

# Jupyter Notebook - Librairie Manquante

CSI4506 Introduction à l'intelligence artificielle

Marcel Turcotte

2025-09-05

## Objectif d'apprentissage

- **Illustrer** le processus d'identification et de résolution des problèmes de bibliothèque manquante dans Google Colab.

! Important

Cet exemple est destiné à être exécuté dans [Google Colab](#).

## Import

Dans ce cahier, nous utilisons la bibliothèque `langdetect` pour identifier avec précision la langue des échantillons de texte donnés.

Tout d'abord, importons `langdetect`.

```
from langdetect import detect
```

L'exécution de la cellule de code ci-dessus entraînera une erreur, car la bibliothèque `langdetect` n'est pas installée par défaut dans Google Colab.

```
-----  
ModuleNotFoundError                                Traceback (most recent call last)  
/tmp/ipython-input-193950075.py in <cell line: 0>()  
----> 1 from langdetect import detect
```

```
ModuleNotFoundError: No module named 'langdetect'
```

-----  
NOTE: Si votre importation échoue en raison d'un package manquant, vous pouvez installer manuellement les dépendances en utilisant soit `!pip` soit `!apt`.

Pour voir des exemples d'installation de certaines dépendances courantes, cliquez sur le bouton "Open Examples" ci-dessous.

-----

Ce problème peut être résolu en ajoutant la ligne de code suivante avant la première instruction `import`. Essayez-le!

```
! pip install langdetect
```

Une fois ce problème résolu, nous pouvons appeler `detect`. Essayez-le!

```
detect("Bonjour tout le monde!")
```

## Meilleure Pratique

Vous auriez pu prédire ce scénario. Si la bibliothèque `langdetect` n'est pas installée, capturez l'exception résultante et procédez à l'installation de la bibliothèque.

```
try:
    from langdetect import detect
except ImportError:
    print("langdetect non trouvé, installation...")
    import sys
    ! pip install langdetect
    from langdetect import detect # réessayer après l'installation
```

## Exploration

! permet d'exécuter des [commandes shell Unix/Linux dans IPython](#). Créez une cellule de code et essayez ces commandes.

- ! `uname -a` affiche des informations sur le système.
- ! `ls` affiche le contenu du répertoire courant.
- ! `ls /` affiche le contenu du répertoire racine.

— ! `pwd` renvoie le nom du répertoire de travail.

Ces commandes sont utiles pour déboguer le code, car elles fournissent des informations sur l'environnement informatique, telles que la version du système d'exploitation et le contenu du répertoire local.