de la loi de Dirichlet lorsque $K = 3$

5.3.5 Étapes du LDA

Appliquer un modèle de **LDA** nécessite d'effectuer en amont un certains nombre d'opérations sur les données collectées.

1. **Tokenisation** : on divise les données textuelles en mots, on remplace les majuscules par des minuscules et on supprime la ponctuation ainsi que les mots de moins de 3 lettres.
2. **Lemmatisation** : les mots à la 3ème personne sont changés à la 1ère personne et les verbes conjugués au passé ou au futur sont conjugués au présent.
3. **Racinalisation** : les mots sont réduits à leurs formes radicales.
4. **Conversion** : on stocke tout dans des dictionnaires où la clé est le mot en question et la valeur son nombre d'occurrences.

5.3.6 Visualisation du LDA

Pour ce faire on peut utiliser les fonctions des librairies **gensim** et **nltk** qui permettent de réaliser toutes ces étapes de preprocessing et d'ensuite appliquer le modèle souhaité. On peut également utiliser la librairie **pyLDAvis** pour extraire des informations d'un topic model LDA et les visualiser dans une application web interactive.

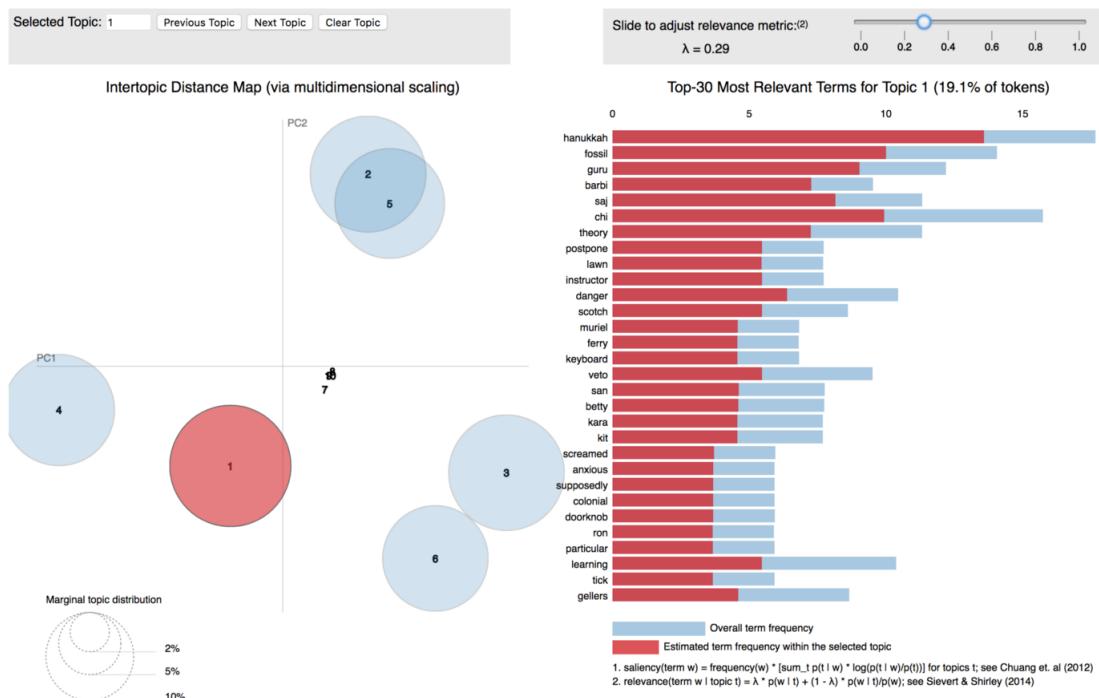


FIGURE 9 – Application Web Interactivve pyLDAvis

6 Web Scraping

6.1 Prérequis

Avant de procéder à l'extraction des données il faut tout d'abord préparer son environnement, pour ce faire on commence par installer le package selenium et le web driver dans notre environnement de développement.

Dans le cadre de ce stage, le navigateur utilisé était Google Chrome. Je suis donc passé par le web driver **chromium**. J'ai également eu besoin de la librairie **Pandas** et de l'instance **webdriver** de la librairie **Selenium**.

6.2 Méthodologie générale

La stratégie adoptée pour les sites de presse en ligne suivait les étapes suivantes :

1. Stocker d'abord les catégories du site dans une liste Python
2. Parcourir les catégories avec une boucle **for** :
<https://www.site.com/category>
3. Ajouter une autre boucle **while** pour incrémenter le numéro de page :
https://www.site.com/category/page_number
4. Pour chaque page de chaque catégorie, scraper uniquement les liens des articles et les stocker dans une liste Python
5. Parcourez la liste de liens d'articles et récupérez les textes à l'intérieur de chaque article pour les stocker **dataframe**
6. Convertir le **dataframe** obtenu en un fichier **csv**

6.3 Récupération des liens

Pour récupérer les liens des articles la méthode adoptée était de récupérer tous les éléments présents dans des balises **href** en utilisant la fonction de recherche selon le chemin **XPATH** de **Selenium** : `driver.find_elements_by_xpath('//.../a[@href]')`. Cette fonction rentrait une liste qu'on pouvait ensuite parcourir avec une boucle **for** pour récupérer le lien avec le getter `get_attribute('href')`.

Afin de s'entraîner à utiliser Selenium ma première tâche de Web Scraping ne nécessitait pas de récupérer un grand nombre d'articles mais plutôt un maximum d'informations sur un même article (titre, contenu, auteur, date de publication...).

Les trois sites de presse sur lesquelles le Web Scraping a été effectué au début sont les suivants : Morocco World News en anglais, Le Matin en français et Hespress en arabe.

MWN MOROCCO WORLD NEWS

SUPPORT US CAREERS CONTACT SIGN IN

NEWS | POLITICS | ECONOMY | OPINION | LIFESTYLE | FEATURES | SOCIETY | EDUCATION | WESTERN SAHARA

Fuel Prices in Morocco Reach New Heights

Less than two months after recording unprecedented increases, fuel prices in Morocco are again expected to reach new heights starting today, June 15.

Elmahdi Echabouch June 15, 2022



FIGURE 10 – Capture d'écran du site Morocco World News

LE MATIN.ma

LIVE COVID-19 Book Club Lifestyle Hi-tech Nos capsules d'info

Raja-WAC : la 25e journée de la Botola au rythme d'un derby casablancais décisif entre le leader et son dauphin

Reunion de travail à Madrid entre Laffit et son homologue espagnol

Volkswagen T-Roc 2.0 TDI Sport : restylage de mi-carrière réussi

Flambée de la Covid... les piqûres de rappel du ministère de la Santé

CYCLES DE CONFERENCES GROUPE LE MATIN EN SAVOIR +

FIGURE 11 – Capture d'écran du site Le Matin

الملك محمد السادس يؤكد حرصه على توفير تعليم جيد لجميع المغاربة

المملكة المغربية تطلق سجن بالرباط

أفرجوا على مواطنين لتأثير شبابهم

أمر دولي يوقف مواطناً بجزرنا بالمغرب

طيران الإمارات يطلق حملة توظيف بال المغرب

غيلان يودي ب尸首 التفاف بجماعة البلياء

حسن أبى يعاقب منادل البرت فى أغنية

موجة درجات الحرارة أوروبا قبل الصيف

الإمارات تجدد الدعم لموريتانيا

كابة ترفض التطبيق على 400 متر

الافتتاحيات تنتقد على التحرير المنساوية

فرنسا تحذف مسؤولًا كبيرًا في "داعش"

هذا يعيش بعقل طفل بمحارب المملكة

مطار جنوب سرت يستأنف الرحلات الجوية

جازوين يبرز تصور الحكومة حول اللائحة

تجدد دعم الصحراء بمجلس حقوق الإنسان

"بيت مال القدس" تختصر في معرض دكار

المملكة المغربية تطلق سجن بالرباط

أدوات من يفهم اللغة الرسمية إلى فراغة

مقال شعرية صحفة "السايس"، يوم الجمعة 10 يونيو 2022 أعادته لوثر غارسيا LUZ GOMEZ GARCIA

الحكومة الرسمية تنشأ عبر موقعها من ذمة الصحراوي، مما يقتضي تقديره على الأداء المتمم للقيادة الرئازية في تعاملها المشتبه مع إسبانيا بالطبع ما كنته المستحقة لوثر غارسيا هو وأيها، وهو رأي...

زنوجة حادة

FIGURE 12 – Capture d'écran du site Hespress

6.4 Récupération des informations

La principale différence entre les trois sites web réside dans la manière de récupérer ses informations. Les sites n'étant pas tous coder de la même façon il a fallu utiliser l'outil **Inspecter l'élément** du navigateur afin de regarder de plus près le code source.

Après analyse de ce code source, je devais déduire quelle était la manière la plus optimale de récupérer l'information souhaitée sachant que **Selenium** offrait une multitude d'option que ce soit par nom de la classe, par chemin XPATH, par identifiant ou encore par **CSS Selector**. Les fonctions de **Selenium** retournaient des listes dans lesquelles il fallait ensuite aller chercher l'information souhaitée qui se trouvait le plus généralement dans l'attribut **text**.

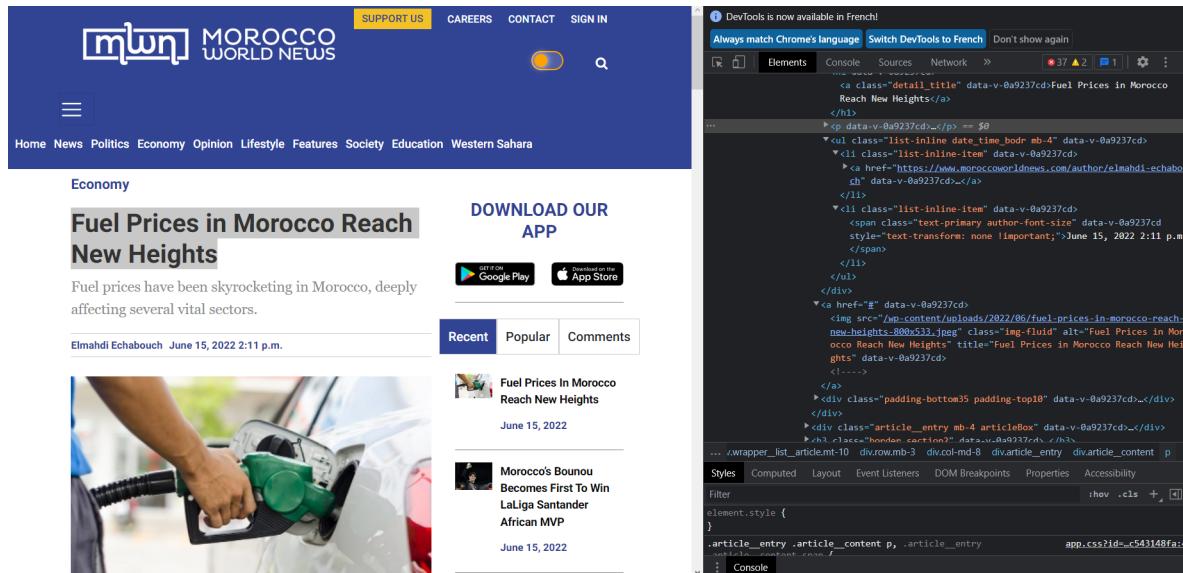


FIGURE 13 – Capture d'écran d'inspection de l'élément

Les données obtenues étaient stockées dans un **data frame** avant d'être converti en fichier **csv**. Les données récupérées ressemblaient à ceci :

	title	lead	author	date	content
0	Spanish FM: Spain Works on 'Constructive, Firm... Algeria decided to suspend its friendship trea...	Safaa Kasraoui	June 09, 2022 10:53 a.m.	Rabat - Spain regrets Algeria's decision to su...	
1	Afro-Atlantic Treaty: Morocco Says Only Unity,... Morocco's Foreign Affairs Minister insists tha...	Aya Benazizi	June 08, 2022 5:08 p.m.	Casablanca - With the Moroccan capital of Rab...	
2	Cape Verde to Open Consulate General in Morocc... Togo also announced the forthcoming opening of...	Safaa Kasraoui	June 08, 2022 4:40 p.m.	Rabat - Cape Verde's Foreign Affairs Minister ...	
3	Mexican Delegation To Visit Morocco, Keen To C... Migration management, the fight against climat...	Souad Anouar	June 08, 2022 4:20 p.m.	Rabat - A Mexican parliamentary delegation is ...	
4	Spanish PM Renews Support for Morocco's Autono... A number of marginal opposition political part...	Safaa Kasraoui	June 08, 2022 2:02 p.m.	Rabat - Spain's Prime Minister Pedro Sanchez h...	

FIGURE 14 – Data Frame de Morocco World News (titre, en-tête, auteur, date, contenu)

7 Web Scraping Massif

7.1 Différences avec le Web Scraping

Une fois la prise en main de Selenium et des techniques de **Web Scraping** réalisé ma tâche suivante était de récupérer un grand nombre d'articles sur Morocco World News et Le Matin en appliquant la même technique que précédemment mais en ajoutant des catégories et en augmentant le nombre de pages visitées.

L'autre différence avec la tâche précédente était que je n'avais plus besoin de récupérer le titre, la date et l'auteur de l'article mais uniquement son contenu textuel et sa catégorie qui allait servir plus tard pour de la classification de textes.

7.2 Résultats obtenus

	category	content
0	politics	Rabat - A confidential report from NATO has ex...
1	politics	Rabat - Top security officials from Morocco an...
2	politics	Rabat - Indie-rock band Big Thief has announce...
3	politics	Rabat - The European Union has called on Alger...
4	politics	Rabat - Spain regrets Algeria's decision to su...

FIGURE 15 – Data Frame de Morocco World News (catégorie et contenu)

7.3 Perspectives

Concernant la taille de données récupérées j'ai pu collecter des informations de 7000 articles de Morocco World News et 8000 articles de Le Matin. La principale limite à la collecte de plus d'informations est le temps d'exécution dû au manque de puissance de calcul.

L'outil Google Colaboratory en version gratuite permet d'utiliser des machines virtuelles pas très puissantes. Toutefois, en utilisant la puissance de calcul à disposition du HCP on peut très largement augmenter le nombre de pages visité dans la boucle `while` et utiliser le même programme pour récupérer des centaines de milliers d'articles.

La même méthodologie peut également être étendue à d'autres sites web de presse en ligne marocain comme Médias24, Aujourd'hui ou encore Le360 puisqu'ils fonctionnent tous avec le même système d'url qui contient à la fois le nom de la catégorie et le numéro de la page. On peut aussi utiliser la même méthodologie pour récupérer des informations sur d'autres types de sites web qui fonctionnent par catégorie et page.

8 Textual Data Visualisation

En utilisant la librairie **WordCloud** de Python on arrive à obtenir les images suivantes correspondantes à deux articles de Morocco World News et deux articles de Le Matin :

8.1 WordCloud Anglais

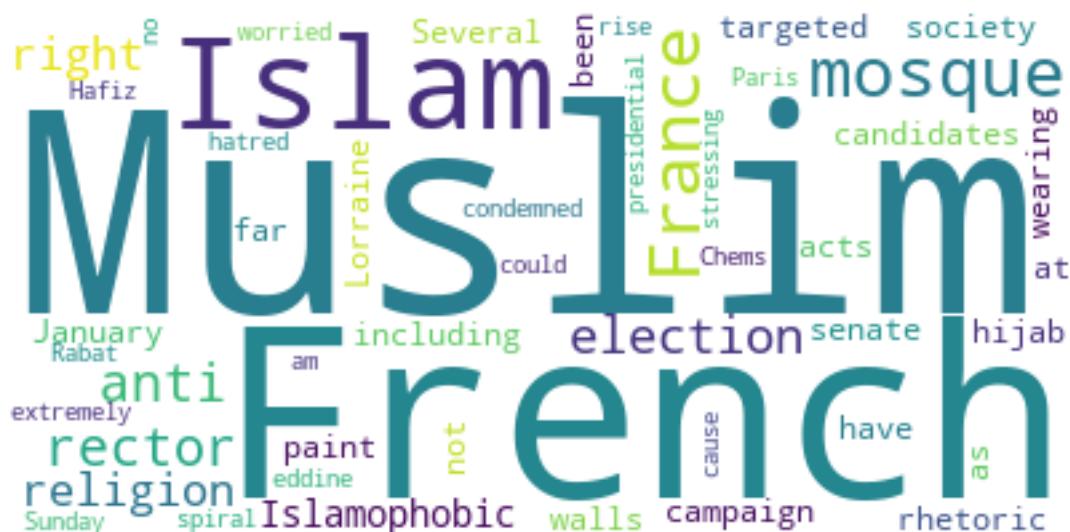


FIGURE 16 – Wordcloud d'un article de Morocco World News



FIGURE 17 – Wordcloud d'un article de Morocco World News

8.2 WordCloud Français



FIGURE 18 – Wordcloud d'un article de Le Matin



FIGURE 19 – Wordcloud d'un article de Le Matin

9 Topic Modeling

9.1 Modèle non supervisé

Le **Topic Modeling** est un modèle de machine learning **non supervisé**, c'est-à-dire qu'il ne nécessite pas d'entraînement. Les données n'étant pas étiquetées, le modèle doit donc découvrir les structures sous-jacentes à ces données de lui-même.

9.2 Preprocessing

On commence tout d'abord par réaliser certaines opérations sur nos données notamment la suppression des stopwords et des mots de moins de 3 lettres puis la lemmatisation après avoir supprimé les valeurs NAN. Pour ce faire ont utilisé les fonctions de la librairie `nltk`.

9.3 Conversion

On stocke nos données après leur nettoyage dans un dictionnaire de **gensim** pour ensuite les convertir au format **Bag Of Words** c'est-à-dire en couple mot et nombre d'occurrences où la clé est représentée par le mot et la valeur par son nombre d'occurrences.

9.4 Application

On applique le topic modeling à l'aide de la fonction **LdaMulticore** de **gensim** en prenant bien soin de préciser le nombre de topic à extraire du corpus, le mapping entre les identifiants des mots (entiers) et les mots (chaîne de caractères) et le nombre à effectuer d'itération dans le corpus.

9.5 Cohérence

Les mesures de cohérence évaluent le degré de similitude sémantique entre les mots les mieux notés dans les topics. Ces mesures aident à faire la distinction entre les topics sémantiquement interprétables et les topics dus à des inférences statistiques. Pour un bon modèle LDA la cohérence doit être comprise entre 0.4 et 0.7 au-delà et en dessous le modèle est très probablement erroné.

La cohérence pour un modèle LDA est calculée en procédant aux étapes suivantes :

- **Segmentation** : création de paires de mots à partir de sous-ensembles ;
- **Calcul des probabilités** : calcul probabilité d'occurrence d'un mot ;
- **Mesure de confirmation** : vérification « dans quelle mesure » un sous-ensemble de mots supporte un autre sous-ensemble de mots dans chaque paire ;
- **Agrégation** : agrégation de toutes les valeurs calculées à l'étape précédente en une seule valeur qui est notre score final de cohérence de sujet.

9.6 Résultats

On stocke finalement les topics obtenus dans un data frame Pandas pour ensuite les garder dans un fichier csv. On peut également utiliser **pyLDAvis** pour visualiser les topics de manière interactive directement sur notre Notebook Python.

	0	1	2	3	4	5	6	7	8	9
Topic 1	(0.026, onlin)	(0.025, digit)	(0.025, exam)	(0.022, candid)	(0.020, internet)	(0.019, network)	(0.017, port)	(0.015, baccalaur)	(0.015, facebook)	(0.014, survey)
Topic 2	(0.017, polisario)	(0.013, spanish)	(0.009, trump)	(0.008, resolut)	(0.006, migrant)	(0.006, migrat)	(0.005, vote)	(0.005, secretari)	(0.005, autonomi)	(0.005, sovereignti)
Topic 3	(0.029, learner)	(0.016, classroom)	(0.014, method)	(0.011, grammar)	(0.011, text)	(0.010, target)	(0.009, reader)	(0.009, facebook)	(0.008, comprehens)	(0.007, border)
Topic 4	(0.008, discours)	(0.007, attitud)	(0.007, book)	(0.006, concept)	(0.006, influenc)	(0.006, theori)	(0.006, principl)	(0.006, ident)	(0.006, linguist)	(0.005, religion)
Topic 5	(0.014, vaccin)	(0.011, bank)	(0.010, agricultur)	(0.009, travel)	(0.008, price)	(0.008, tourism)	(0.008, export)	(0.007, food)	(0.007, test)	(0.007, flight)
Topic 6	(0.015, parent)	(0.012, classroom)	(0.009, exam)	(0.007, applic)	(0.006, write)	(0.005, answer)	(0.005, graduat)	(0.005, degre)	(0.005, motiv)	(0.005, grade)
Topic 7	(0.007, violenc)	(0.005, sexual)	(0.004, polic)	(0.004, love)	(0.004, girl)	(0.004, religion)	(0.004, victim)	(0.003, protest)	(0.003, stori)	(0.003, street)
Topic 8	(0.015, rank)	(0.011, innov)	(0.010, youth)	(0.009, confer)	(0.008, competit)	(0.006, british)	(0.006, team)	(0.006, vocat)	(0.006, professor)	(0.005, engin)
Topic 9	(0.013, algerian)	(0.012, israel)	(0.008, terror)	(0.007, iran)	(0.007, regim)	(0.007, terrorist)	(0.007, isra)	(0.006, democraci)	(0.006, saudi)	(0.006, west)
Topic 10	(0.021, mother)	(0.016, tongu)	(0.013, literaci)	(0.011, tamazight)	(0.010, curriculum)	(0.008, linguist)	(0.007, amazigh)	(0.007, unesco)	(0.007, programm)	(0.007, coloni)

FIGURE 20 – Topic Model Morocco World News

	0	1	2	3	4	5	6	7	8	9
Topic 1	(0.039, alopec)	(0.025, médic)	(0.022, cheveux)	(0.019, cultur)	(0.017, fréquent)	(0.017, repouss)	(0.017, poil)	(0.014, agad)	(0.014, corp)	(0.013, américain)
Topic 2	(0.042, logist)	(0.026, digitalis)	(0.024, transport)	(0.020, salon)	(0.015, variol)	(0.014, sing)	(0.014, adapt)	(0.014, professionnel)	(0.009, urgenc)	(0.009, tedros)
Topic 3	(0.071, cybersécur)	(0.044, hackathon)	(0.035, Imp)	(0.021, startup)	(0.020, numer)	(0.020, météorolog)	(0.019, orag)	(0.018, security)	(0.018, cyb)	(0.018, enjeux)
Topic 4	(0.041, taux)	(0.020, heur)	(0.015, mainten)	(0.015, immun)	(0.015, majest)	(0.014, instant)	(0.012, plateform)	(0.012, réclam)	(0.012, médical)	(0.011, royal)
Topic 5	(0.033, franklin)	(0.019, personnag)	(0.019, américain)	(0.018, britann)	(0.018, incarn)	(0.018, franco)	(0.017, pierr)	(0.017, épisod)	(0.017, michael)	(0.016, benjamin)
Topic 6	(0.006, format)	(0.006, conseil)	(0.005, relat)	(0.005, gouvern)	(0.005, régional)	(0.004, commun)	(0.004, domain)	(0.004, initi)	(0.004, professionnel)	(0.004, univers)
Topic 7	(0.006, vari)	(0.006, enfant)	(0.005, femm)	(0.005, omicron)	(0.004, risqu)	(0.004, médecin)	(0.004, prix)	(0.004, expliqu)	(0.004, sanitair)	(0.004, faut)
Topic 8	(0.022, hybrid)	(0.022, aircross)	(0.016, conduit)	(0.016, styl)	(0.016, confort)	(0.015, motoris)	(0.013, conducteur)	(0.013, litr)	(0.013, bord)	(0.013, moteur)
Topic 9	(0.029, rir)	(0.025, afric)	(0.025, startup)	(0.019, talent)	(0.019, spectacl)	(0.018, forum)	(0.018, catégor)	(0.018, char)	(0.013, francophon)	(0.012, nomm)
Topic 10	(0.026, trafic)	(0.019, euro)	(0.018, perform)	(0.018, nador)	(0.017, west)	(0.015, aéroport)	(0.015, banqu)	(0.015, commercial)	(0.014, sal)	(0.014, régional)

FIGURE 21 – Topic Model Le Matin

10 Text Classification

10.1 Modèle supervisé

La classification de texte est un modèle de machine learning **supervisé**, c'est-à-dire qu'il nécessite un entraînement préalable sur des données étiquetées, le modèle doit apprendre une fonction de prédiction à partir des données annotées dont il dispose.

10.2 Preprocessing

On commence tout d'abord par réaliser certaines opérations sur nos données notamment la suppression des caractères spéciaux, des caractères uniques, des espaces multiples puis on convertit tout en minuscule avant d'appliquer la lemmatisation.

10.3 Conversion

On stocke nos données après leur nettoyage dans un **Bag Of Words**. On utilise ensuite la méthode **Term Frequency - Inverse Document Frequency** pour pouvoir pondérer l'importance d'un mot dans un document. Pour ce faire on utilise les fonctions `CountVectorizer` et `TfidfTransformer` de la librairie `sklearn.feature_extraction.text`.

10.4 K Nearest Neighbour

Le modèle de classification des k plus proches voisins est un des modèles de classification les plus simples, il suit l'algorithme suivant :

1. Sélectionner le nombre k de voisins et calculer les distances ;
2. Prendre les k voisins les plus proches selon la distance calculée ;
3. Parmi ces k voisins, compter le nombre de points appartenant à chaque catégorie ;
4. Attribuer le nouveau point à la catégorie la plus présente parmi ces k voisins.

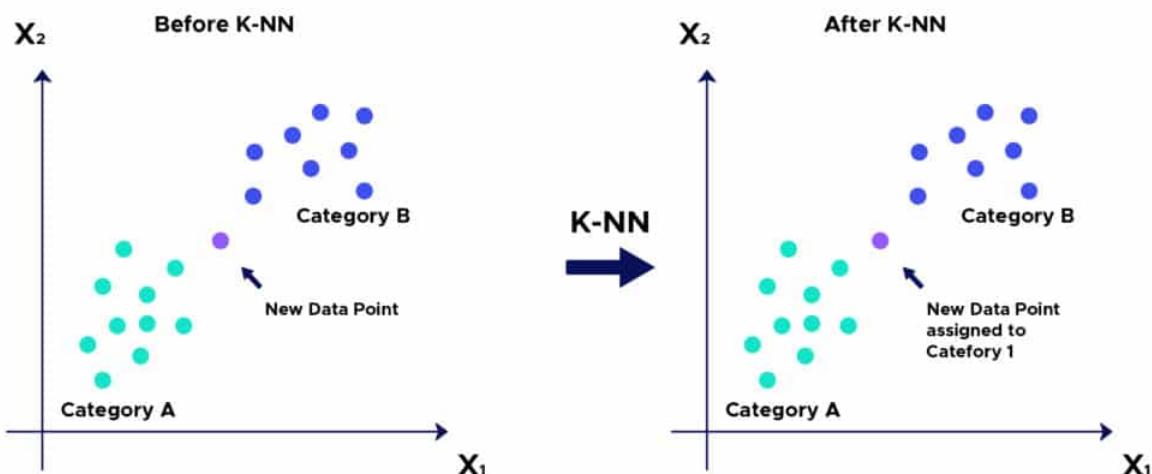


FIGURE 22 – Une itération de l'algorithme KNN

10.5 Logistic Regression

Un modèle de régression logistique permet aussi de prédire la probabilité qu'un événement arrive. Lorsque la valeur prédictée est supérieure à un seuil, l'événement est susceptible de se produire. Pour ce faire, on utilise une fonction sigmoïde $\sigma(x) = \frac{1}{1+\exp -x}$.

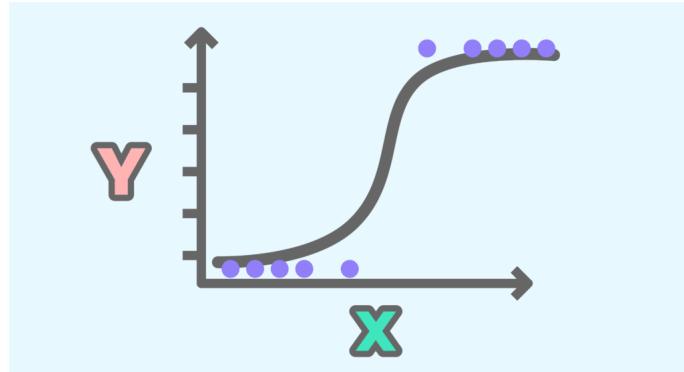


FIGURE 23 – Courbe Fonction Sigmoïde

La classification par régression logistique correspond à un simple problème d'optimisation où nous essayons d'obtenir les meilleurs paramètres θ afin que notre courbe sigmoïde $h(X) = \sigma(\theta X)$ colle le plus possible aux données d'entraînement.

10.6 Decision Tree

Sur un arbre de décision, chaque question correspond à un noeud. En fonction de la réponse à chaque question, nous allons nous orienter vers une branche de l'arbre pour finalement arriver sur une feuille de l'arbre qui contiendra la réponse finale.

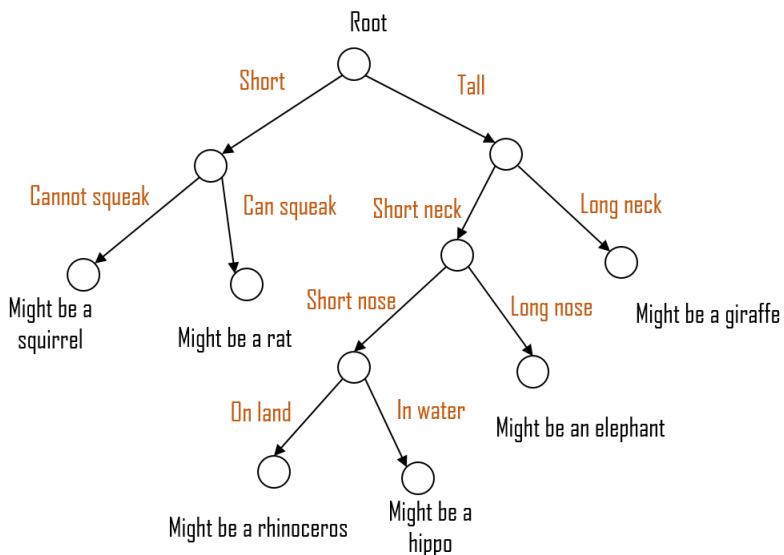


FIGURE 24 – Exemple d'arbre de décision pour la reconnaissance d'animaux

L'algorithme se demande à chaque noeud quelle feature est la plus intéressante en calculant le gain d'information. Son objectif final étant de maximiser ce gain c'est pourquoi l'arbre choisit la question qui maximise ce gain.

10.7 Random Forest

La forêt aléatoire est une méthode d'ensemble puisqu'elle combine plusieurs résultats pour obtenir un résultat final. Elle combine en réalité les résultats de plusieurs arbres de décision. Chaque arbre est entraîné sur un sous-ensemble aléatoirement constitué du dataset d'entraînement.

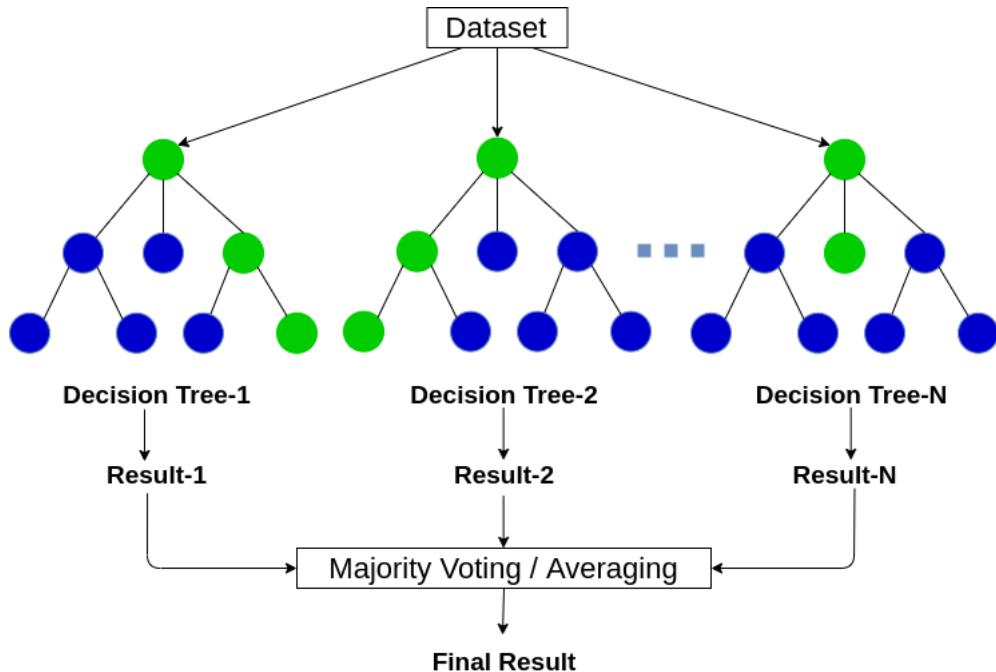


FIGURE 25 – Forêt Aléatoire de N Arbres de Décision

Les résultats de tous les arbres de décision sont alors combinés pour donner une réponse finale. Chaque arbre “vote” et la réponse finale est celle qui a eu la majorité de votes. C'est ce qu'on appelle la méthode du **bagging**.

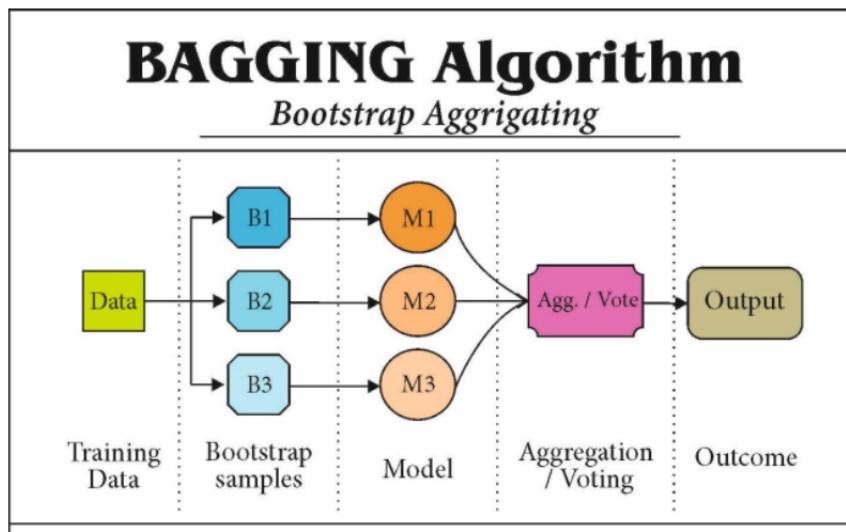


FIGURE 26 – Schéma de la méthode du Bagging

10.8 Support Vector Machine

La classification suivant une Machine à Vecteurs de Support repose sur le tracé d'une droite qui délimite des espaces différents et constitue une réponse au problème. On cherche donc la droite qui sépare le mieux nos classes en cherchant celle dont la marge est la plus élevée. La marge étant la distance qui sépare une droite de l'observation la plus proche.

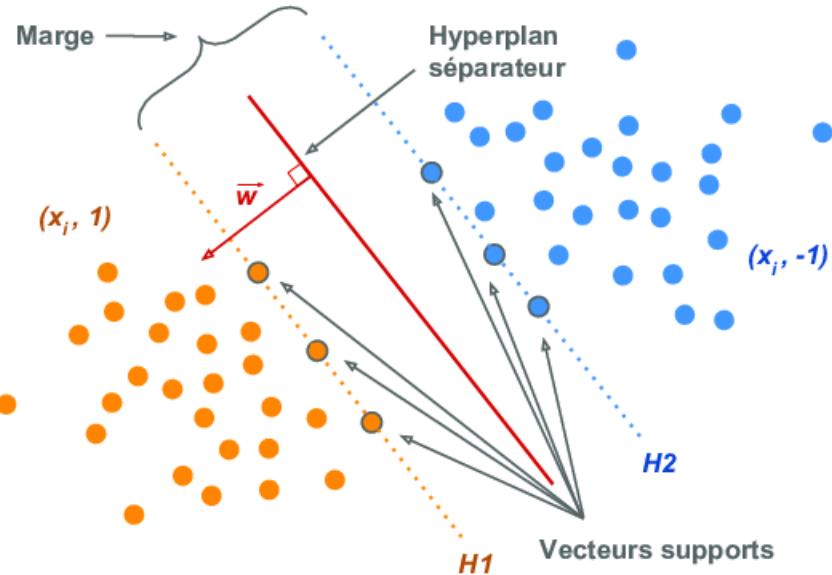


FIGURE 27 – Représentation du SVM

Malheureusement dans certains cas cette droite n'existe pas et le modèle n'est donc pas applicable ou alors il faut l'appliquer en omettant certaines valeurs.

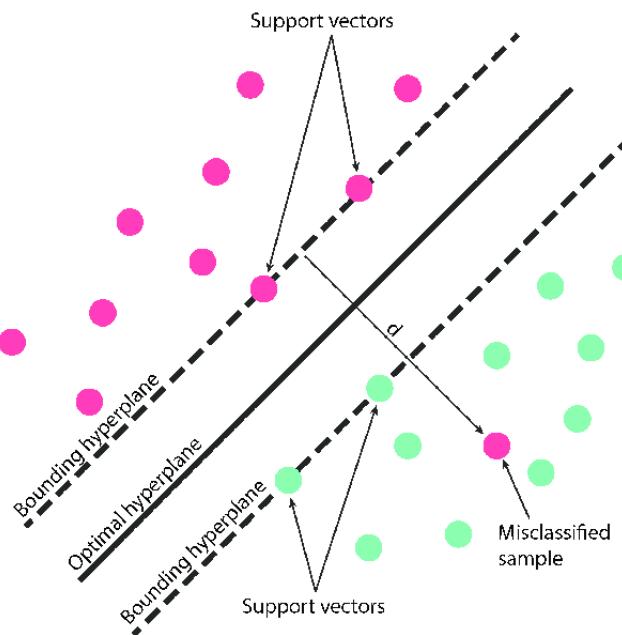


FIGURE 28 – Représentation du SVM avec une valeur extrême

10.9 Stochastic Gradient Descent

La Descente de Gradient est un algorithme d'optimisation qui permet de trouver le minimum de n'importe quelle fonction convexe en convergeant progressivement vers celui-ci. Dans notre cas la fonction à minimiser est la fonction de coût.

On part d'un point initial aléatoire puis on mesure la valeur de la pente en ce point en calculant le gradient. On progresse ensuite d'une certaine distance dans la direction de la pente qui descend.

Cette opération a pour résultat de modifier la valeur des paramètres de notre modèle. On répète ensuite ce processus de très nombreuses fois jusqu'à obtenir un modèle suffisamment satisfaisant.

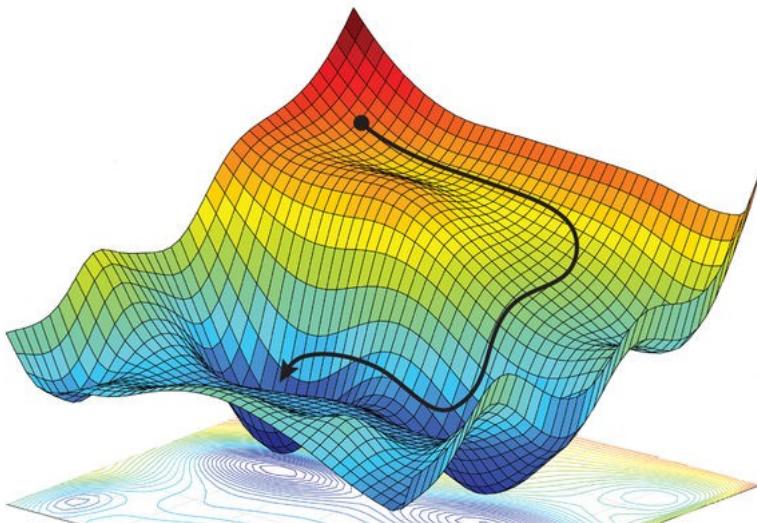


FIGURE 29 – Représentation de la descente de gradient

La descente de gradient stochastique est un type de descente de gradient qui traite un exemple d'entraînement par itération. Par conséquent, les paramètres sont mis à jour même après une itération dans laquelle un seul exemple a été traité. C'est donc plus rapide que la descente de gradient classique.

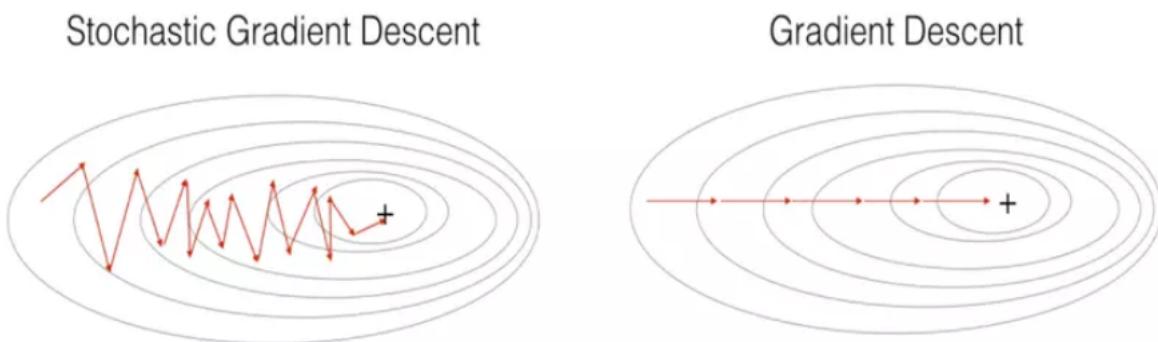


FIGURE 30 – Différence entre descente de gradient stochastique et classique

10.10 Multi-Layer Perceptron

10.10.1 Réseaux neuronaux

Le perceptron multicouche est un type de réseau neuronal artificiel à propagation directe c'est-à-dire au sein duquel l'information circule de la couche d'entrée vers la couche de sortie uniquement.

Le réseau se compose d'au moins trois couches de noeuds : une couche d'entrée, une couche cachée et une couche de sortie. À l'exception des noeuds d'entrée, chaque noeud est un neurone qui utilise une fonction d'activation non linéaire.

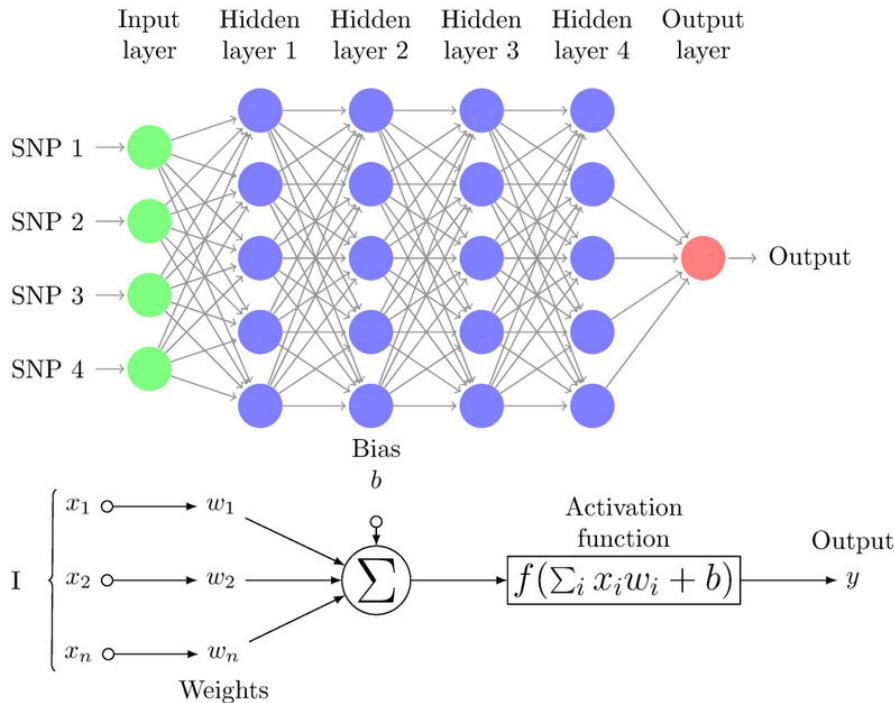


FIGURE 31 – Perceptron Multicouche

La fonction d'activation est une fonction mathématique appliquée à un signal en sortie d'un neurone artificiel. Le terme vient de l'équivalent biologique "potentiel d'activation" qui désigne le seuil de stimulation qui, une fois atteint entraîne une réponse du neurone.

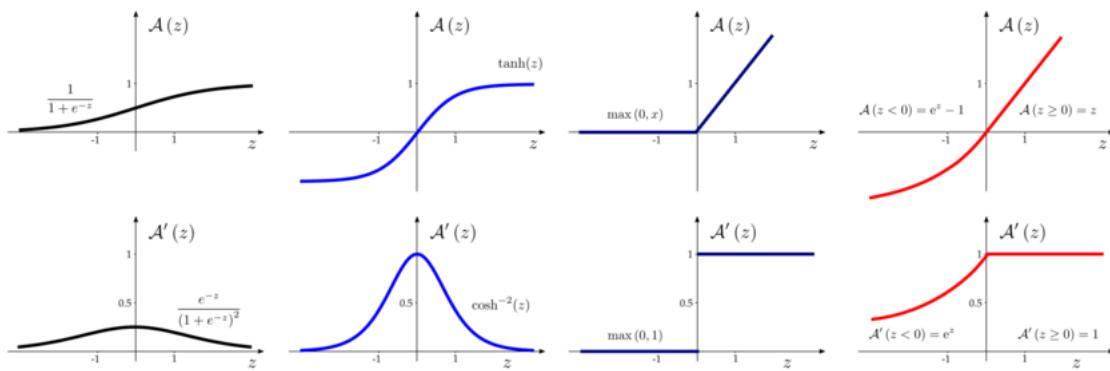


FIGURE 32 – Fonctions d'activation

10.10.2 Rétro-propagation

Le réseau utilise une technique d'apprentissage supervisée appelée **rétro-propagation**. Ses multiples couches et son activation non linéaire distinguent le réseau multicouche d'un perceptron linéaire. Il peut distinguer les données qui ne sont pas linéairement séparables.

La **rétro-propagation** calcule le gradient de la fonction de coût par rapport aux poids du réseau pour un seul exemple d'entrées-sorties en mettant à jour les poids pour minimiser les pertes.

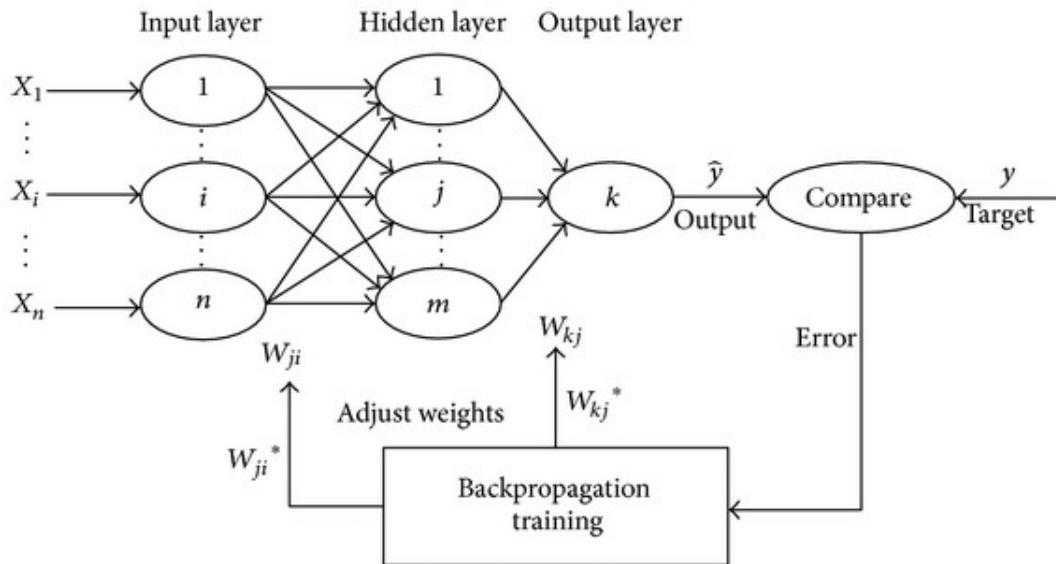


FIGURE 33 – Visualisation de la rétro-propagation

10.11 Évaluation

Pour évaluer la précision des modèles on va utiliser la librairie `sklearn.metrics` pour calculer le `accuracy_score` qui nous donne une mesure de la précision globale de notre modèle sur les données de test.

On utilise aussi la fonction `classification_report` qui nous donnent pour chacune des catégories de nos données les mesures suivantes :

- **précision** : taux de prédictions positives correctes ;
- **recall** : taux de positifs correctement prédits ;
- **f1-score** : capacité d'un modèle à bien prédire les individus positifs.

10.12 Résultats Classification Report

	precision	recall	f1-score	support
covid-19	0.80	0.82	0.81	277
economy	0.69	0.62	0.65	130
education	0.73	0.83	0.78	279
maghreb-news	0.83	0.68	0.75	294
opinion	0.57	0.79	0.66	280
politics	0.44	0.21	0.29	84
society	0.75	0.06	0.12	48
accuracy			0.70	1392
macro avg	0.69	0.57	0.58	1392
weighted avg	0.71	0.70	0.69	1392

FIGURE 34 – Classification Report K-NN

	precision	recall	f1-score	support
covid-19	0.91	0.88	0.90	277
economy	0.74	0.85	0.79	130
education	0.88	0.90	0.89	279
maghreb-news	0.83	0.90	0.87	294
opinion	0.85	0.93	0.89	280
politics	0.82	0.48	0.60	84
society	0.73	0.23	0.35	48
accuracy			0.85	1392
macro avg	0.82	0.74	0.76	1392
weighted avg	0.85	0.85	0.84	1392

FIGURE 35 – Classification Report Régression Logistique

	precision	recall	f1-score	support
covid-19	0.88	0.92	0.90	277
economy	0.75	0.79	0.77	130
education	0.80	0.88	0.84	279
maghreb-news	0.83	0.88	0.86	294
opinion	0.84	0.92	0.88	280
politics	0.89	0.37	0.52	84
society	1.00	0.12	0.22	48
accuracy			0.83	1392
macro avg	0.85	0.70	0.71	1392
weighted avg	0.84	0.83	0.82	1392

FIGURE 36 – Classification Report Forêts Aléatoires

	precision	recall	f1-score	support
covid-19	0.92	0.89	0.90	277
economy	0.75	0.86	0.80	130
education	0.85	0.92	0.88	279
maghreb-news	0.85	0.90	0.88	294
opinion	0.87	0.91	0.89	280
politics	0.78	0.45	0.57	84
society	0.79	0.31	0.45	48
accuracy			0.85	1392
macro avg	0.83	0.75	0.77	1392
weighted avg	0.85	0.85	0.85	1392

FIGURE 37 – Classification Report Support Vector Machine

	precision	recall	f1-score	support
covid-19	0.93	0.87	0.90	277
economy	0.76	0.85	0.80	130
education	0.86	0.91	0.89	279
maghreb-news	0.85	0.90	0.87	294
opinion	0.87	0.91	0.89	280
politics	0.78	0.54	0.63	84
society	0.67	0.33	0.44	48
accuracy			0.85	1392
macro avg	0.81	0.76	0.78	1392
weighted avg	0.85	0.85	0.85	1392

FIGURE 38 – Classification Report Descente de Gradient Stochastique

	precision	recall	f1-score	support
covid-19	0.88	0.87	0.88	277
economy	0.74	0.79	0.76	130
education	0.80	0.86	0.83	279
maghreb-news	0.86	0.87	0.87	294
opinion	0.85	0.86	0.86	280
politics	0.75	0.56	0.64	84
society	0.67	0.46	0.54	48
accuracy			0.83	1392
macro avg	0.79	0.75	0.77	1392
weighted avg	0.83	0.83	0.83	1392

FIGURE 39 – Classification Report Perceptron Multicouche

10.13 Interprétation

On remarque que des modèles avec quasiment le même Accuracy Score peuvent en réalité avoir des résultats assez différents sur chacune des catégories séparément. De même, on remarque que certains modèles, même avec un Accuracy Score assez bas, peuvent avoir des très bonnes précisions sur une catégorie donnée.

Pour avoir une meilleure idée des erreurs de nos modèles et surtout des confusions qu'il peut faire, c'est-à-dire savoir quelle catégorie il prédit lorsqu'il se trompe et avec quelle fréquence, on va utiliser les matrices de confusions. Ces mesures détaillées par catégorie permettant de rendre compte de la performance d'un modèle de classification et peuvent ensuite permettre de comprendre pourquoi le modèle se trompe afin de l'améliorer.

		Réponse de l'expert	
		p	n
Réponse du classifier	Y	Vrai Positif	Faux Positif
	N	Faux Négatif	Vrai Négatif

FIGURE 40 – Matrice de Confusion

On peut facilement dire que la quantité de données d'entraînement joue un rôle important dans la qualité de classification. Les catégories avec peu d'articles comme **politics** et **society** sont les moins bien prédites contrairement à des catégories avec plus d'articles comme **opinion** et **education** qui sont les mieux prédits.

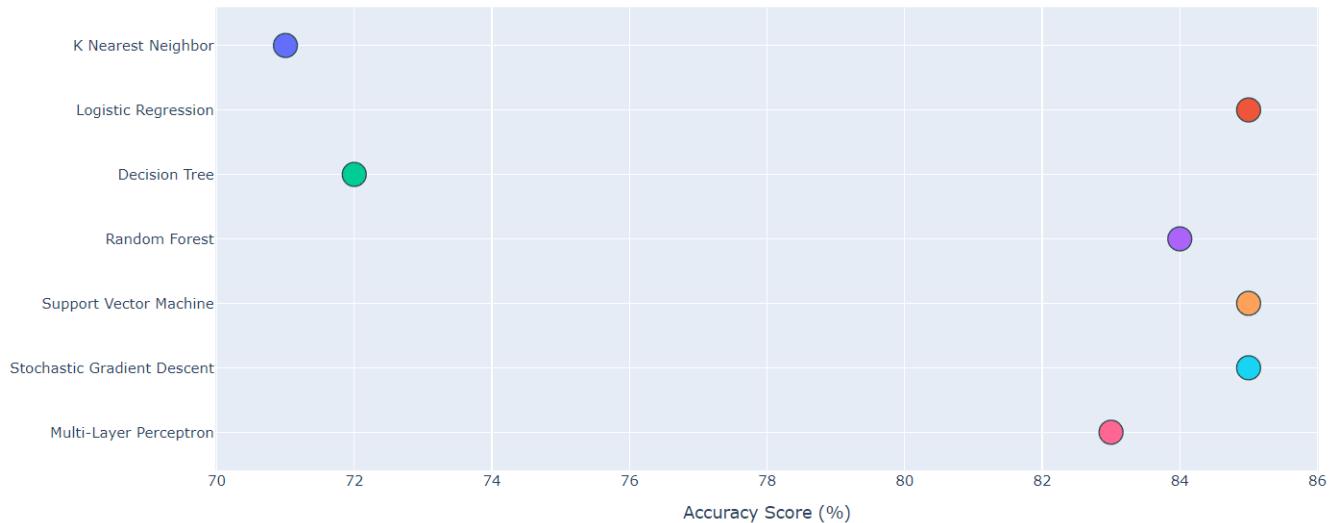


FIGURE 41 – Précision des différents modèles

10.14 Résultats Confusion Matrix

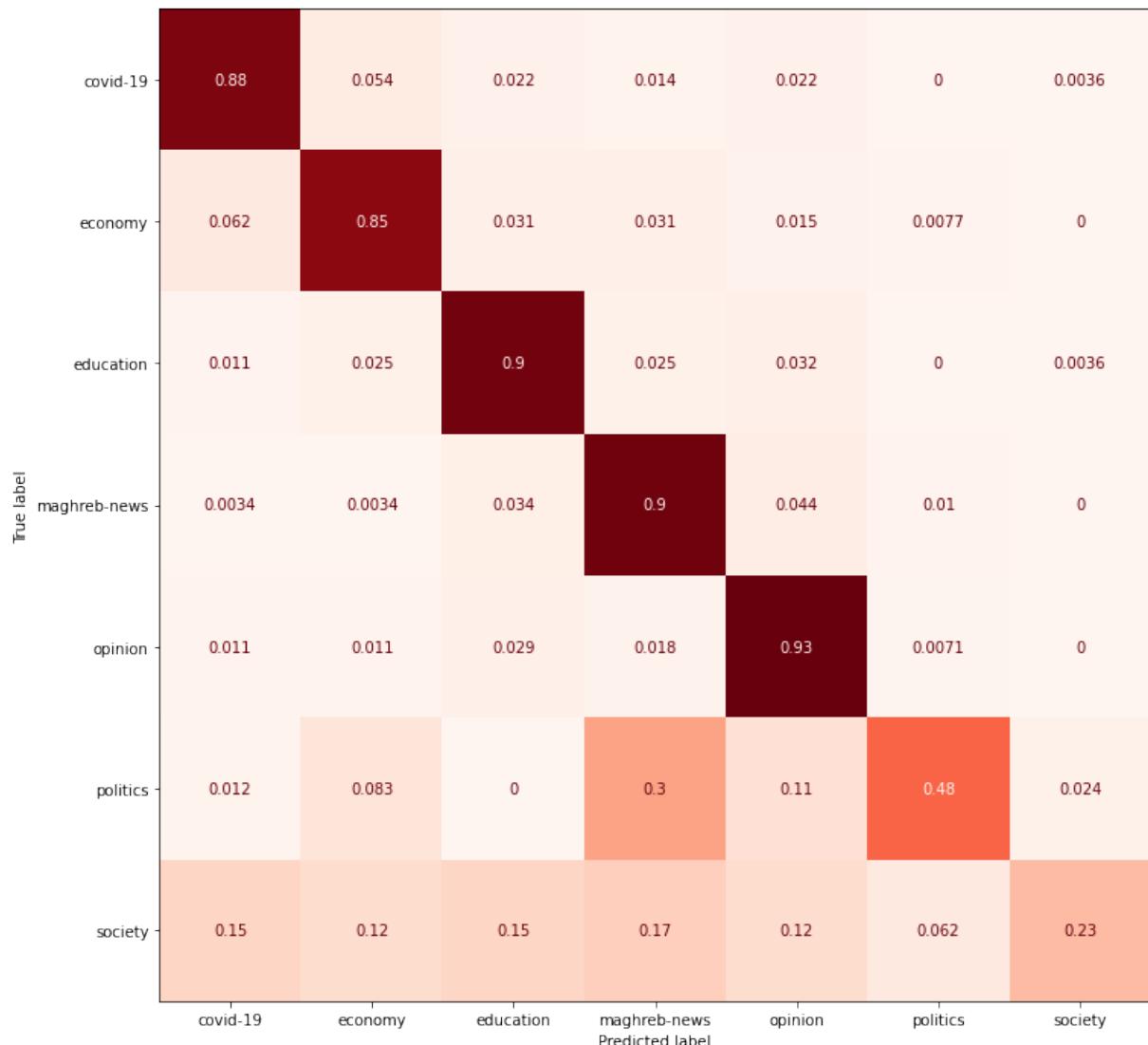


FIGURE 42 – Matrice de Confusion pour la Régression Logistique

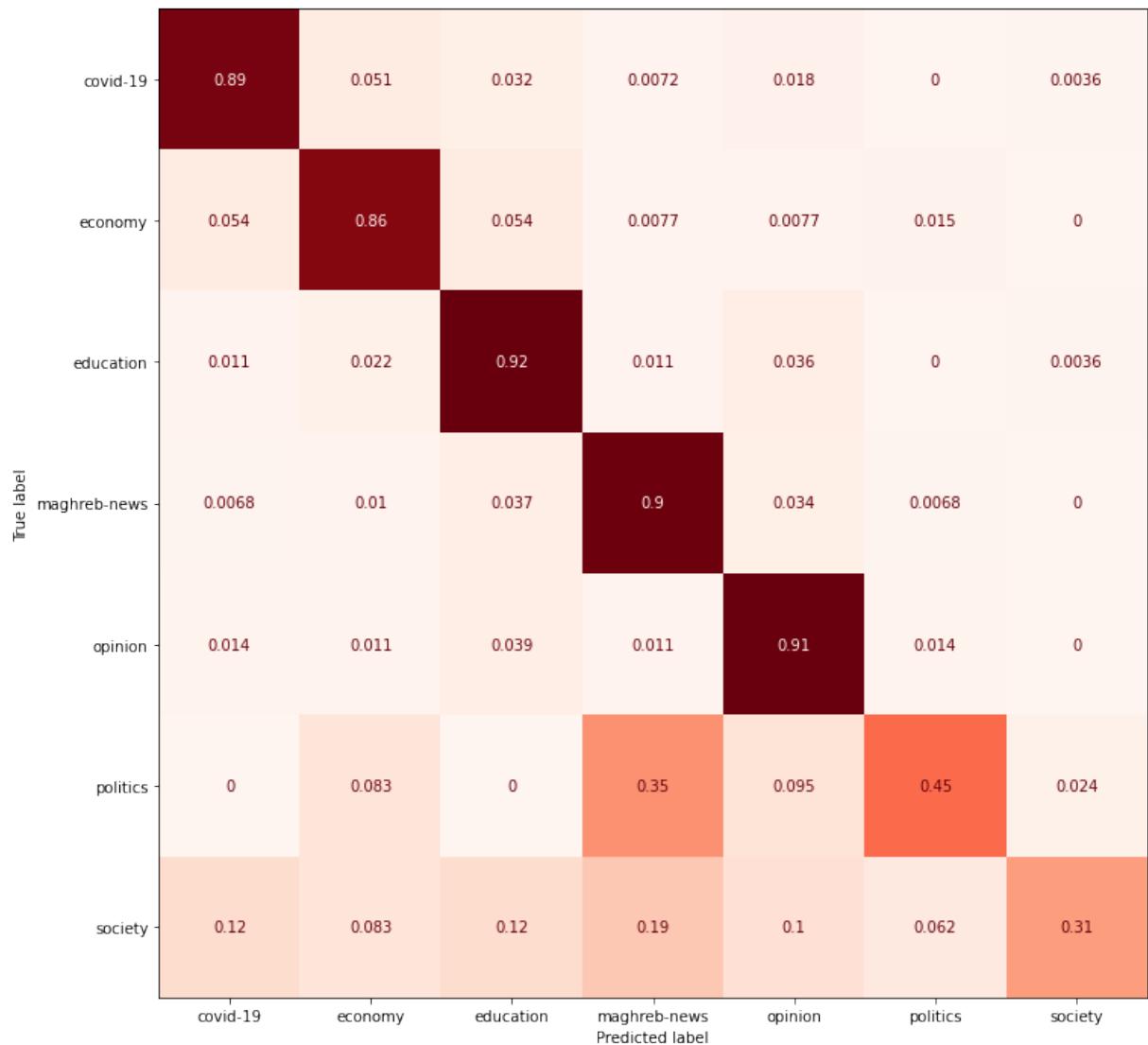


FIGURE 43 – Matrice de Confusion pour la Support Vector Machine

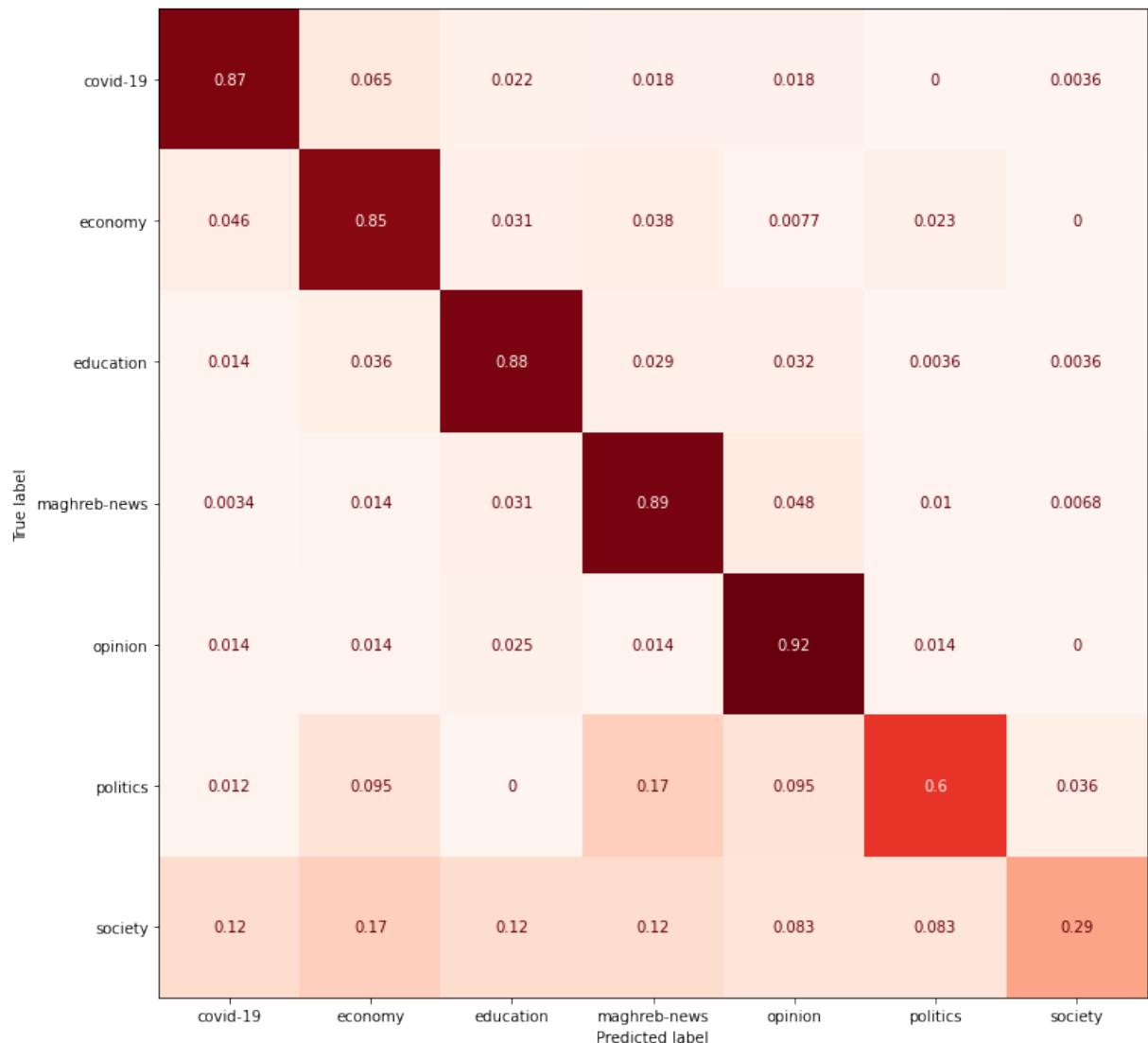


FIGURE 44 – Matrice de Confusion pour la Descente de Gradient Stochastique

11 DashBoard

11.1 Streamlit

La dernière tâche de mon stage à consister en la création d'une **application web** pour présenter visuellement les résultats obtenus. Cette application web devait contenir un échantillon du data set obtenu par **web scraping**, les résultats du **topic modeling**, la répartition des catégories et des auteurs, les scores des différents modèles de classification et enfin les matrices de confusion des trois meilleurs modèles.

La librairie utilisée est **Streamlit** qui permet de transformer les scripts en applications web partageables le tout en Python sans avoir besoin de technologies front-end. La librairie est très bien documentée et propose une *Cheat Sheet* très intuitive. Pour exécuter l'application il suffit d'utiliser un terminal et la commande **streamlit run** :

```
Terminal: Local + ▾
adib@LAPTOP-MKI2VFOP:/mnt/c/Users/Adib/Documents/Stage/Stage 1A/DashBoard$ streamlit run app.py
2022-07-07 12:22:39.146 INFO    numexpr.utils: Note: NumExpr detected 16 cores but "NUMEXPR_MAX_THREADS" not set, so enforcing safe limit of 8.
2022-07-07 12:22:39.147 INFO    numexpr.utils: NumExpr defaulting to 8 threads.

You can now view your Streamlit app in your browser.

Network URL: http://192.168.1.12:8501
External URL: http://197.247.248.135:8501
```

FIGURE 45 – Lancement de l'application Web

Afin de pouvoir tracer des graphiques et afficher des images sur le DashBoard j'ai utilisé respectivement **Plotly** et **PIL**. De plus j'ai eu besoin de **streamlit.components** pour ajouter le code HTML obtenu avec **pyLDAvis**.

11.2 Résultat

title	lead	author	date
0 Spanish FM: Spain Works on 'Constructive, Firm' Response to Algeria's Decision	Algeria decided to suspend its friendship treaty with Spain to further protest the country's endorsement of the Sahrawi cause.	Safaa Kasraoui	June 09, 2022 10:53 a.
1 Afro-Atlantic Treaty: Morocco Says Only Unity, Solidarity Can Save Africa	Morocco's Foreign Affairs Minister insists that only continental solidarity can save Africa from being a pawn in the global struggle.	Aya Benazizi	June 08, 2022 5:08 p.m.
2 Cape Verde to Open Consulate General in Morocco's Dakha	Togo also announced the forthcoming opening of a consulate in Dakha.	Safaa Kasraoui	June 08, 2022 4:40 p.m.
3 Mexican Delegation To Visit Morocco, Keen To Consolidate Relations	Migration management, the fight against climate change, and the Western Sahara dispute are expected to be key topics.	Souad Anouar	June 08, 2022 4:20 p.m.
4 Spanish PM Renews Support for Morocco's Autonomy Plan Amid Hostile Campaign	A number of marginal opposition political parties in Spain are taking issue with the Sanchez government's policies.	Safaa Kasraoui	June 08, 2022 2:02 p.m.
5 Moroccan Ambassador: Polisario's Sultan Khaya Is Not a Human Rights Defender	While presenting herself as a pacifist champion of human rights, Sultan Khaya has been recorded as supporting the Polisario Front.	Safaa Kasraoui	June 08, 2022 11:10 a.
6 Minister: Spain Wants to Preserve Ties with 'Reliable, Fraternal' Morocco	Since endorsing Morocco's Autonomy Plan for Western Sahara, Madrid has repeatedly renewed its support for the Polisario Front.	Safaa Kasraoui	June 07, 2022 5:02 p.m.
7 Saudi Arabia Engages in 'Serious Talks' With Israel For Potential Normalization	Many have suggested normalization of ties between Saudi Arabia and Israel is just a matter of time.	Safaa Kasraoui	June 07, 2022 2:14 p.m.

FIGURE 46 – Capture d'écran DashBoard

Topic Modeling using Latent Dirichlet Allocation

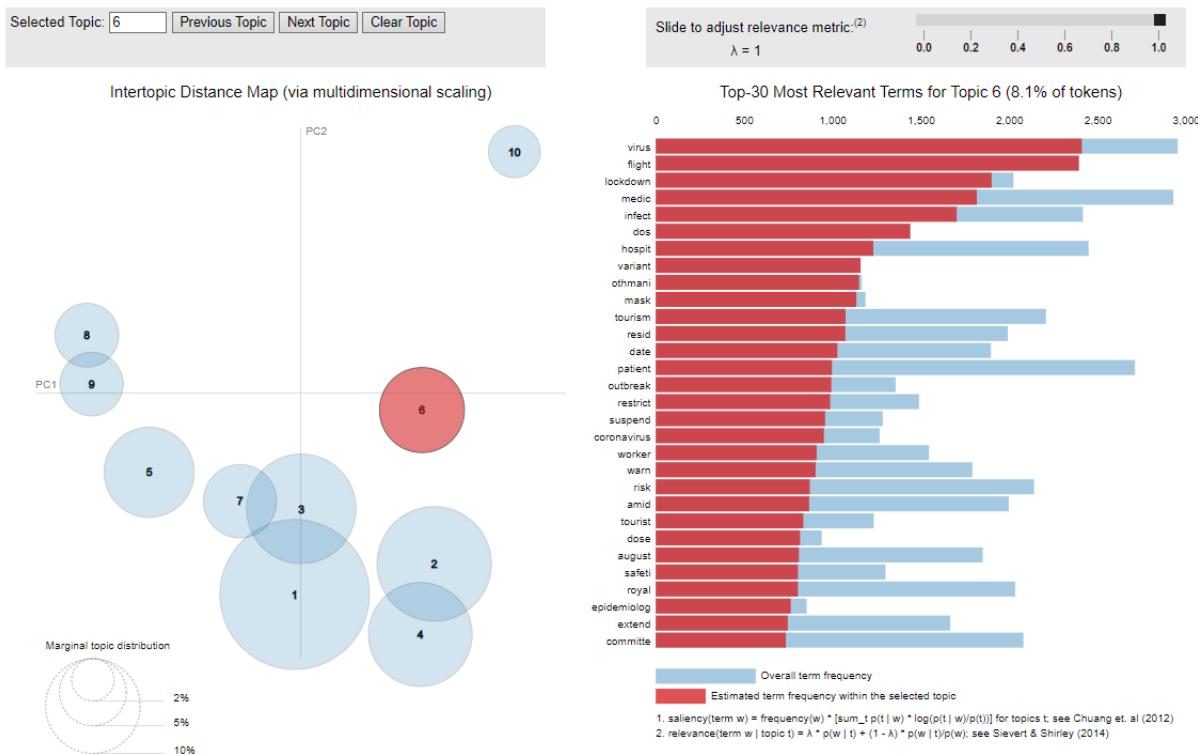


FIGURE 47 – Capture d'écran DashBoard

Morocco World News Categories

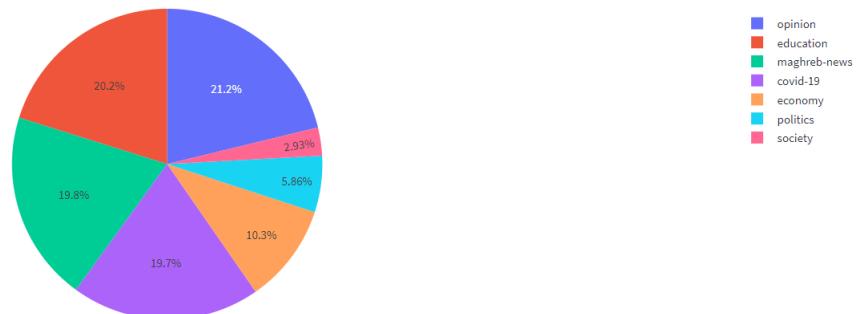


FIGURE 48 – Capture d'écran DashBoard

Morocco World News Authors

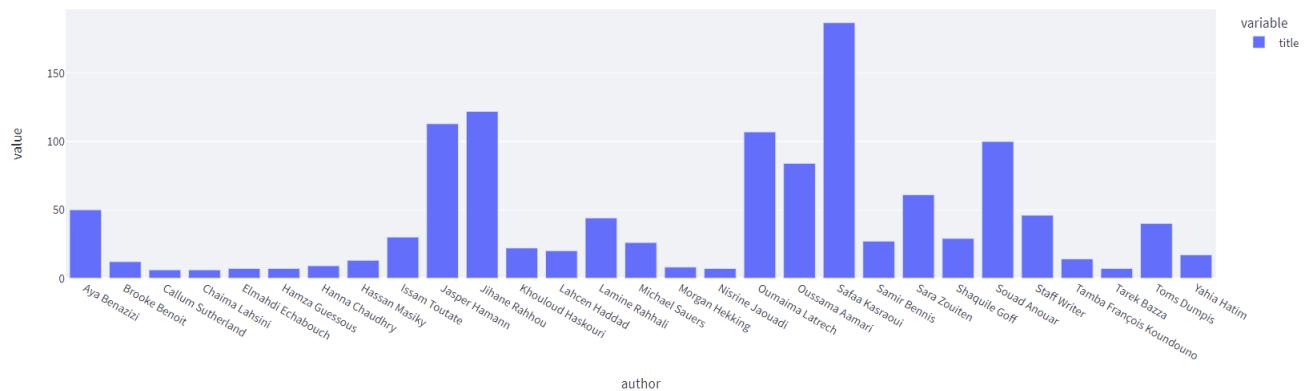


FIGURE 49 – Capture d'écran DashBoard

Classification Models Accuracies

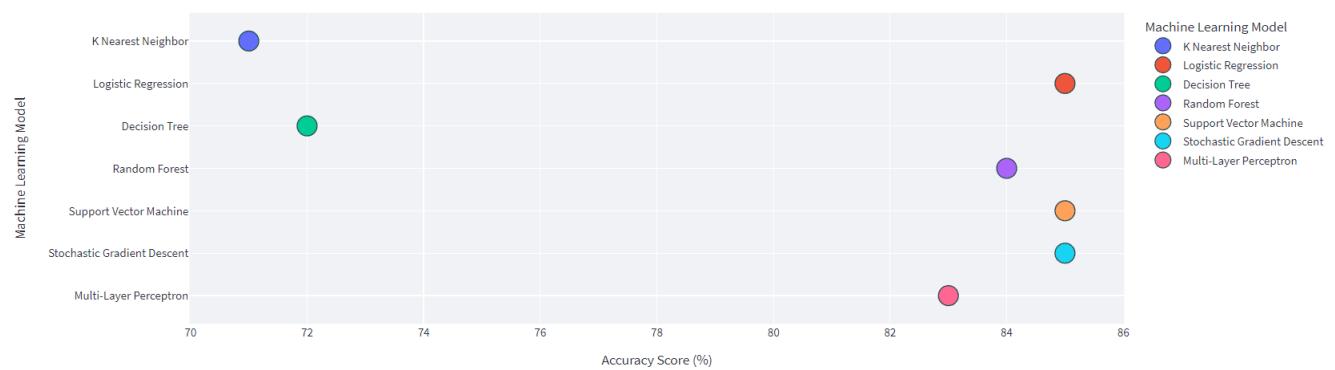


FIGURE 50 – Capture d'écran DashBoard

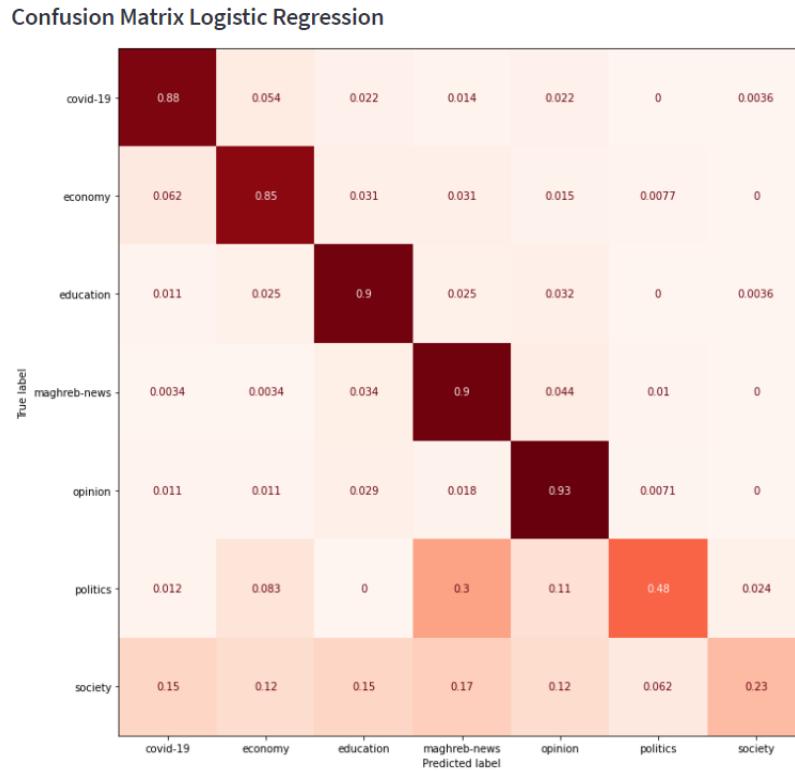


FIGURE 51 – Capture d'écran DashBoard

11.3 Configuration

D'autres librairies Python peuvent être utilisées à la place de **Streamlit** pour créer une application Web comme **Flask** ou encore **FastAPI**. Qui permettent un peu plus de liberté au niveau de la configuration que **Streamlit** :

```

app.py x config.toml x
[theme]
# Primary accent for interactive elements
primaryColor = '#E694FF'

# Background color for the main content area
backgroundColor = '#C2C2C2'

# Background color for sidebar and most interactive widgets
secondaryBackgroundColor = '#EBEBEB'

# Color used for almost all text
textColor = '#FFFFFF'

# Font family for all text in the app, except code blocks
# Accepted values (serif | sans serif | monospace)
# Default: "sans serif"
font = "sans serif"

```

FIGURE 52 – Fichier de configuration de Streamlit

12 Conclusion

Durant ce stage, le *Web-Scraping* a permis de récupérer de manière efficace des données sur les articles de presse en ligne marocaine. L'extraction des sujets principaux des articles grâce au *Topic Modeling* a donné des résultats très satisfaisants pour les articles en anglais et assez satisfaisants pour les articles en français. Enfin, la classification de texte par catégorie en utilisant différents modèles de *Machine Learning* a permis d'identifier les modèles les plus efficaces pour cette tâche à savoir la *Régression Logistique*, le *Stochastic Gradient Descent* et le *Support Vector Machine*.

Cependant, on se rend compte que sur certaines catégories les résultats ne sont pas satisfaisant à cause du nombre assez faible d'articles dans les données d'entraînement. Une amélioration envisageable pour la suite et la réalisation d'une campagne de *Web-Scraping* massif afin de récolter une dizaine de milliers d'articles par catégories. Ensuite, une exploration plus poussée des modèles cités précédemment s'imposent afin de les paramétrier de manière à ce qu'il soit le plus efficace possible pour notre cas d'usage.

La suite logique pour le *Haut-Commissariat au Plan* est de mettre en oeuvre à plus grande échelle toutes ces étapes en déployant une *pipeline* d'extraction des données, de nettoyage, d'application du *Topic Modeling* et de la *Text Classification* puis du monitoring des résultats via application Web.

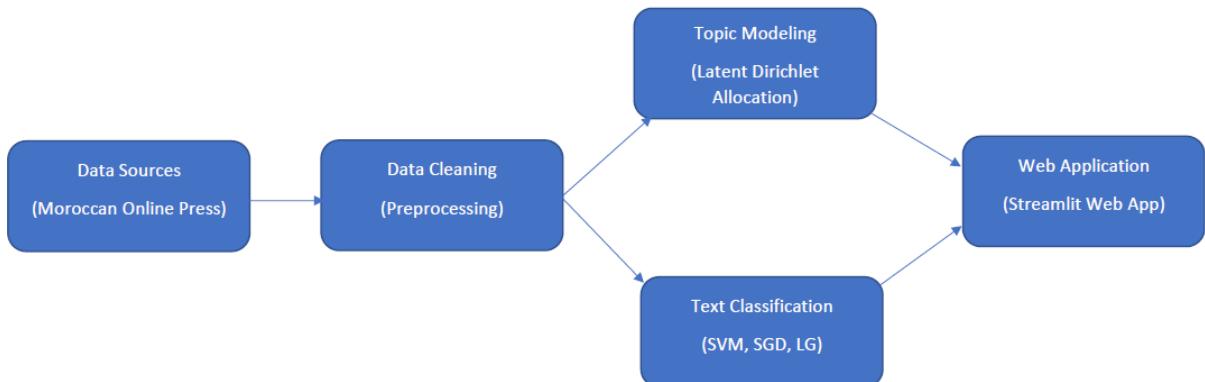


FIGURE 53 – Data Pipeline

L'ensemble des missions que j'ai effectuées durant mon stage ont permis au *HCP* d'étudier la faisabilité de l'ensemble du processus, d'identifier l'ensemble des technologies nécessaires (*Selenium*, *Nltk*, *Gensim*, *pyLDAvis*, *WordCloud*, *Plotly*, *Sklearn*, *Streamlit*) et aussi de sélectionner les modèles de *Machine Learning* les plus performants.

Dans son optique de développement, le *HCP* a effectué une refonte complète de son site mise en ligne durant mon stage le 5 juillet 2022, mes travaux rentrent en compte dans le *Nouveau Modèle de Développement* puisque le numérique figure en premier lieu dans les chantiers transformateurs. J'espère que dans les mois à venir, le projet auquel j'ai participé, qui n'est encore qu'à ses balbutiements, pourra voir le jour et que la science des données se retrouvera au cœur des outils de prises de décisions.

13 Bibliographie

- [1] Atindra Bandi, Medium, *Web Scraping Using Selenium — Python*, Lien vers l'article
- [2] Darek Tidwell, Medium, *Using Selenium with Google Colaboratory*, Lien vers l'article
- [3] Baiju Muthukadan, *Selenium Documentation*, Lien vers la documentation
- [4] Susan Li, Towards Data Science, *A Complete Exploratory Data Analysis and Visualization for Text Data*, Lien vers l'article
- [5] Michael Waskom, *Seaborn Documentation*, Lien vers la documentation
- [6] Priya Dwivedi, Towards Data Science, *NLP : Extracting the main topics from your dataset using LDA in minutes*, Lien vers l'article
- [7] Susan Li, Towards Data Science, *Topic Modelling in Python with NLTK and Gensim*, Lien vers l'article
- [8] NLTK Team, *NLTK Documentation*, Lien vers la documentation
- [9] Radim Řehůřek, *Gensim Documentation*, Lien vers la documentation
- [10] Ben Mabey, *pyLDAvis Documentation*, Lien vers la documentation
- [11] João Pedro, Towards Data Science, *Understanding Topic Coherence Measures*, Lien vers l'article
- [12] Usman Malik, Stack Abuse, *Text Classification with Python and Scikit-Learn*, Lien vers l'article
- [13] Deepak Singh, Analytics Vidhya, *Text Classification of News Articles*, Lien vers l'article
- [14] Scikit-Learn Team, *Scikit-Learn Classification Documentation*, Lien vers la documentation
- [15] Yohan C, DataScientest, *Qu'est ce que l'algorithme KNN ?*, Lien vers l'article
- [16] Chloé G, DataScientest, *Random Forest : Forêt d'arbre de décision- Définition et fonctionnement*, Lien vers l'article

- [17] Adrien R, DataScientest, *La régression logistique, qu'est-ce que c'est ?*, Lien vers l'article
- [18] Alban T, DataScientest, *SVM, quoi, comment, pourquoi ?*, Lien vers l'article
- [19] Wikipedia, *Stochastic Gradient Descent*, Lien vers l'article
- [20] Wikipedia, *Backpropagation*, Lien vers l'article
- [21] Wikipedia, *Multilayer Perceptron*, Lien vers l'article
- [22] Andrew Long, Towards Data Science, *Understanding Data Science Classification Metrics in Scikit-Learn in Python*, Lien vers l'article
- [23] Sarang Narkhede, Towards Data Science, *Understanding Confusion Matrix*, Lien vers l'article
- [24] Streamlit Inc, *Documentation Streamlit*, Lien vers la Cheat Sheet
- [25] Haut-Commissariat au Plan, *Publications*, Lien vers les publications
- [26] Haut-Commissariat au Plan, *Bases de Données Statistiques*, Lien vers les BDS

14 Glossaire

Web Scraping : Le Web Scraping est une technique d'extraction du contenu d'un site Web via un script ou un programme dans le but de transformer les données extraites afin de les utiliser dans d'autres contextes. [Page 1](#)

Natural Language Processing : Le traitement du langage naturel (en français) est un domaine interdisciplinaire qui comprend la linguistique, l'informatique et l'intelligence artificielle, dans le but de développer des outils de traitement du langage naturel pour une variété d'applications. [Page 4](#)

Topic Modeling : En apprentissage automatique et en traitement automatique du langage naturel, un modèle thématique est un modèle probabiliste permettant de déterminer des sujets ou thèmes abstraits dans un document. [Page 4](#)

Text Classification : La classification et catégorisation de documents est l'activité du traitement automatique du langage naturel qui consiste à classer de façon automatique des ressources documentaires. Ainsi la classification peut se faire par genre, par thème, ou encore par opinion. [Page 4](#)

Bases de Données Statistiques : Une base de données statistiques est une base de données classique qui retourne uniquement des informations et des données statistiques aux requêtes des utilisateurs, en fonction des enregistrements qui sont utilisée à des fins d'analyse statistique. [Page 7](#)

Latent Dirichlet Allocation : Dans le domaine du traitement automatique du langage naturel, l'allocation de Dirichlet latente est un modèle génératif probabiliste permettant d'expliquer des ensembles d'observations définis par des similarités de données par le moyen de groupes non observés. [Page 9](#)

Natural Language Toolkit : Développée par Steven Bird et Edward Loper du département informatique de l'Université de Pennsylvanie, la boîte à outils de traitement du langage naturel est une bibliothèque de logiciels Python pour le traitement automatique du langage. [Page 11](#)

Preprocessing : La manipulation ou à la suppression de données avant qu'elles ne soient utilisées afin d'assurer ou d'améliorer les performances, il constitue une étape importante dans le processus d'exploration de données. Son résultat est l'ensemble des données d'entraînement et de test. [Page 19](#)

DashBoard : Tableau de bord en français, est un outil permettant la visualisation de données de manière interactive ou non en faisant appel à différentes représentations visuelles, différents indicateurs et différentes hiérarchisations des données. [Page 26](#)

15 Annexes

15.1 Développement Durable et Responsabilité Sociétale

Dans une optique de responsabilité sociétale et environnementale le HCP s'est engagé sur plusieurs fronts. Il s'est donc doté d'une plateforme ODD spécialement conçue pour avoir un suivi de l'avancée de ces objectifs avec différents indicateurs basés sur des données disponibles pour tous.



FIGURE 54 – Capture d'écran du site ODD de l'HCP

En plus de mener ses propres études en interne, le HCP fournit aux différents organismes nationaux ainsi qu'aux bureaux d'études internationaux des données fiables et des indicateurs précis pour pouvoir mener leurs études autour de ses différents objectifs.



FIGURE 55 – Capture d'écran du site ODD de l'HCP

La plateforme ODD de suivi des objectifs de développement durable comprend 17 objectifs principaux distincts à différents horizons qui sont les suivants :

1. Éliminer la pauvreté sous toutes ses formes et partout dans le monde
2. Éliminer la faim, assurer la sécurité alimentaire, améliorer la nutrition et promouvoir l'agriculture durable
3. Permettre à tous de vivre en bonne santé et promouvoir le bien-être de tous à tout âge
4. Assurer à tous une éducation équitable, inclusive et de qualité et des possibilités d'apprentissage tout au long de la vie
5. Parvenir à l'égalité des sexes et autonomiser toutes les femmes et les filles
6. Garantir l'accès de tous à des services d'alimentation en eau et d'assainissement gérés de façon durable
7. Garantir l'accès de tous à des services énergétiques fiables, durables et modernes, à un coût abordable
8. Promouvoir une croissance économique soutenue, partagée et durable, le plein emploi productif et un travail décent pour tous
9. Bâtir une infrastructure résiliente, promouvoir une industrialisation durable qui profite à tous et encourager l'innovation
10. Réduire les inégalités dans les pays et d'un pays à l'autre
11. Faire en sorte que les villes et les établissements humains soient ouverts à tous, sûrs, résilients et durables
12. Établir des modes de consommation et de production durables
13. Prendre d'urgence des mesures pour lutter contre les changements climatiques et leurs répercussions
14. Conserver et exploiter de manière durable les océans, les mers et les ressources marines aux fins du développement durable
15. Préserver et restaurer les écosystèmes terrestres, en veillant à les exploiter de façon durable, gérer durablement les forêts, lutter contre la désertification, enrayer et inverser le processus de dégradation des terres et mettre fin à l'appauvrissement de la biodiversité
16. Promouvoir l'avènement de sociétés pacifiques et inclusives aux fins du développement durable, assurer l'accès de tous à la justice et mettre en place, à tous les niveaux, des institutions efficaces, responsables et ouvertes à tous
17. Renforcer les moyens de mettre en œuvre le Partenariat mondial pour le développement durable et le revitaliser

À ce sujet, le HCP a publié deux rapports nationaux sur les objectifs du développement durable au Maroc. Ils sont tous les deux téléchargeables aux liens suivants :

Rapport National 2021 : Les objectifs du développement durable au Maroc dans le contexte de la Covid-19 (Version française)

Rapport National 2020 sur la mise en œuvre par le Royaume du Maroc des Objectifs de Développement Durable (Version Française)

15.2 Annexe A : Extrait Code Web Scraping

Installation du package selenium et du chromedriver

```
# installation du package selenium
!pip install selenium
# installation du chromedriver
!apt install chromium-chromedriver
# copie du chromedriver
!cp C:\Users\Adib\Documents\Stage\Stage 1A\Selenium\chromedriver.exe
```

```
Reading package lists... Done
Building dependency tree
Reading state information... Done
chromium-chromedriver is already the newest version (101.0.4951.64-0ubuntu0.18.04.1).
The following package was automatically installed and is no longer required:
  libnvidia-common-460
Use 'apt autoremove' to remove it.
0 upgraded, 0 newly installed, 0 to remove and 45 not upgraded.
cp: cannot stat 'C:UsersAdibDocumentsStageStage': No such file or directory
```

Importation des librairies nécessaires

```
from selenium import webdriver
import pandas as pd
```

Modification des options du chromedriver

```
# options chromedriver pour l'utiliser dans un notebook
options = webdriver.ChromeOptions()
options.add_argument('--headless')
options.add_argument('--no-sandbox')
options.add_argument('--disable-dev-shm-usage')
```

Instanciation du WebDriver de Chrome

```
# création de l'instance WebDriver de Chrome
driver = webdriver.Chrome('chromedriver', options = options)
# Lien du site Morocco World News
website = 'https://www.moroccoworldnews.com/'
```

Liste des catégories du site

```
# catégories du site Morocco World News
categories = ['politics', 'economy', 'opinion', 'lifestyle', 'features',
```

Récupération des liens d'articles

```
# Liste des liens vers les articles
articles_links = []
# parcours des catégories
for category in categories:
    category_link = website + category
    # parcours des pages de chaque catégorie
    page_number = 0
    while (page_number < 10):
        webpage_link = category_link + '/' + str(page_number)
        page_number += 1
        # parcours des articles de chaque page
        driver.get(webpage_link)
        href_links = driver.find_elements_by_xpath('//h3/a[@href]')
        for href_link in href_links:
            articles_links.append(href_link.get_attribute('href'))
```

Récupération du contenu des articles

```
# data frame des données collectés
df_articles = pd.DataFrame()
# parcours des liens collectés
for article_link in articles_links[:500]:
    # accès à la page de l'article
    driver.get(article_link)
    # récupération du titre de l'article
    title = driver.find_element_by_id('title').text
    # récupération de l'auteur de l'article
    author = driver.find_element_by_class_name('author').text.split(',') [0]
    # récupération de la date de publication de l'article
    date = driver.find_element_by_xpath('//time').text
    # récupération du contenu de l'article
    content_scrap = driver.find_elements_by_xpath('//div/p')
    content = [p.text for p in content_scrap if len(p.text) > 50]
    content = ' '.join(content)
    # création de la liste de nos données
    article = {'title' : title, 'author' : author, 'date' : date, 'content' : content}
    # transformation en data frame
    df_article = pd.DataFrame(article, index = [0])
    # transformation en data frame
    df_articles = df_articles.append(df_article, ignore_index=True)
```

15.3 Annexe B : Extrait Code WordCloud

Importation des libraires nécessaires

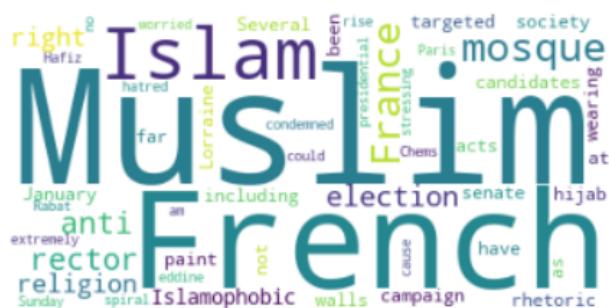
```
from wordcloud import WordCloud  
import matplotlib.pyplot as plt  
import pandas as pd
```

Importation des données textuelles

```
morocco_world_news = pd.read_csv('morocco_world_news_articles.csv')  
  
morocco_world_news_text = morocco_world_news['content']  
  
le_matin_articles = pd.read_csv('le_matin_articles.csv')  
  
le_matin_articles_text = le_matin_articles['content']
```

Génération des nuages de mots

```
morocco_world_news_wordcloud = WordCloud(background_color = 'white',  
                                         stopwords = stop_words,  
                                         max_words = 50).generate(morocco_world_news_text[90])  
plt.imshow(morocco_world_news_wordcloud)  
plt.axis("off")  
plt.show();
```



```
morocco_world_news_wordcloud.to_file('article_1.png')
```

15.4 Annexe C : Extrait Code Topic Modeling

Importation des données

```
import pandas as pd

morocco_world_news = pd.read_csv('morocco_world_news_articles.csv', engine='python', error_bad_lines=False)

morocco_world_news.head()

  Unnamed: 0  category          content
0            0    politics  Rabat - A confidential report from NATO has ex...
1            1    politics  Rabat - Top security officials from Morocco an...
2            2    politics  Rabat - Indie-rock band Big Thief has announce...
3            3    politics  Rabat - The European Union has called on Alger...
4            4    politics  Rabat - Spain regrets Algeria's decision to su...
```

Installation des packages

```
!pip install nltk
!pip install gensim
```

Importation des librairies

```
import numpy as np

import gensim
from gensim.utils import simple_preprocess
from gensim.parsing.preprocessing import STOPWORDS
from gensim.models import CoherenceModel

import nltk
from nltk.stem import WordNetLemmatizer, SnowballStemmer
from nltk.stem.porter import *
nltk.download('wordnet')
nltk.download('omw-1.4')
```

Fonction de preprocessing

- on supprime tous les stopwords (this, that, where...)
- on supprime les mots de moins de 3 lettres
- on applique la lemmatisation

```
stemmer = SnowballStemmer('english')

def lemmatize_stemming(text) :
    return stemmer.stem(WordNetLemmatizer().lemmatize(text, pos='n'))

def preprocess(text) :
    result = []
    for token in gensim.utils.simple_preprocess(text) :
        if token not in gensim.parsing.preprocessing.STOPWORDS and len(token) > 3 :
            result.append(lemmatize_stemming(token))
    return result
```

Preprocessing des données

- on supprime les valeurs NAN de nos données
- on applique la fonction de preprocessing

```
morocco_world_news.dropna(subset = ["content"], inplace=True)

processed_docs = [preprocess(doc) for doc in morocco_world_news['content']]

processed_docs[10][:10]

['rabat',
 'spanish',
 'interior',
 'minist',
 'fernando',
 'grand',
 'marlaska',
 'reiter',
 'countri',
 'commit']
```

Stockage des données après preprocessing

- on utilise un dictionnaire qui contient le mot comme clé et son nombre d'occurrences comme valeur

```
dictionary = gensim.corpora.Dictionary(processed_docs)
```

Nettoyage du dictionnaire

- on supprime les mots trop rares qui apparaissent moins de 15 fois
- on supprime les mots trop fréquents qui apparaissent dans plus de 10% des documents
- à la fin on ne garde que les 100 000 mots les plus fréquents

```
dictionary.filter_extremes(no_below=15, no_above=0.1, keep_n=100000)
```

Conversion en Bag-Of-Words

- on convertit notre dictionnaire en couple mot et nombres d'occurrences : format bag-of-words

```
bow_corpus = [dictionary.doc2bow(doc) for doc in processed_docs]
```

```
for i in range(10) :
    print("Word {} ('{}') appears {} time.".format(bow_corpus[10][i][0], dictionary[bow_corpus[10][i][0]])
```

```
Word 6 ("amid") appears 1 time.
Word 8 ("autonomi") appears 1 time.
Word 10 ("basi") appears 1 time.
Word 22 ("credibl") appears 1 time.
Word 25 ("disput") appears 1 time.
Word 27 ("endors") appears 1 time.
Word 38 ("immedi") appears 1 time.
Word 43 ("madrid") appears 3 time.
Word 47 ("outlet") appears 1 time.
Word 60 ("sanchez") appears 3 time.
```

Exécution du LDA

- LdaMulticore pour utiliser tout les coeurs du CPU afin de gagner en temps d'exécution
- num_topics : nombre de topic à extraire du corpus
- id2word : mapping des identifiants de mots (entiers) aux mots (chaînes de caractères)
- passes : nombre d'itération d'entraînement sur le corpus

```
lda_model = gensim.models.LdaMulticore(bow_corpus, num_topics = 10, id2word = dictionary, passes = 1000)
```

```
topics = []
for idx, topic in lda_model.print_topics(-1):
    print("Topic: {} > Words: {}".format(idx, topic))
    topics.append(topic)
```

Cohérence du topic model

- Les mesures de cohérence évaluent le degré de similitude sémantique entre les mots les mieux notés dans le topics
- Ces mesures aident à faire la distinction entre les topics sémantiquement interprétables et les topics dû à des inférences statistiques
- Pour un bon modèle LDA la cohérence doit être comprise entre 0.4 et 0.7 au delà le modèle est probablement erroné

```
coherence_model_lda = CoherenceModel(model=lda_model, texts=processed_docs, dictionary=dictionary)
coherence_lda = coherence_model_lda.get_coherence()
print('\nCoherence Score: ', coherence_lda)
```

Coherence Score: 0.5129831809687769

Stockage des résultats

```
all_topic_model = []
for i in range(len(topics)):
    str = topics[i].split(' + ')
    topic_model = []
    for j in range(10):
        weight = str[j][0:5]
        word = str[j][7:len(str[j])-1]
        topic_model.append((weight, word))
    all_topic_model.append(topic_model)
```

Visualisation des résultats

```
!pip install pyLDAvis
```

```
import pyLDAvis.gensim_models
```

```
pyLDAvis.enable_notebook()
pyLDAvis.gensim_models.prepare(lda_model, bow_corpus, dictionary)
```

15.5 Annexe D : Extrait Code Text Classification

Importation des librairies nécessaires

```
import pandas as pd
import numpy as np
import pickle
import re
```

```
import nltk
nltk.download('wordnet')
nltk.download('omw-1.4')
nltk.download('stopwords')
from nltk.corpus import stopwords
from nltk.stem import WordNetLemmatizer
```

```
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import TfidfTransformer
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.naive_bayes import MultinomialNB, GaussianNB
from sklearn.tree import DecisionTreeClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.svm import SVC
from sklearn.metrics import classification_report, accuracy_score
```

Importation des données

```
data = pd.read_csv('morocco_world_news_articles.csv')
X, y = data.content, data.category
data.drop(columns=['Unnamed: 0'], inplace = True)
data.dropna(inplace = True)
```

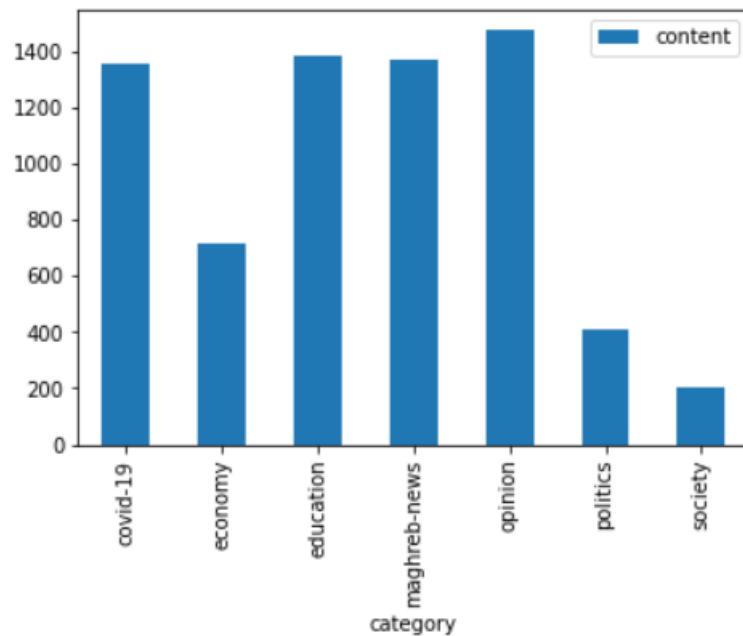
Visualisation des catégories

```
data["category"].value_counts()
```

```
opinion      1476
education    1390
maghreb-news 1370
covid-19     1359
economy      715
politics     407
society       203
Name: category, dtype: int64
```

```
data.groupby('category').count().plot.bar(ylim=0)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fb13e6b0cd0>
```



Preprocessing des données

```
stemmer = WordNetLemmatizer()

documents = []

for sen in range(0, len(X)):
    # suppression des caractères spéciaux
    document = re.sub(r'\W', ' ', str(X[sen]))
    # suppression des caractères uniques
    document = re.sub(r'\s+[a-zA-Z]\s+', ' ', document)
    document = re.sub(r'^[a-zA-Z]\s+', ' ', document)
    # suppression des espaces multiples
    document = re.sub(r'\s+', ' ', document, flags=re.I)
    # suppression des préfixes b
    document = re.sub(r'^b\s+', ' ', document)
    # concersion en minuscule
    document = document.lower()
    # Lemmatisation
    document = document.split()
    document = [stemmer.lemmatize(word) for word in document]
    document = ' '.join(document)
    # ajout à la liste
    documents.append(document)
```

Bag Of Words

```
# conversion des données textuelles en données numériques
vectorizer = CountVectorizer(max_features=1500, min_df=5, max_df=0.7, stop_words=stopwords.words('english'))
X = vectorizer.fit_transform(documents).toarray()
```

TF - IDF

```
# Term Frequency = Nombre d'occurrences d'un mot / Nombre total des mots dans le document
# Inverse Document Frequency = Log(Nombre total de documents / Nombre de documents contenant le mot)
tfidfconverter = TfidfTransformer()
X = tfidfconverter.fit_transform(X).toarray()
```

Division des données

```
# division des données en données d'entraînement (80%) et données de test (20%)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=0)
```

K Nearest Neighbour

Entraînement du modèle

```
# utilisation de l'algorithme des k plus proches voisins sur nos données d'entraînement
model_knn = KNeighborsClassifier(n_neighbors=10, metric='minkowski', p=4)
model_knn.fit(X_train, y_train)
```

```
# prédiction à partir de nos données de test
y_pred = model_knn.predict(X_test)
```

Evaluation du modèle

```
# évaluation du score globale du modèle
print(accuracy_score(y_test, y_pred))
```

0.7047413793103449

```
# évaluation de la précision : taux de prédictions positives correctes
# évaluation du recall : taux de positifs correctement prédits
# évaluation du f1-score : capacité d'un modèle à bien prédire les individus positifs
print(classification_report(y_test, y_pred))
```

	precision	recall	f1-score	support
covid-19	0.80	0.82	0.81	277
economy	0.69	0.62	0.65	130
education	0.73	0.83	0.78	279
maghreb-news	0.83	0.68	0.75	294
opinion	0.57	0.79	0.66	280
politics	0.44	0.21	0.29	84
society	0.75	0.06	0.12	48
accuracy			0.70	1392
macro avg	0.69	0.57	0.58	1392
weighted avg	0.71	0.70	0.69	1392

Arbre de décision

Entraînement du modèle

```
# utilisation d'arbres de décision sur nos données d'entraînement
model_decisiontree = DecisionTreeClassifier()
model_decisiontree.fit(X_train, y_train)
```

```
# prédiction à partir de nos données de test
y_pred = model_decisiontree.predict(X_test)
```

Evaluation du modèle

```
# évaluation du score globale du modèle
print(accuracy_score(y_test, y_pred))
```

```
0.7205459770114943
```

```
# évaluation de la précision : taux de prédictions positives correctes
# évaluation du recall : taux de positifs correctement prédicts
# évaluation du f1-score : capacité d'un modèle à bien prédire les individus positifs
print(classification_report(y_test, y_pred))
```

	precision	recall	f1-score	support
covid-19	0.85	0.83	0.84	277
economy	0.56	0.63	0.59	130
education	0.72	0.78	0.75	279
maghreb-news	0.77	0.76	0.77	294
opinion	0.76	0.74	0.75	280
politics	0.40	0.35	0.37	84
society	0.34	0.27	0.30	48
accuracy			0.72	1392
macro avg	0.63	0.62	0.62	1392
weighted avg	0.72	0.72	0.72	1392

Forêts Aléatoires

Entraînement du modèle

```
# utilisation de l'algorithme des forêts aléatoires sur nos données d'entraînement
model_randomforest = RandomForestClassifier(n_estimators=1000, random_state=0)
model_randomforest.fit(X_train, y_train)
```

```
# prédiction à partir de nos données de test
y_pred = model_randomforest.predict(X_test)
```

Evaluation du modèle

```
# évaluation du score globale du modèle
print(accuracy_score(y_test, y_pred))
```

```
0.8304597701149425
```

```
# évaluation de la précision : taux de prédictions positives correctes
# évaluation du recall : taux de positifs correctement prédits
# évaluation du f1-score : capacité d'un modèle à bien prédire les individus positifs
print(classification_report(y_test, y_pred))
```

	precision	recall	f1-score	support
covid-19	0.88	0.92	0.90	277
economy	0.75	0.79	0.77	130
education	0.80	0.88	0.84	279
maghreb-news	0.83	0.88	0.86	294
opinion	0.84	0.92	0.88	280
politics	0.89	0.37	0.52	84
society	1.00	0.12	0.22	48
accuracy			0.83	1392
macro avg	0.85	0.70	0.71	1392
weighted avg	0.84	0.83	0.82	1392

Régression Logistique

Entraînement du modèle

```
# utilisation de la régression logistique sur nos données d'entraînement
model_logisticregression = LogisticRegression(random_state=0)
model_logisticregression.fit(X_train, y_train)
```

```
# prédiction à partir de nos données de test
y_pred = model_logisticregression.predict(X_test)
```

Evaluation du modèle

```
# évaluation du score globale du modèle
print(accuracy_score(y_test, y_pred))
```

```
0.8505747126436781
```

```
# évaluation de la précision : taux de prédictions positives correctes
# évaluation du recall : taux de positifs correctement prédits
# évaluation du f1-score : capacité d'un modèle à bien prédire les individus positifs
print(classification_report(y_test, y_pred))
```

	precision	recall	f1-score	support
covid-19	0.91	0.88	0.90	277
economy	0.74	0.85	0.79	130
education	0.88	0.90	0.89	279
maghreb-news	0.83	0.90	0.87	294
opinion	0.85	0.93	0.89	280
politics	0.82	0.48	0.60	84
society	0.73	0.23	0.35	48
accuracy			0.85	1392
macro avg	0.82	0.74	0.76	1392
weighted avg	0.85	0.85	0.84	1392

Machine à vecteurs de support

Entraînement du modèle

```
# utilisation de la machine à vecteurs de support sur nos données d'entraînement
model_svc = SVC()
model_svc.fit(X_train, y_train)
```

```
# prédiction à partir de nos données de test
y_pred = model_svc.predict(X_test)
```

Evaluation du modèle

```
# évaluation du score globale du modèle
print(accuracy_score(y_test, y_pred))
```

```
0.853448275862069
```

```
# évaluation de la précision : taux de prédictions positives correctes
# évaluation du recall : taux de positifs correctement prédits
# évaluation du f1-score : capacité d'un modèle à bien prédire les individus positifs
print(classification_report(y_test, y_pred))
```

	precision	recall	f1-score	support
covid-19	0.92	0.89	0.90	277
economy	0.75	0.86	0.80	130
education	0.85	0.92	0.88	279
maghreb-news	0.85	0.90	0.88	294
opinion	0.87	0.91	0.89	280
politics	0.78	0.45	0.57	84
society	0.79	0.31	0.45	48
accuracy			0.85	1392
macro avg	0.83	0.75	0.77	1392
weighted avg	0.85	0.85	0.85	1392

Gradient Stochastique

Entraînement du modèle

```
# utilisation de l'algorithme du gradient stochastique sur nos données d'entraînement
model_sgd = SGDClassifier()
model_sgd.fit(X_train, y_train)
```

```
# prédition à partir de nos données de test
y_pred = model_sgd.predict(X_test)
```

Evaluation du modèle

```
# évaluation du score globale du modèle
print(accuracy_score(y_test, y_pred))
```

```
0.853448275862069
```

```
# évaluation de la précision : taux de prédictions positives correctes
# évaluation du recall : taux de positifs correctement prédits
# évaluation du f1-score : capacité d'un modèle à bien prédire les individus positifs
print(classification_report(y_test, y_pred))
```

	precision	recall	f1-score	support
covid-19	0.93	0.87	0.90	277
economy	0.76	0.85	0.80	130
education	0.86	0.91	0.89	279
maghreb-news	0.85	0.90	0.87	294
opinion	0.87	0.91	0.89	280
politics	0.78	0.54	0.63	84
society	0.67	0.33	0.44	48
accuracy			0.85	1392
macro avg	0.81	0.76	0.78	1392
weighted avg	0.85	0.85	0.85	1392

Perceptron Multicouche

Entraînement du modèle

```
# utilisation d'un perceptron multicouche sur nos données d'entraînement
model_mlp = MLPClassifier()
model_mlp.fit(X_train, y_train)
```

```
# prédiction à partir de nos données de test
y_pred = model_mlp.predict(X_test)
```

Evaluation du modèle

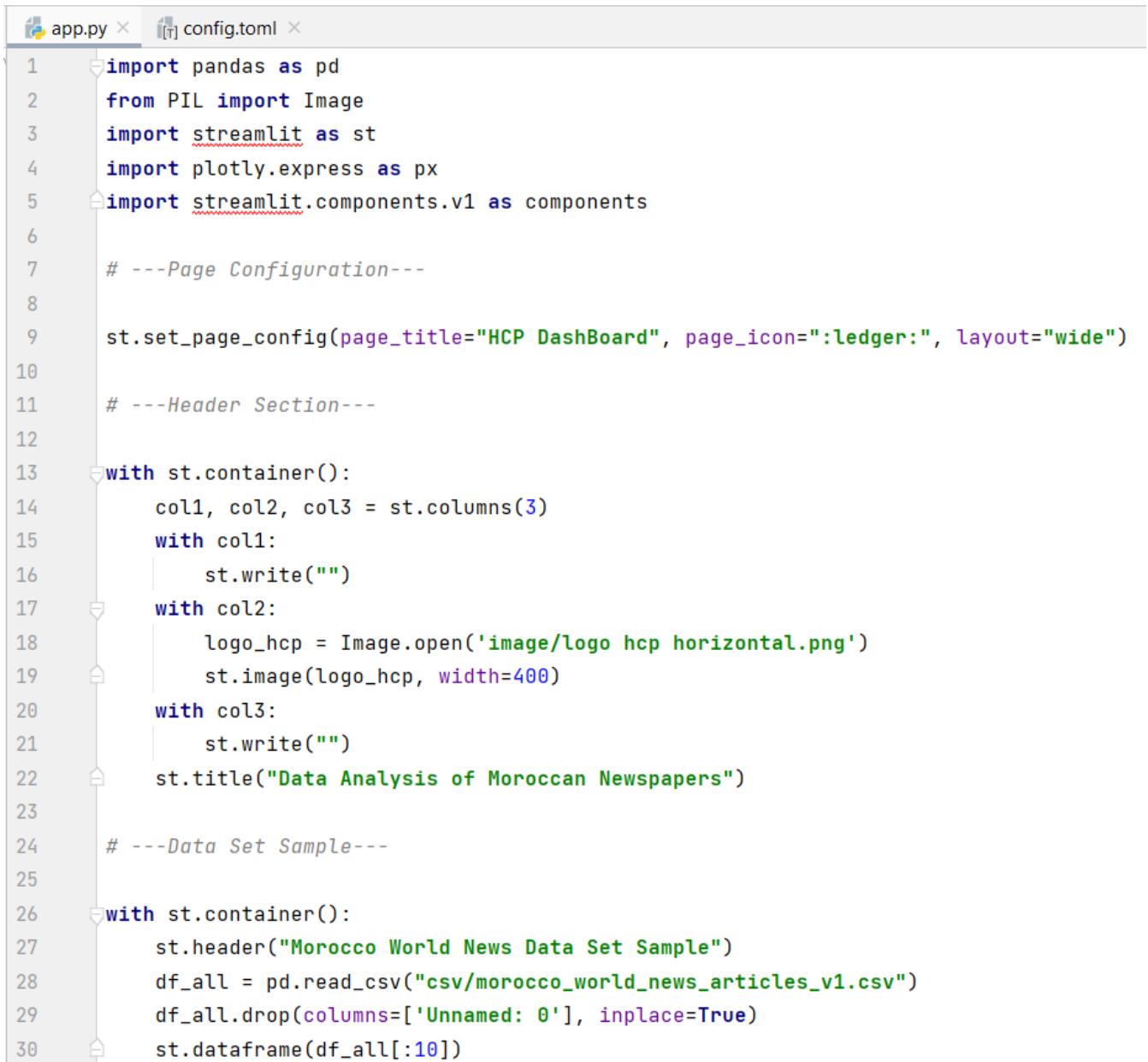
```
# évaluation du score globale du modèle
print(accuracy_score(y_test, y_pred))
```

0.8283045977011494

```
# évaluation de la précision : taux de prédictions positives correctes
# évaluation du recall : taux de positifs correctement prédis
# évaluation du f1-score : capacité d'un modèle à bien prédire les individus positifs
print(classification_report(y_test, y_pred))
```

	precision	recall	f1-score	support
covid-19	0.88	0.87	0.88	277
economy	0.74	0.79	0.76	130
education	0.80	0.86	0.83	279
maghreb-news	0.86	0.87	0.87	294
opinion	0.85	0.86	0.86	280
politics	0.75	0.56	0.64	84
society	0.67	0.46	0.54	48
accuracy			0.83	1392
macro avg	0.79	0.75	0.77	1392
weighted avg	0.83	0.83	0.83	1392

15.6 Annexe E : Extrait Code DashBoard



The screenshot shows a code editor with two tabs at the top: "app.py" and "config.toml". The "app.py" tab is active and displays the following Python code:

```
1 import pandas as pd
2 from PIL import Image
3 import streamlit as st
4 import plotly.express as px
5 import streamlit.components.v1 as components
6
7 # ---Page Configuration---
8
9 st.set_page_config(page_title="HCP Dashboard", page_icon=":ledger:", layout="wide")
10
11 # ---Header Section---
12
13 with st.container():
14     col1, col2, col3 = st.columns(3)
15     with col1:
16         st.write("")
17     with col2:
18         logo_hcp = Image.open('image/logo_hcp_horizontal.png')
19         st.image(logo_hcp, width=400)
20     with col3:
21         st.write("")
22     st.title("Data Analysis of Moroccan Newspapers")
23
24 # ---Data Set Sample---
25
26 with st.container():
27     st.header("Morocco World News Data Set Sample")
28     df_all = pd.read_csv("csv/morocco_world_news_articles_v1.csv")
29     df_all.drop(columns=['Unnamed: 0'], inplace=True)
30     st.dataframe(df_all[:10])
```

```
app.py x config.toml x
31
32     # ---Topic Model Vis---
33
34     with st.container():
35         st.header("Topic Modeling using Latent Dirichlet Allocation")
36         lda = open("html/lda.html", 'r', encoding='utf-8')
37         source_code = lda.read()
38         components.html(source_code, width=1700, height=800)
39
40     # ---News Categories---
41
42     with st.container():
43         st.header("Morocco World News Categories")
44         df = pd.read_csv("csv/morocco_world_news_articles.csv")
45         fig = px.pie(df, names="category")
46         st.plotly_chart(fig, use_container_width=True)
47
48     # ---News Authors---
49
50     with st.container():
51         st.header("Morocco World News Authors")
52         author = df_all.groupby('author').count()
53         author.drop(author[author.title < 6].index, inplace=True)
54         author.drop(["lead", "date", "content"], axis=1, inplace=True)
55         fig = px.bar(author)
56         st.plotly_chart(fig, use_container_width=True)
```

```
app.py x config.toml x
58     # ---Classification Accuracy---
59
60     with st.container():
61         st.header("Classification Models Accuracies")
62         acc = pd.DataFrame(
63             ['K Nearest Neighbor', 'Logistic Regression', 'Decision Tree',
64              'Random Forest', 'Support Vector Machine',
65              'Stochastic Gradient Descent', 'Multi-Layer Perceptron'],
66              [71, 85, 72, 84, 85, 85, 83])
67         fig = px.scatter(acc, color='value',
68                           labels={'value': 'Machine Learning Model', 'index': 'Accuracy Score (%)'})
69         fig.update_traces(marker=dict(size=20, line=dict(width=1, color="DarkSlateGrey")),
70                           selector=dict(mode="markers"))
71         st.plotly_chart(fig, use_container_width=True)
72
73     # ---Confusion Matrix---
74
75     with st.container():
76         st.header("Confusion Matrix Logistic Regression")
77         LRmat = Image.open('image/LRmat.png')
78         st.image(LRmat, width=1200)
79     with st.container():
80         st.header("Confusion Matrix Stochastic Gradient Descent")
81         SGDmat = Image.open('image/SGDmat.png')
82         st.image(SGDmat, width=1200)
83     with st.container():
84         st.header("Confusion Matrix Support Vector Machine")
85         SVMmat = Image.open('image/SVMmat.png')
86         st.image(SVMmat, width=1200)
```