**TP D’APPLICATION**

Voici un TP d'apprentissage pour débutants sur **Hugging Face**. L’objectif de cet exercice consiste à regrouper tous les modèles dans une seule application.

1. Créer une autre application Flask pour tester le model TensorFlow

Cf figures ci-dessous pour le modèle lineaire

Une image contenant texte, capture d’écran, logiciel, affichage

Description générée automatiquement

Une image contenant texte, logiciel, Police, Page web

Description générée automatiquement

**Objectifs du TP**

1. **Accéder au Hugging Face Model** : http://127.0.0.1:5000/hugging
2. **Accéder au modèle Tensor :** http://127.0.0.1:5000/tensor

**Pré-requis**

* Python 3.x
* Bibliothèque transformers (pip install transformers)
* Bibliothèque torch (pip install torch)
* tensorflow (pip install tensorflow)

Si ce n’est pas déjà fait, installez-les avec la commande suivante :

**pip install transformers**

**pip install torch**

**Plan du TP**

**Partie 1 : Création d'une application basique**

1. **Créer un fichier app.py** qui va contenir la logique de notre application Flask. Créez un fichier app.py dans le répertoire de votre choix avec le code suivant :

Une image contenant texte, capture d’écran, logiciel, Logiciel multimédia

Description générée automatiquement

1. **Créer un Template HTML**. Créez un dossier Template dans le même répertoire que app.py, puis créez un fichier index.html, hugging.html et tensor.html

Index.html

Une image contenant texte, capture d’écran, logiciel, Logiciel multimédia

Description générée automatiquement

Hugging.html

Une image contenant texte, capture d’écran, logiciel, Logiciel multimédia

Description générée automatiquement

Tensor.html

Une image contenant texte, capture d’écran, logiciel, Logiciel multimédia

Description générée automatiquement

**Partie 3 : Ajouter des styles CSS**

1. **Créer un dossier static** à côté de app.py et y ajouter un fichier style.css :

Une image contenant texte, capture d’écran, logiciel, Logiciel multimédia

Description générée automatiquement

1. **Lancer l'application**. Dans le terminal, exécutez python app.py. Accédez à http://127.0.0.1:5000/ dans votre navigateur pour voir la page avec le message "Bienvenue dans votre application !".

Une image contenant texte, capture d’écran, logiciel, Police

Description générée automatiquement

1. **Modifier app.py pour traiter les différents formulaires** :

Une image contenant texte, capture d’écran, logiciel, Système d’exploitation

Description générée automatiquement

Une image contenant texte, capture d’écran, logiciel, Logiciel multimédia

Description générée automatiquement

Une image contenant texte, capture d’écran, logiciel

Description générée automatiquement

1. **Tester le formulaire** : Lancez à nouveau l'application avec python app.py, allez sur <http://127.0.0.1:5000>

**Une image contenant texte, capture d’écran, Police

Description générée automatiquement**

**Une image contenant texte, Police, capture d’écran

Description générée automatiquement**

Exemple de texte : Artificial intelligence (AI), in its broadest sense

**Une image contenant texte, capture d’écran, Police, logiciel

Description générée automatiquement**

**Une image contenant texte, capture d’écran, Police

Description générée automatiquement**

**Une image contenant texte, capture d’écran, logiciel, affichage

Description générée automatiquement**

Exemple : y=2x−1. Calculons Y avec x =1

Une image contenant texte, capture d’écran, Police, ligne

Description générée automatiquement

**PARTIE 4 :**

**Ajouter un troisième lien :** [**http://127.0.0.1:5000/bert**](http://127.0.0.1:5000/bert)

**Pour la simulation du modèle Bert**