



**IUT LUMIÈRE**

# **Documentation Utilisateur**

**Z. ABASSI  
N. AJOURJA  
G. BREGAND  
C. GUY**

---

**Juin 2025**

# Sommaire

---

**01.** Objectif du projet

**02.** Glossaire

**03.** Avant l'exécution

Kaggle

HuggingFace

Les librairies

**04.** Instructions d'utilisation

# Objectif du projet

---

Dans le cadre d'un projet universitaire, il nous a été demandé de mettre au point un agent conversationnel pour aider les étudiants du BUT Sciences des données dans le cadre des modules « Bases de la programmation ».

Nous avons développé notre système sur Kaggle, plateforme qui permet d'utiliser GPU ce qui nous permet d'avoir plus d'espace de stockage et de la mémoire. Il s'appuie sur des supports de cours (PDF, notebooks, csv).

# Glossaire

---

**Agent conversationnel / chatbot** : un programme informatique conçu pour simuler une conversation avec des utilisateurs, souvent par écrit ou par la voix.

**Lemmatisation** : c'est un processus qui me permet de ramener un mot à sa forme de base, qu'on appelle la lemme. Par exemple : manges > mangeons > mangera deviennent tous manger.

**Modèle de langue** : c'est un programme informatique qui a appris à comprendre et produire du texte en se basant sur de grandes quantités de textes écrits par des humains. Il va essayer de prédire les mots qui vont venir dans une phrase, un peu comme lorsqu'on devine ce qu'une personne va dire ou écrire à partir des premiers mots.

**Token** : c'est un petit morceau de texte. Quand un ordinateur lit un texte, il le découpe en petits morceaux, appelés tokens, pour pouvoir le traiter plus facilement.

**GPU** : c'est une puce électronique qui aide l'ordinateur à faire des calculs très rapides, surtout pour des tâches visuelles ou très complexes. Ils sont aussi très utiles pour faire tourner des modèles d'intelligence artificielle.

**RAG (Retrieval-Augmented Generation)** : c'est une technique utilisée en intelligence artificielle pour améliorer les réponses données par un modèle en lui permettant d'aller chercher des informations à l'extérieur avant de répondre.

**Chunk** : c'est une façon de découper l'information en petites unités plus faciles à traiter.

# Avant l'exécution

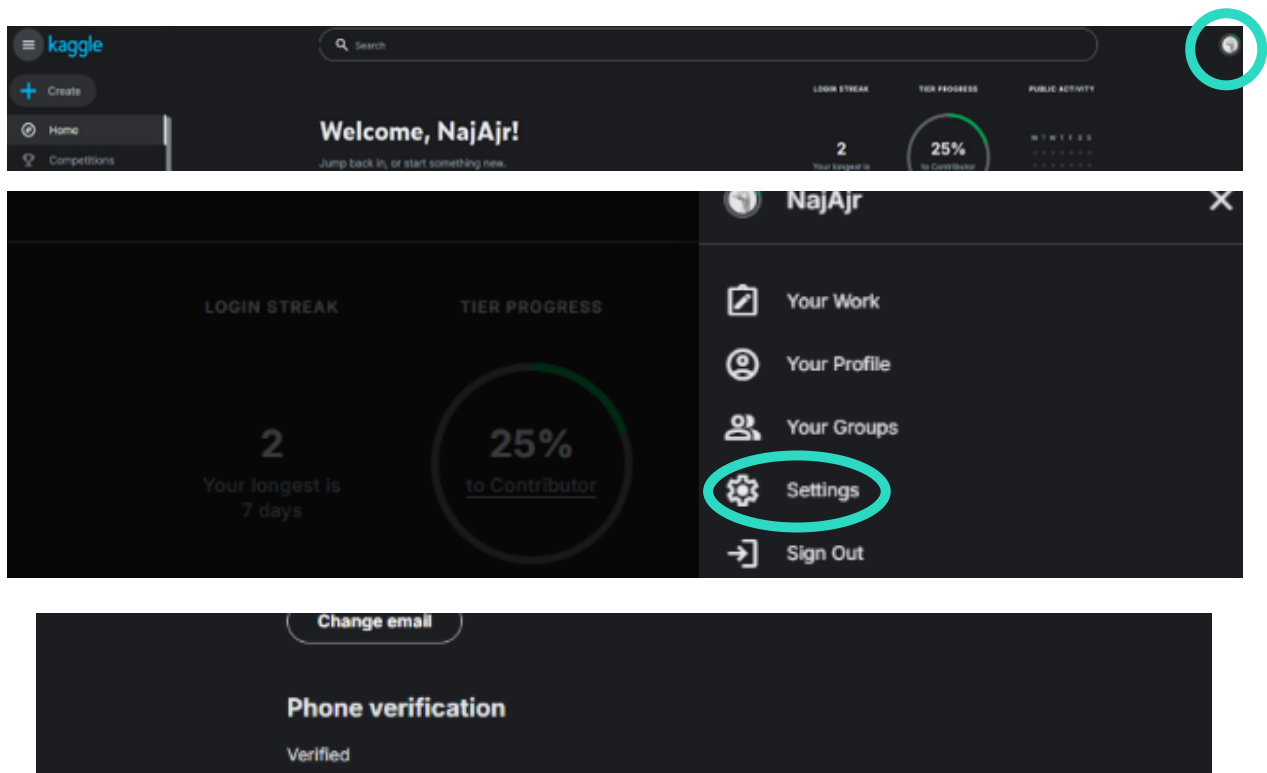
Afin d'optimiser au mieux votre expérience avec notre agent conversationnel. Pour cela, il vous faudra un compte Kaggle et HuggingFace pour lancer le code. c'est ce que nous expliquons dans cette partie. Suivez étape par étape pour ne rien manquer et ainsi faciliter la prise en main du code.

## 1 Kaggle

Assurez-vous d'avoir un compte Kaggle pour pouvoir exécuter le code sur la plateforme.

kaggle

1. Connectez vous ou créez votre compte
  - Si ce n'est pas déjà fait, vérifiez votre compte avec un numéro de téléphone dans l'onglet **Setting** > **Phone verification** pour permettre par la suite l'utilisation de GPU.



## 2

## HuggingFace

Un compte [HuggingFace](https://huggingface.co/) sera également nécessaire pour avoir accès aux modèles all-MiniLM-L6-v2 et mistralai/Mistral-7B-Instruct-v0.3 grâce un token :



1. Allez sur <https://huggingface.co/>.
2. Connectez-vous ou créez votre compte.
3. Cliquez sur votre photo de profil > Access Tokens > Create new token puis choisissez un nom pour votre token et sélectionnez :
  - Read access to contents of all repos under your personal namespace
  - Read access to contents of all public gated repos you can access
4. Cliquez sur Create token
5. Copiez votre token et gardez-le sous la main : plus tard dans le programme, il vous sera demandé de l'insérer (cela se fera de manière sécurisée directement avec le logging de la bibliothèque HuggingFace)

## 2

## Les librairies

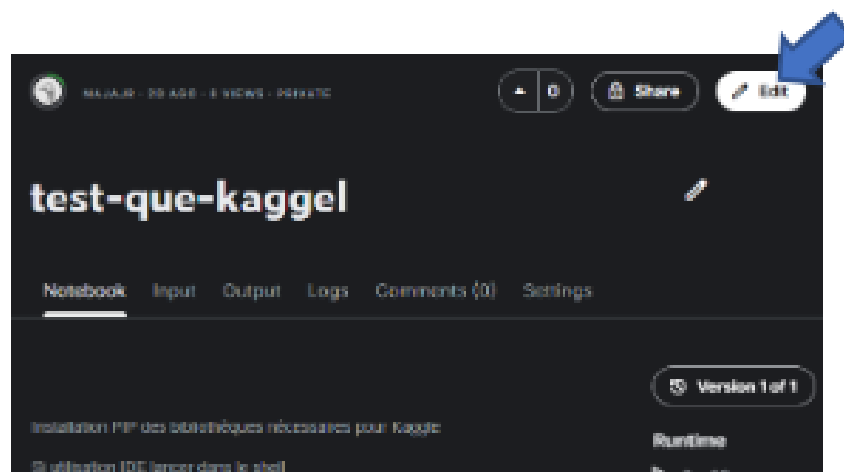
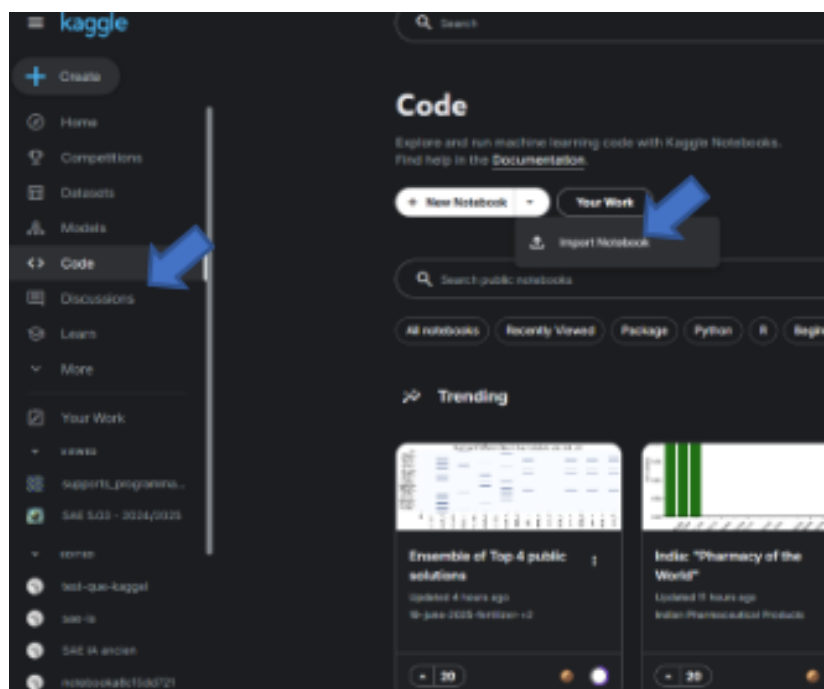
Plusieurs librairies python sont à installer pour la bonne exécution du code. Elles sont déjà incluses dans le début du code :

- pytesseract : permet de lire le texte contenu dans les images (OCR).
- pillow : sert à ouvrir, modifier ou analyser des images.
- llama\_index : le cœur du chatbot : il organise les documents, les transforme en vecteurs, et permet de poser des questions pertinentes.
- pymupdf : utilisé pour lire et extraire le texte des fichiers PDF.
- ftfy : corrige les textes mal encodés (accents mal affichés, caractères bizarres.)
- nltk : aide à découper et analyser les phrases pour mieux comprendre le langage.
- faiss-cpu : permet de retrouver très rapidement dans quels documents on parle d'un sujet donné (moteur de recherche interne).

Il vous sera également nécessaire d'installer le modèle de langue Spacy. Cela permet d'obtenir un modèle de langue française de taille moyenne nécessaire pour certains traitements linguistiques (lemmatisation, POS tagging, ...).

# Instructions d'utilisation

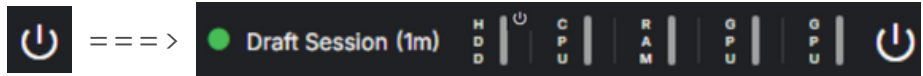
1. Préparez les fichiers de supports de cours dans un fichier zip
2. Connectez vous sur Kaggle, aller dans Code, puis cliquer dans Import Notebook.
3. Importez le fichier de code test-que-kaggle.ipynb. Vous pourrez seulement voir le code mais pas l'exécuter ou le modifier.
4. Pour ce faire, cliquez sur Edit :



5. Importer le fichier .zip dans l'espace de travail Kaggle avec Upload, nommé ce dataset zip-supports-programmation dans Kaggle.

6. Dans l'onglet Settings > Accelerator, choisissez le GPU T4x2.

7. Démarrer la session

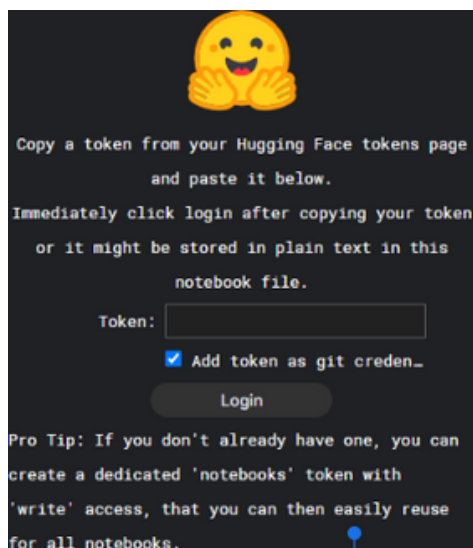


8. Pour lancer le code, cliquer sur la cellule de code et appuyer sur

A screenshot of a code cell in a Jupyter notebook. The first cell's run button (a right-pointing triangle) is circled in red. The code in the cell is as follows:

```
%%writefile /kaggle/working/ConversaSD_parametre.py
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
#
```

9. Lorsque vous lancerez la 7ème cellule, une clé token vous sera demandé de renseigner votre token (que vous avez crée plus tôt) :



10. L'agent conversationnel est prêt ! Posez votre question et tapez sur Entrée

