Rapport NLP Topic Modeling

SOMMAIRE

I-Introduction

Notre objectif

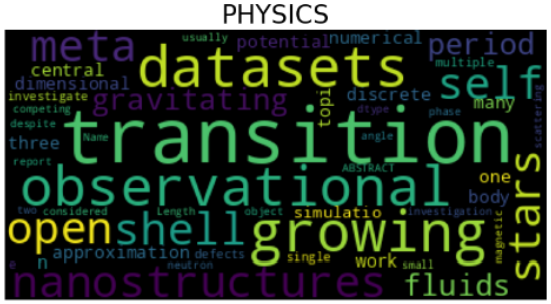
Notre objectif dans ce projet est de créer un outil qui utilise des techniques de traitement de texte pour répondre aux besoins de votre client.
Ici, nous avons créé une interface streamlit pour prédire le sujet du texte donné.

II-EDA

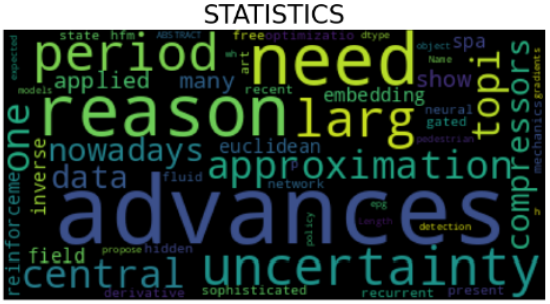
Pour la partie EDA nous avons commencé par créer des words clouds afin de visualiser les mots les plus utilisés pour chaque catégorie de sujets.

Exemples pour topics :

PHYSICS

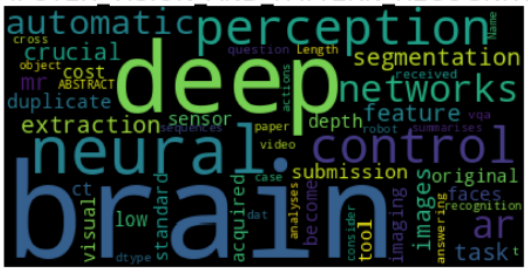


STATISTICS

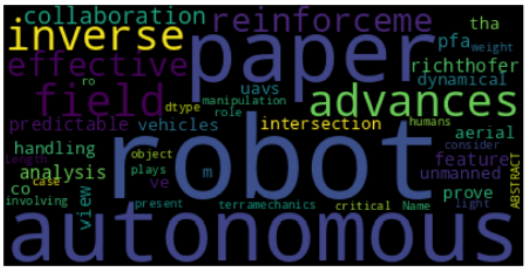


Exemple pour subtopics :

COMPUTER_VISION_AND_PATTERN_RECOGNITION



ROBOTICS



III-Modèles

Nous avons réalisés deux types de modèles comprenants plusieurs itérations

1/ Modèle LDA (Non Supervisé)

Mise en œuvre:

L'application est construite avec Python et differentes bibliothèques, notamment Streamlit, Gensim et heapq.

Étiquetage du sujet principale:

L'application comprend une fonctionnalité d'étiquetage de sujet qui attribue des étiquettes sectorielles pertinentes à un texte donné. Les secteurs et leurs mots-clés correspondants sont définis à l'aide de listes prédéfinies.
La fonction label_topic() utilise des expressions régulières pour rechercher ces mots clés dans le texte et compte leurs occurrences.
Les sujets principaux avec les nombres les plus élevés sont considérés comme les étiquettes du texte.

Prétraitement :

La fonction preprocess_text() traite le texte d'entrée en le segmentant en mots individuels et en supprimant les mots vides à l'aide de la bibliothèque Gensim. Cette étape permet d'améliorer la qualité des données textuelles pour la modélisation des sujets.

Modélisation de sujet :

La fonction perform_topic_modeling() effectue une modélisation de sujet sur le texte prétraité à l'aide de l'algorithme Latent Dirichlet Allocation (LDA) de la bibliothèque Gensim. Il crée un dictionnaire de mots uniques dans le texte, convertit le texte en une représentation de corpus de groupe de mots et applique le modèle LDA pour identifier les sujets les plus pertinents.
La fonction renvoie une liste de rubriques, chacune représentée par un ensemble de mots-clés.

Conclusion:

L'application Topic Modeling and Labeling offre un moyen pratique et efficace d'extraire des informations à partir de données textuelles. En tirant parti des techniques de modélisation

des sujets et des listes de mots-clés prédéfinis du secteur, l'application fournit aux utilisateurs une compréhension des sujets présents dans le texte et attribue des étiquettes sectorielles pertinentes. Cela peut être utile dans divers domaines, notamment l'analyse de contenu, la recherche d'informations et les tâches de classification de texte spécifiques à l'industrie.

2/ Modèle Classifier(Supervisé)

1. CountVectorizer :
    - CountVectorizer est initialisé avec un maximum de 10 000 caractéristiques. Cela signifie que le vectoriseur considérera uniquement les 10 000 mots les plus fréquents dans les données textuelles.
    - Les colonnes 'ABSTRACT' des ensembles de données d'entraînement et de test sont combinées dans une liste appelée 'combined'.
    - CountVectorizer est ajusté sur les données 'ABSTRACT' combinées, ce qui signifie qu'il apprend le vocabulaire et crée une représentation numérique des données textuelles.
2. Data Splitting :
    - L'ensemble de données d'entraînement est divisé en ensembles d'entraînement et de validation à l'aide de la fonction train_test_split de scikit-learn. L'ensemble de validation sera utilisé pour évaluer les performances du classificateur entraîné.
3. Text Transformation :
    - Les colonnes 'ABSTRACT' des ensembles de données d'entraînement, de validation et de test sont transformées à l'aide du CountVectorizer précédemment ajusté. Cela convertit les données textuelles en une représentation numérique adaptée aux algorithmes d'apprentissage automatique.
    - La fonction transform est utilisée pour convertir le texte en une représentation matricielle où chaque ligne représente un document et chaque colonne représente un mot. Les valeurs dans la matrice indiquent la fréquence de chaque mot dans le document correspondant.
4. OneVsRestClassifier :
    - OneVsRestClassifier est initialisé avec LogisticRegression comme estimateur de base. Cela signifie que plusieurs modèles de régression logistique seront entraînés, chacun représentant une classe ou une catégorie différente.
    - OneVsRestClassifier étend les algorithmes de classification binaire à la classification multiclasse en entraînant un classificateur distinct pour chaque classe. Il attribue la classe avec la plus grande confiance comme classe prédite pour une entrée donnée.
5. Training the Classifier :
    - OneVsRestClassifier est ajusté sur les données d'entraînement transformées. Cela entraîne les modèles de régression logistique multiples pour chaque classe.
6. Prediction and Evaluation :

- Le classificateur entraîné est utilisé pour prédire les étiquettes de l'ensemble de validation.
- La fonction classification_report est utilisée pour générer un rapport comprenant la précision, le rappel, le score F1 et le support pour chaque classe. Ce rapport fournit des informations sur les performances du classificateur sur l'ensemble de validation.
- Le rapport de classification est affiché dans la console, permettant à l'utilisateur d'évaluer la qualité des résultats de classification.

IV-Interface graphique

Concernant l'interface graphique, nous avons choisi Streamlit.

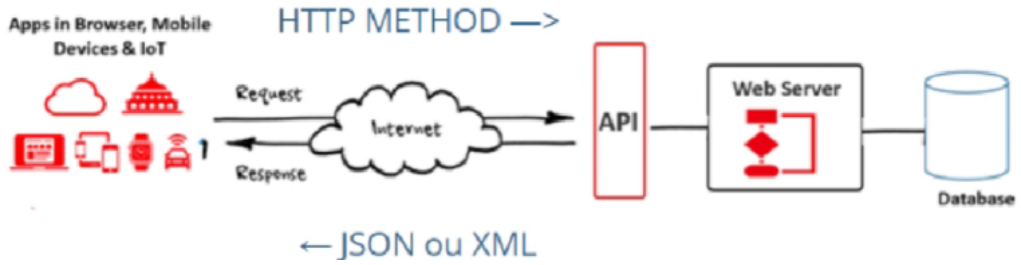
Interface interactive:
La bibliothèque Streamlit est utilisée pour créer une interface conviviale pour l'application.
L'interface fournit une zone de texte dans laquelle les utilisateurs peuvent saisir un texte.
En cliquant sur le bouton "Analyser", l'application effectue la modélisation et l'étiquetage des sujets sur le texte fourni. Les résultats, y compris le texte d'origine, les sujets identifiés et les étiquettes de sujets, sont affichés dans des colonnes séparées.

Utilisation:
Pour utiliser l'application, les utilisateurs doivent choisir l'option "Texte" dans le menu de la barre latérale. Ils peuvent ensuite saisir leurs données textuelles dans la zone de texte fournie. Après avoir cliqué sur le bouton "Analyser", l'application effectue la modélisation et l'étiquetage des sujets. Les sujets identifiés et les labels industriels correspondants sont présentés à l'utilisateur.

V-API

L'API a été réalisée en utilisant Azure, Streamlit et GitHub pour assurer une mise en production et une distribution facile de l'application.

Mise en œuvre :



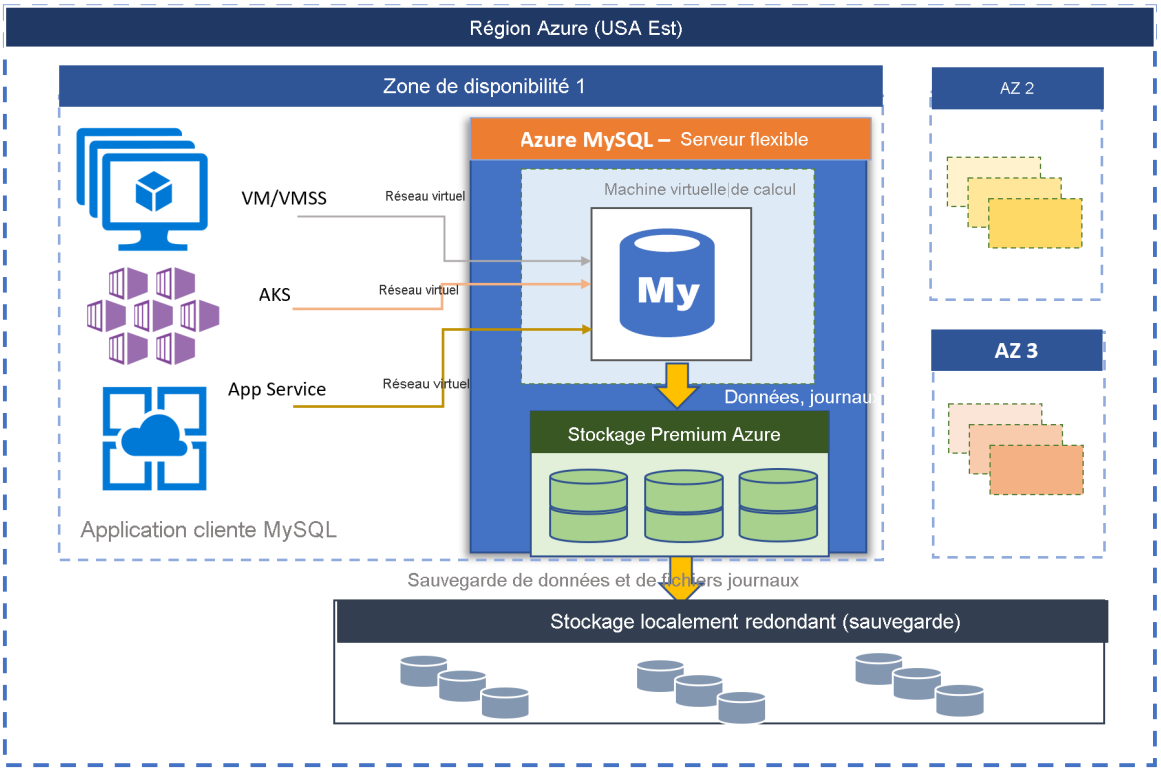
Pour créer l'API, nous avons utilisé Azure pour héberger et déployer l'application. Nous avons utilisé les services d'hébergement et de déploiement d'Azure pour rendre notre application accessible via une URL.

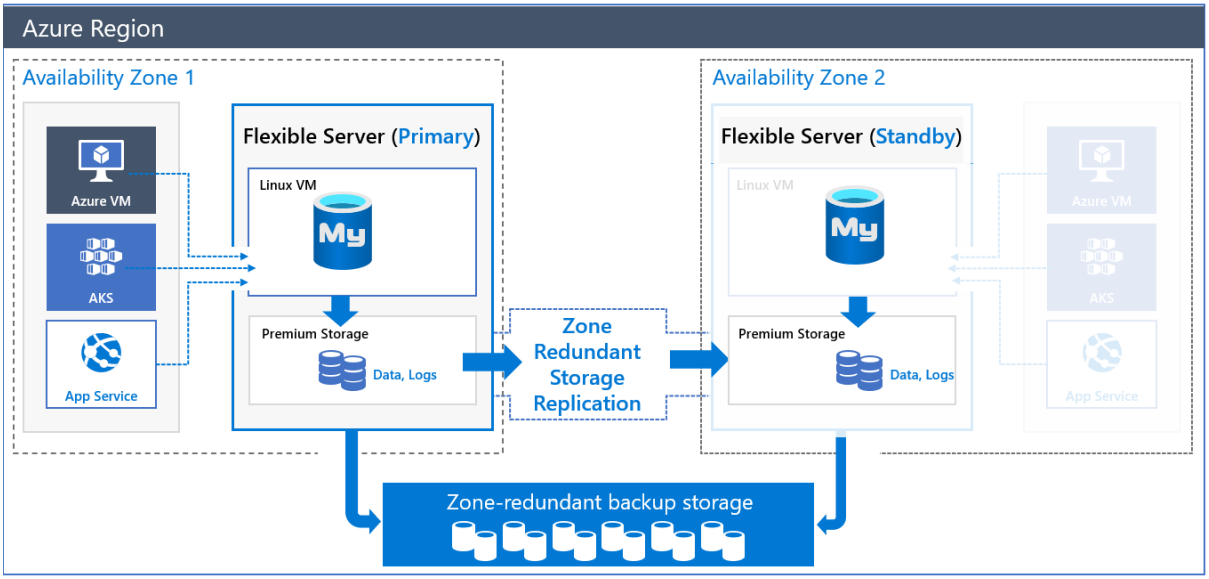
Interface utilisateur avec Streamlit :
La bibliothèque Streamlit a été utilisée pour créer l'interface utilisateur de l'API. Les utilisateurs peuvent accéder à l'API via un navigateur web et interagir avec l'interface pour obtenir des prédictions sur le sujet du texte donné.

Gestion de code avec GitHub :
GitHub nous a permis de collaborer efficacement en tant qu'équipe de développement, de suivre les modifications apportées au code et de gérer les différentes versions de l'application. De plus, GitHub facilite le déploiement continu de l'API en intégrant des pipelines de déploiement automatisés.

En combinant Azure, Streamlit et GitHub, nous avons créé une API performante et facilement accessible. Les utilisateurs peuvent interagir avec l'API en utilisant un navigateur web, saisir du texte et obtenir des prédictions sur les sujets correspondants. Cette approche offre une expérience utilisateur fluide et permet une distribution et une mise à jour efficaces de l'application grâce à l'intégration de GitHub pour la gestion du code et Azure pour le déploiement de l'API.

VI-Connexion BDD





1) Installez MySQL Workbench :

Si vous ne l'avez pas déjà fait, téléchargez et installez MySQL Workbench sur votre ordinateur local. MySQL Workbench est un outil visuel qui vous permet de gérer et d'interagir avec les bases de données MySQL.

1) Obtenez les détails de la connexion au serveur flexible Azure :

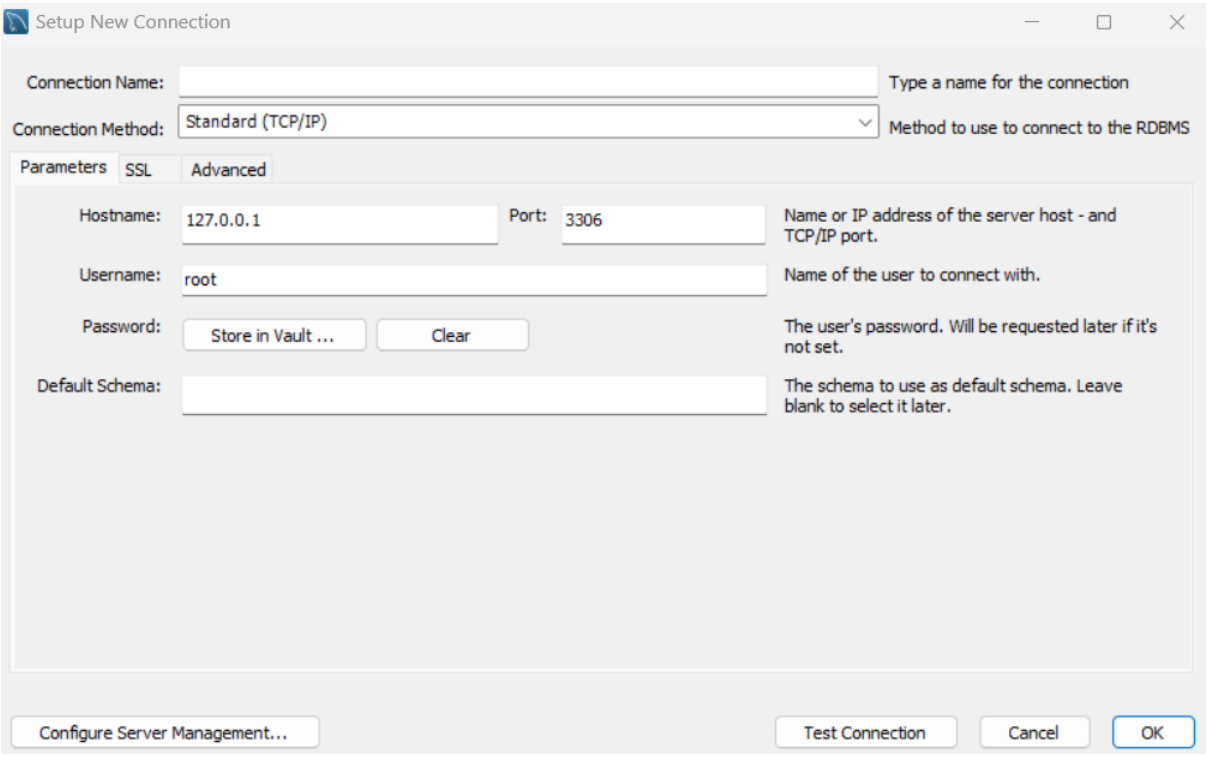
Dans le portail Azure, accédez à votre instance Azure Flexible Server.

Dans la section "Chaînes de connexion", vous trouverez les informations nécessaires pour vous connecter à la base de données, notamment le nom du serveur, le nom d'utilisateur et le mot de passe.

2) Ouvrez MySQL Workbench :

Lancez MySQL Workbench sur votre ordinateur local.
Créez une nouvelle connexion MySQL :
Dans MySQL Workbench, cliquez sur le bouton "Nouvelle connexion" dans la section "Connexions MySQL". Cela ouvrira la fenêtre "Configurer une nouvelle connexion".

3) Configurez les paramètres de connexion :



Dans la fenêtre "Configurer une nouvelle connexion", fournissez les détails suivants :

- Nom de connexion : Donnez un nom descriptif pour la connexion.
- Méthode de connexion : sélectionnez "Standard TCP/IP over SSH" dans le menu déroulant.
- Nom d'hôte SSH : saisissez le nom DNS ou l'adresse IP du serveur flexible Azure.
- Nom d'utilisateur SSH : fournissez le nom d'utilisateur SSH, qui est généralement le même que le nom d'utilisateur de la base de données.
- Mot de passe SSH : entrez le mot de passe SSH associé à l'utilisateur de la base de données.

- Nom d'hôte MySQL : saisissez le nom DNS ou l'adresse IP du serveur flexible Azure.
- Port du serveur MySQL : spécifiez le numéro de port pour MySQL, qui est généralement 3306.
- Nom d'utilisateur : indiquez le nom d'utilisateur de l'utilisateur de la base de données.
- Mot de passe : saisissez le mot de passe de l'utilisateur de la base de données.

Testez la connexion :
Cliquez sur le bouton "Tester la connexion" pour vérifier si la connexion peut être établie avec succès. MySQL Workbench tentera de se connecter au serveur flexible Azure à l'aide des informations fournies.

VII- Axes d'amélioration

- Securité : Nous aimerions prochainement sécuriser avec des tokens et clés
- Model : Améliorer le modèle
- Terminer déploiement