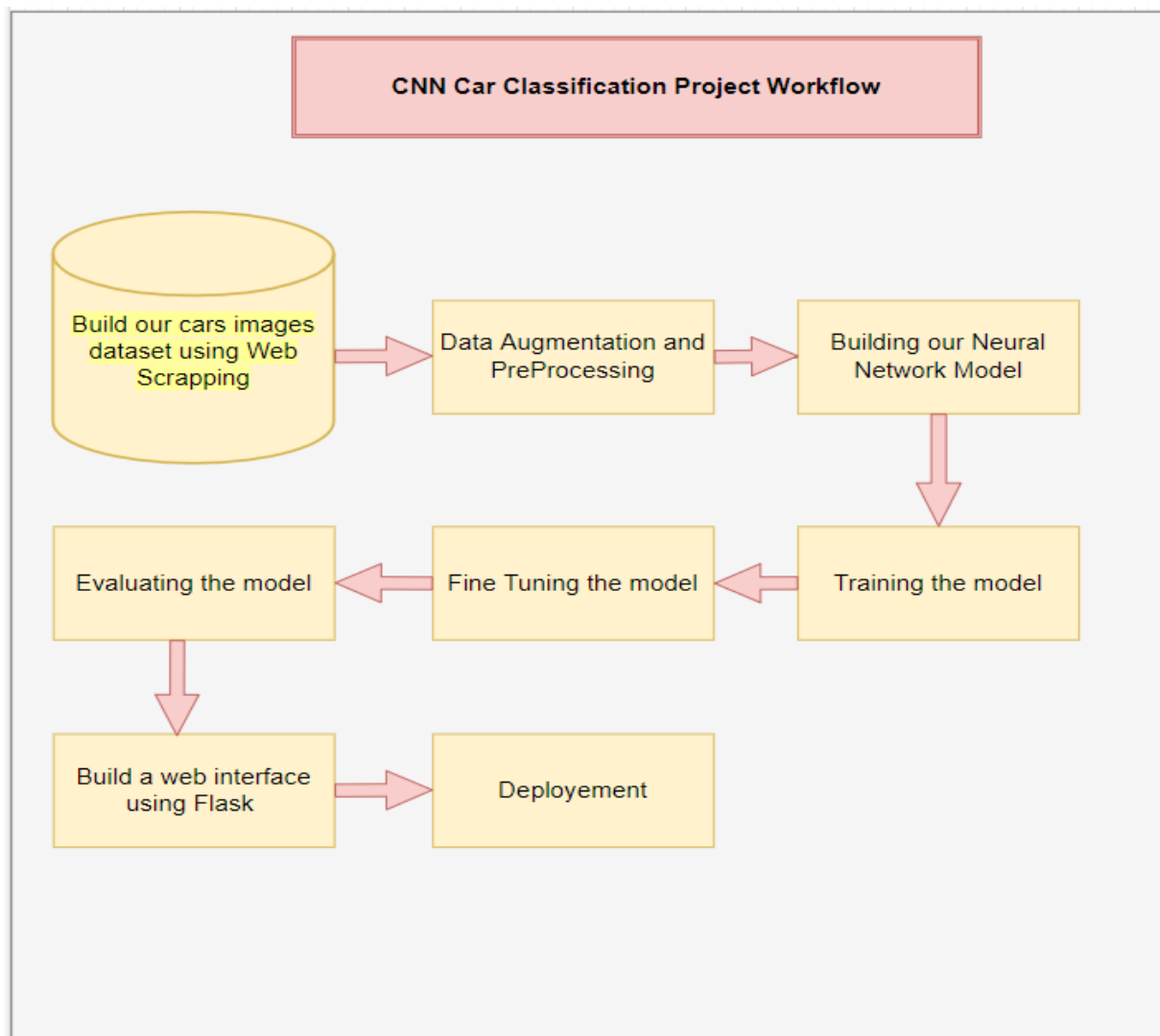


CNN CASE STUDY

Project Workflow :

Dans ce projet on va traiter principalement des images dans le cadre du computer vision à l'aide des techniques du Deep Learning. Nous allons créer un système pour la classification des véhicules : On va commencer par extraire notre base de données en utilisant le Web Scrapping. Ensuite, on passe à l'étape d'images preprocessing et Data Augmentation pour traiter nos images afin de rendre notre modèle plus accurate. Et comme vous pouvez deviner, l'étape suivante est l'implémentation et l'entraînement du modèle. On peut par la suite effectuer le fine tuning pour améliorer d'avantage notre modèle et enfin on peut visualiser nos true positive TP, true negative TN, False Positive FP et False Negative FN à l'aide de la matrice de confusion.

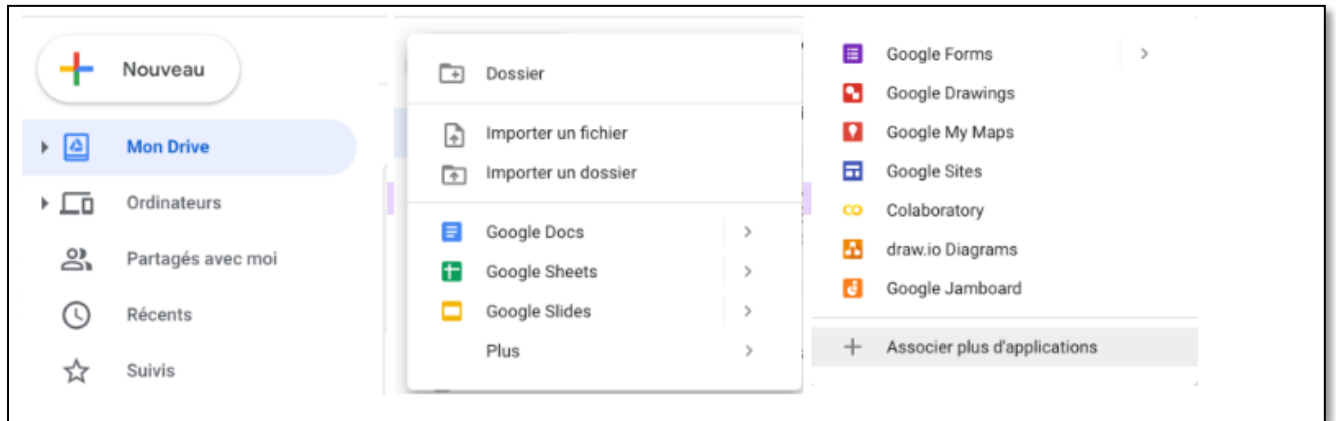
Pour pouvoir visualiser les prédictions de notre modèle, on va réaliser une interface web à l'aide de l'API Flask. Et enfin, on va la déployer.



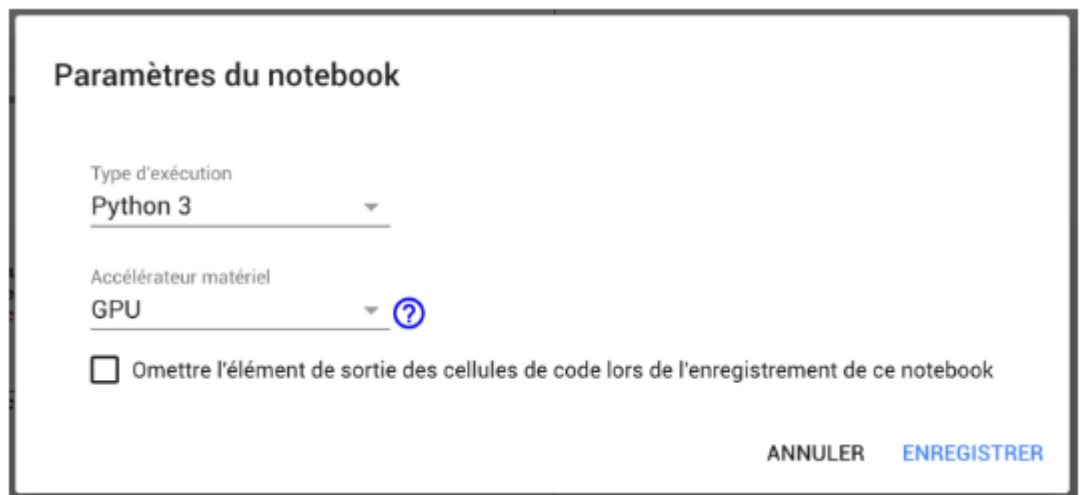
Google Colab :

Durant ce projet on va travailler sur google colab notebooks , donc si vous n'avez jamais utiliser google colab veuillez simplement suivre ces étapes :

- Pour utiliser Google colab il suffit d'aller sur google drive ensuite cliquer sur nouveau ensuite sur « plus » et choisissez « Colaboratory »



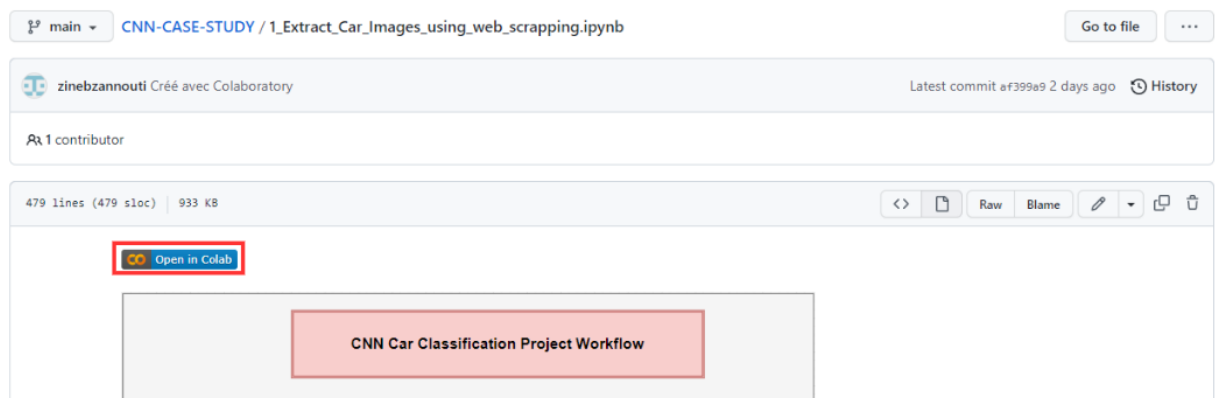
- Si vous n'avez pas l'option « Colaboratory » cliquez sur « Associer plus d'applications » puis rechercher « Colaboratory » et cliquez sur connecter.
- Connectez votre notebook au GPU pour des exécution plus rapide :
 - Pour passer en mode GPU, dans la barre des options choisir « exécution » puis « modifier le type d'exécution » et mettre l'option accélérateur matériel en mode GPU.



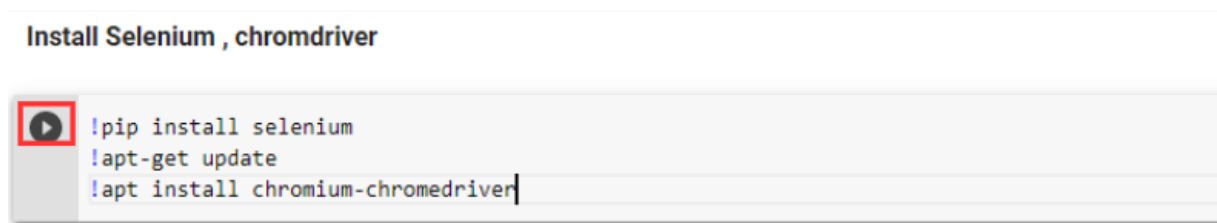
○

1- Extract car images using Web Scrapping :

- Notre but ici est de pouvoir extraire les images de la classe qu'on veut (On peut même créer une base de données de toutes les catégories et marque de voitures, mais pour optimiser le temps d'exécution nous allons juste se contenter d'une trentaine de classes) depuis Google image et les enregistrer dans le drive afin qu'on peut travailler avec dans l'intégralité de notre projet.
- Go to https://github.com/zinebzannouti/CNN-CASE-STUDY/blob/main/1_Extract_Car_Images_using_web_scrapping.ipynb
- Cliquez sur open in Colab

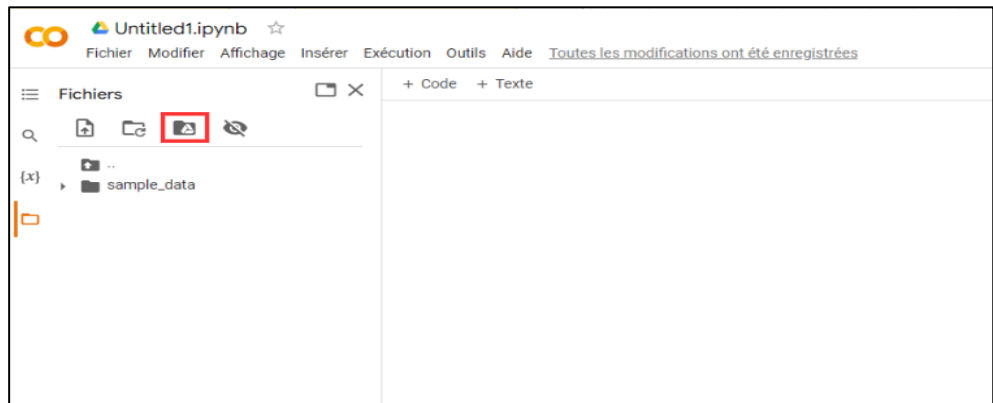


- Installez Selenium et chromdriver en exécutant la première cellule

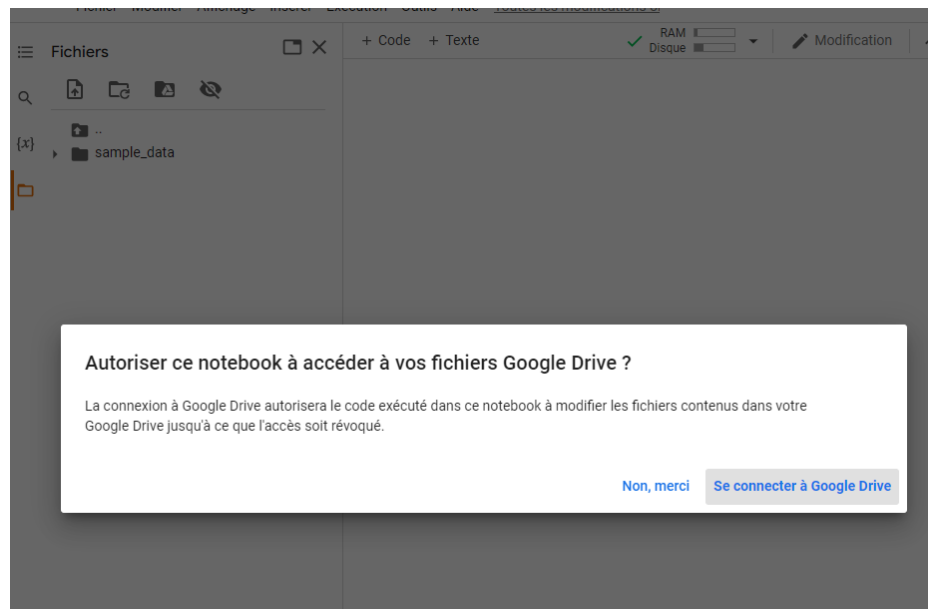


- Vous pouvez consulter la documentation de Selenium à chaque fois que vous en avez besoin <https://selenium-python.readthedocs.io/>
- Exécuter le reste du notebook (Attention à la section #TODO où il faut que vous ajoutiez des lignes de codes)
- Avant d'exécuter la dernière cellule, il faut connecter votre notebook avec Google drive afin que vous enregistrerez votre base de données et créer un dossier nommé 'CarsData' où vous allez mettre vos données.

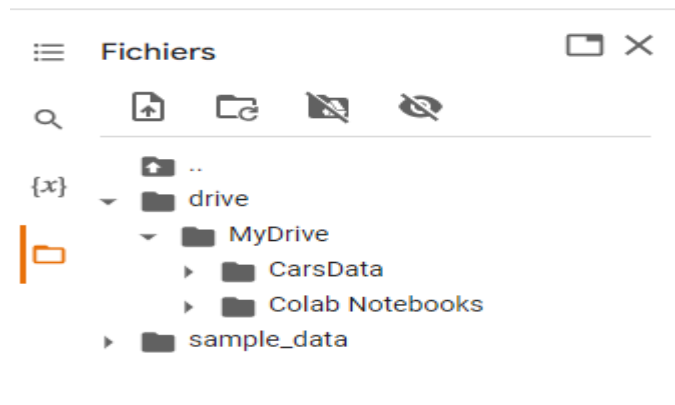
- Pour le faire , Cliquez sur cette icone :



- Cliquez sur Se connecter à Google Drive :



- Sélectionner votre compte.
- Créer votre dossier CarsData



- Maintenant vous pouvez exécuter la dernière commande (ça prend du temps entre 30min et 1 heures)

2- Image Preprocessing and Data Augmentation :

Un prétraitement est nécessaire pour nettoyer les données d'image pour l'entrée du modèle. L'augmentation aide votre modèle à généraliser en ajoutant plus de données d'entraînement.

- Go to https://github.com/zinebzannouti/CNN-CASE-STUDY/blob/main/Project-Exercices-Part/2_Image_Preprocessing_and_Data_Augmentation.ipynb
- Essayez de remplir les trous de codes en faisant attention à l'explication dans le notebook.
- Ensuite , vous pouvez exécuter l'ensemble du notebook.

3- CNN Model with Efficientnet and transfer learning:

On va créer notre modèle de Convolutional Neural Networks (CNN) on utilisant le transfer learning où on va prendre le modèle EfficientNetB3 déjà entrainer sur ImageNet et on va ajouter des couches à notre base modèle pour l'adapter à notre problématique de classification des images de voitures selon la marque et le modèle.

- Go to https://github.com/zinebzannouti/CNN-CASE-STUDY/blob/main/Project-Exercices-Part/3_CNN_Model_Transfer_Learning.ipynb
- Essayez de remplir les trous de codes en faisant attention à l'explication dans le notebook.
- Avant d'exécuter ce notebook il faut que vous copier les cellules de ce notebook et les ajouter dans le notebook précédant d'image processing et data augmentation.
- Après avoir collé vos cellules, vous pouvez exécuter la suite (ça peut prendre 1 heure pour compléter l'exécution)

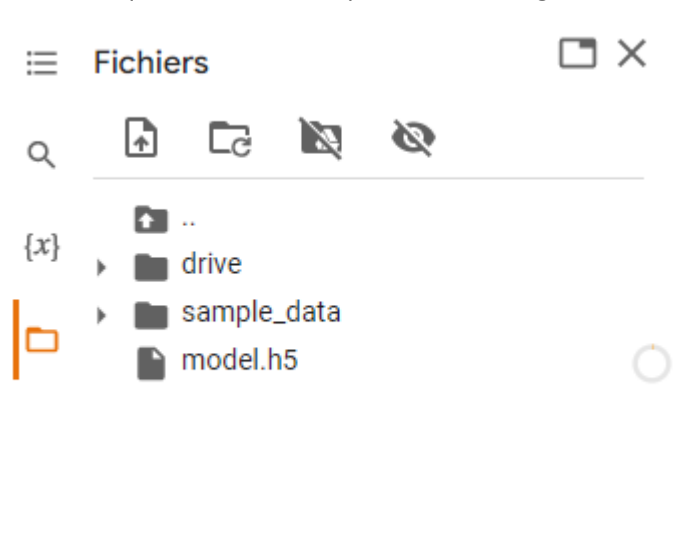
4- Fine Tuning:

Une fois que votre modèle a convergé sur les nouvelles données, vous pouvez essayer de dégeler tout ou partie du modèle de base et ré-entraîner tout le modèle de bout en bout avec un taux d'apprentissage très faible.

Il s'agit d'une dernière étape facultative qui peut potentiellement vous apporter des améliorations incrémentielles.

- Go to https://github.com/zinebzannouti/CNN-CASE-STUDY/blob/main/Project-Exercices-Part/4_Fine_Tuning.ipynb
- Essayez de remplir les trous de codes en faisant attention à l'explication dans le notebook.
- Avant d'exécuter ce notebook il faut que vous copier les cellules de ce notebook et les ajouter dans le notebook précédant.
- Après avoir collé vos cellules, vous pouvez exécuter la suite (ça peut prendre 30 minute pour compléter l'exécution)
- La dernière cellule c'est pour enregistrer notre modèle afin l'utiliser par la suite. Télécharger model.h5 dans votre machine locale.

- Double cliquez sur model.h5 pour le télécharger :



5- Confusion Matrix :

La performance d'un algorithme de Machine Learning est directement liée à sa capacité à prédire un résultat. Lorsque l'on cherche à comparer les résultats d'un algorithme à la réalité, on utilise une matrice de confusion.

- Go to [https://github.com/zinebzannouti/CNN-CASE-STUDY/blob/main/Project-Exercices-Part/5 Confusion matrix.ipynb](https://github.com/zinebzannouti/CNN-CASE-STUDY/blob/main/Project-Exercices-Part/5%20Confusion%20matrix.ipynb)
- Essayer de lire et comprendre le notebook
- Copier les cellules et les coller dans le notebook précédant .
- Exécuter cette partie.

6- Deploy Our model using Flask API :

What is Flask?

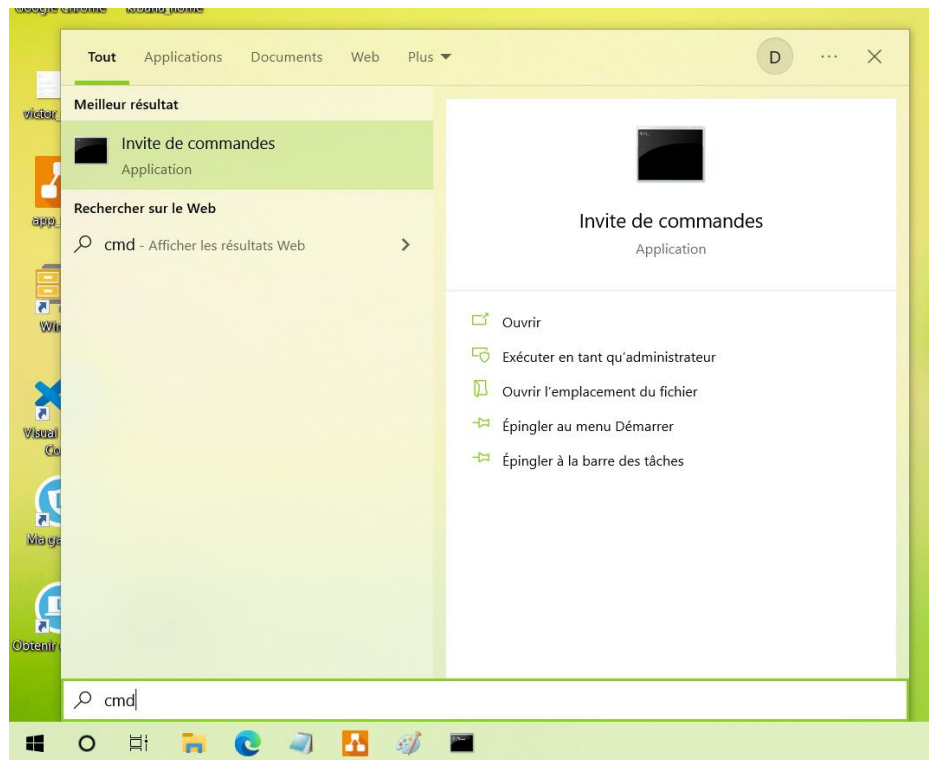


Flask

Flask est un framework d'application Web écrit en Python. Il dispose de plusieurs modules qui permettent à un développeur Web d'écrire plus facilement des applications sans avoir à se soucier des détails tels que la gestion des protocoles, la gestion des threads, etc.

Flask offre une variété de choix pour développer des applications Web et nous donne les outils et bibliothèques nécessaires qui nous permettent de créer une application Web.

- Pour réaliser une interface web avec flask il faut créer une template de notre page web en utilisant HTML : Dans notre projet on va créer un interface basic où vous pouvez uploader une image ou bien coller l'URL et avoir la prédiction , voici la template HTML qu'on a créer :
 - <https://github.com/zinebzannouti/DL-CarClassificationWithFlask/blob/main/templates/index.html>
- Ensuite on a notre flask app où on met le modèle qu'on a télécharger précédemment afin de pouvoir prédire la classe de chaque image, voici le code python :
 - <https://github.com/zinebzannouti/DL-CarClassificationWithFlask/blob/main/flaskapp.py>
- Veuillez suivre ces étapes pour déployer notre modèle et accéder à l'interface web :
 - Ouvrez votre terminal



- Go To <https://github.com/zinebzannouti/DL-CarClassificationWithFlask/blob/main/Deployment.md>
 - Suivez les étapes mentionner dans Deployment.md
- Une fois que vous avez réussi d'accéder à l'interface web :

Car Model and Make Classification

Upload Your Image :

Aucun fichier choisi

Use Link

- Allez au dictionnaire où on a les différentes classes de voiture
<https://github.com/zinebzannouti/DL-CarClassificationWithFlask/blob/main/dictionnaire.txt> ,
copier une classe, prenez une image de cette classe depuis Google et tester le résultat

Car Model and Make Classification

Upload Your Image :

Aucun fichier choisi

Use Link

Your Prediction : *FIAT 500*

