ECOLE MAROCAINE DES SCIENCES DE L'INGENIEUR
Membre de HONORIS UNITED UNIVERSITIES

Streamlit

Niveau : 5ère année
Filière : IIR
Année universitaire : 2025/2026

TP – Streamlit

Streamlit est une bibliothèque Python open-source qui facilite la création et le partage d'applications Web personnalisées pour le machine learning et la science des données. En utilisant Streamlit, vous pouvez rapidement créer et déployer des applications de données puissantes.

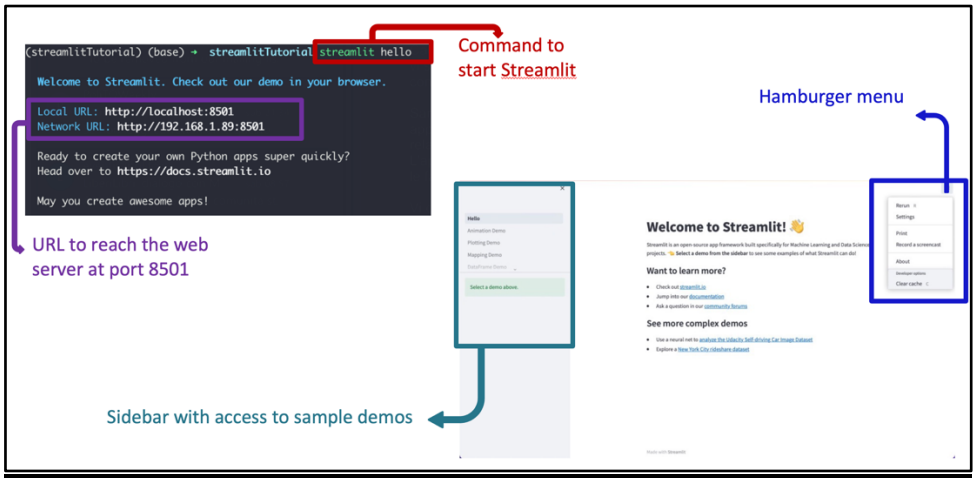
Dans le cadre du développement d'applications interactives pour visualiser des données, tester des modèles d'IA ou créer des dashboards simples, Streamlit est un framework Python très utilisé.

Il permet de transformer un script Python en une application web sans écrire une seule ligne de HTML, CSS ou JavaScript. Ce TP vous introduit progressivement à ses concepts clés.

Partie 1 — Installation et prise en main

1. Créez un nouvel environnement Python.
2. Installez Streamlit : *pip install streamlit*
3. Vérifiez l'installation : *streamlit hello*
4. Notez ce que vous observez dans votre navigateur (page d'exemple interactive).

Plus d'informations sur : **https://docs.streamlit.io/get-started/installation**



Partie 2 — Créer votre première application

Créez un fichier app.py et écrivez un script minimal :

- Afficher un titre.
- Afficher un texte descriptif.
- Afficher une image ou un emoji.

```
import streamlit as st

# Titre
st.title("Ma première application Streamlit")

# Texte
st.write("Bienvenue dans votre première application web créée avec Streamlit !")

# Image ou emoji
st.write("Voici un emoji pour vous encourager : 😄")
```

- Lancer l'application via : *streamlit run app.py*

Partie 3 — Widgets et Interactivité

Dans votre app.py, ajoutez les éléments suivants :

1. Un champ texte où l'utilisateur écrit son nom.
2. Un bouton "Afficher".
3. Lorsque le bouton est cliqué, afficher un message personnalisé :
   "Bonjour, EMSI - Casa ! Bienvenue dans votre première app Streamlit."
4. Ajoutez une barre latérale (st.sidebar) avec :
   - un slider de sélection de nombre,
   - une sélection (selectbox) pour choisir une couleur.

```
import streamlit as st
st.title("Application Interactive - Débuter avec Streamlit")
st.write("Ce TP vous montre comment utiliser les widgets.")

# -------- CHAMP TEXTE --------
nom = st.text_input("Entrez votre nom :")

# -------- BOUTON --------
if st.button("Afficher"):
st.success(f"Bonjour, EMSI - Casa ! Bienvenue dans votre première app Streamlit 😊")
```

```
# -------- SIDEBAR --------
st.sidebar.title("Paramètres")
nombre = st.sidebar.slider("Choisissez un nombre :", 1, 100, 25)
couleur = st.sidebar.selectbox("Choisissez une couleur :", ["Rouge", "Vert", "Bleu"])

# Affichage dynamique
st.write(f"Vous avez choisi : **{nombre}** et la couleur **{couleur}**.")
```

Partie 4 — Manipulation de données

1. Créez un petit DataFrame pandas (ex. données aléatoires ou un mini dataset).
2. Affichez-le avec : *st.dataframe(df)*
3. Ajoutez un graphique simple (ligne ou barres) avec matplotlib ou plotly.

```
import streamlit as st
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

st.title("Affichage de données avec Streamlit")

# -------- DATAFRAME --------
data = {
    "A": np.random.randint(1, 100, 10),
    "B": np.random.randint(1, 100, 10)
}
df = pd.DataFrame(data)

st.subheader("Données du DataFrame")
st.dataframe(df)

# -------- GRAPHIQUE --------
st.subheader("Graphique de la colonne A")

fig, ax = plt.subplots()
ax.plot(df["A"])
ax.set_title("Valeurs de la colonne A")
ax.set_xlabel("Index")
ax.set_ylabel("Valeur")

st.pyplot(fig)
```

Partie 5 : Mini-projet – Déploiement d'un modèle ANN / CNN avec Streamlit

Déployer un modèle de classification d'images déjà entraîné ANN/CNN pour classer images.

1. Préparation de l'environnement

a. Sauvegarder le modèle déjà entraîné sous format .h5 (pour Keras/TensorFlow)

NB: Appelez model.save pour enregistrer l'architecture, les pondérations et la configuration d'entraînement d'un modèle dans un seul fichier/dossier. Cela vous permet d'exporter un modèle afin qu'il puisse être utilisé sans avoir accès au code Python d'origine*. Étant donné que l'état de l'optimiseur est récupéré, vous pouvez reprendre l'entraînement exactement là où vous l'aviez laissé.

b. Installer Streamlit (NB: support python 3.8 à 3.12)

- Créer un environnement avec la commande python -m venv .venv
- Activer l'environnement .\.venv\Scripts\Activate.ps1 (powershell)

NB: If PowerShell is blocking the execution of scripts for security reasons type: Set-ExecutionPolicy -Scope Process -ExecutionPolicy Bypass

2. Développement de l'application Streamlit

Pour structurer un projet Streamlit pour le déploiement d'un modèle de classification d'images, il est utile d'organiser les fichiers et dossiers de manière à rendre le projet clair, modulaire, et facile à maintenir.

Voici une structure de projet recommandée :

CNN_streamlit/

```
├── main.py   # Script principal de l'application Streamlit
├── requirements.txt   # Liste des dépendances nécessaires (packages Python)
├── README.md   # Fichier de description du projet
├── model/
```

```
│   └── Bestmodel.h5    # Modèle pré-entraîné
├── assets/
│   └── Test_image.jpg    # Images d'exemple pour tester l'application
```

a. Créer un fichier main.py et importer les bibliothèques nécessaires (streamlit, tensorflow,..).

b. Configurer la structure de base de l'application avec un titre, une description de l'application, et une zone pour télécharger des images.

- Charger le modèle depuis un fichier local.
- Ajouter une fonctionnalité pour télécharger une image via st.file_uploader().
- Écrire une fonction de prétraitement pour transformer l'image téléchargée en entrée compatible avec le modèle (redimensionner, normaliser).
- Passer l'image traitée au modèle et obtenir les probabilités de prédiction pour chaque classe.
- Afficher les résultats sous forme de texte ou de graphique à l'aide de st.write() ou st.bar_chart().

c. Exécuter l'application Streamlit **streamlit run app.py**