

## **Sujet 1. Chatbot Basique avec Traitement Automatique du Langage (TAL)**

Objectif : Créer un chatbot simple capable de répondre à des questions basiques en utilisant des techniques de traitement automatique du langage.

Description :

- Introduction au TAL : Expliquer ce qu'est le traitement automatique du langage, y compris les concepts de base comme la tokenisation (découper le texte en mots ou phrases), la lemmatisation (réduire les mots à leur forme de base), et l'analyse syntaxique.
- Interface Utilisateur : Développer une interface où les utilisateurs peuvent entrer des questions ou des messages.
- Moteur de Réponse : Utiliser des bibliothèques de TAL comme NLTK (Python) ou spaCy pour analyser les questions et générer des réponses pré-définies ou basées sur des modèles simples.
- Exemples de Réponses : Fournir des réponses basées sur des règles ou des patterns (par exemple, "Quelle est la météo ?" pourrait répondre avec "Je ne sais pas, mais vous pouvez vérifier votre application météo.").

Technologies : Python, NLTK, spaCy, ou des outils similaires.

## **Sujet 2. Système de Recommandation de Films**

Objectif : Développer un système qui recommande des films aux utilisateurs en fonction de leurs préférences ou de leurs comportements passés.

Description :

- Collecte des Données : Utiliser des bases de données de films et des évaluations d'utilisateurs (par exemple, MovieLens).

- **Modèle de Recommandation** : Expliquer les deux types principaux de systèmes de recommandation :
- **Filtrage Collaboratif** : Recommande des films en fonction des préférences similaires d'autres utilisateurs (par exemple, si deux utilisateurs ont aimé des films similaires, recommander un film que l'un a aimé mais que l'autre n'a pas encore vu).
- **Filtrage Basé sur le Contenu** : Recommande des films en fonction des caractéristiques du film que l'utilisateur a aimé (par exemple, si un utilisateur aime les films d'action, recommander d'autres films d'action).
- **Interface Utilisateur** : Créer une interface où les utilisateurs peuvent entrer leurs préférences ou voir les recommandations.

Technologies : Python, Scikit-learn pour les algorithmes, Pandas pour la manipulation des données, et éventuellement des frameworks comme TensorFlow ou PyTorch pour des modèles plus avancés.

## **Sujet 3. Application de Reconnaissance de Plantes**

**Objectif** : Créer une application capable d'identifier des plantes en utilisant des photos prises par les utilisateurs.

**Description** :

- **Collecte des Données** : Utiliser des ensembles de données d'images de plantes avec des étiquettes (par exemple, PlantNet ou des ensembles de données open-source).
- **Modèle de Reconnaissance** : Utiliser des modèles de vision par ordinateur pour identifier les plantes. Les modèles de réseau de neurones convolutifs (CNN) sont souvent utilisés pour ce type de tâche.
- **Développement de l'Application** : Créer une interface où les utilisateurs peuvent prendre une photo ou télécharger une image de plante. L'application traitera l'image et affichera le nom de la plante ou des informations connexes.
- **Fonctionnalités Supplémentaires** : Ajouter des informations supplémentaires sur les plantes identifiées (par exemple, conseils de culture, caractéristiques).

Technologies : Python, TensorFlow ou PyTorch pour les modèles d'apprentissage automatique, OpenCV pour le traitement d'image, et un framework comme Flutter ou React Native pour le développement de l'application mobile.