Practical - 7

**AIM : Deployment of ML project using Streamlit.**

Streamlit deployment within Docker in an ML lab is valuable for creating user-friendly, interactive ML applications. It combines the simplicity of Streamlit for building data apps with the containerization benefits of Docker, enabling easy sharing, reproducibility, and scalability of data-driven applications.

This approach streamlines the deployment of ML tools and enhances collaboration among researchers and stakeholders.

**Task 1:** Ensure that the required libraries are installed:

streamlit==1.10.0

pandas==1.2.3

scikit-learn==0.24.1

**Task 2:** Create the docker file using the steps described in theory material.

1. Create a Dockerfile :

# Use the official Python image as a base image

FROM python:3.8-slim

# Set the working directory inside the container

WORKDIR /app

# Copy the current directory contents into the container at /app

COPY . /app

# Install any dependencies specified in requirements.txt

RUN pip install --no-cache-dir -r requirements.txt

# Expose port 80 to the outside world

EXPOSE 80

# Set environment variable

ENV NAME World

# Define the command to run your application

CMD ["gunicorn", "--bind", "0.0.0.0:80", "app:app"]

1. Create a requirement.txt file :

scikit-learn==0.24.2

pandas==1.3.3

numpy==1.21.2

gunicorn==20.1.0

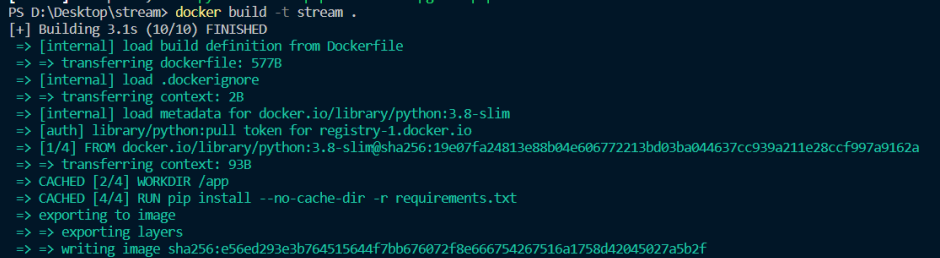
Flask==2.0.2

Werkzeug==2.0.2

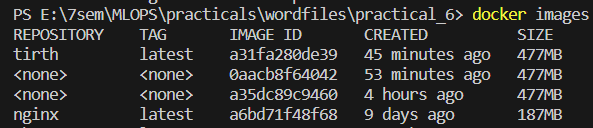
1. Create a Streamlit file :



1. Create a Docker Image :



Check the image is created or not :



**Task 4:** Run the docker container to execute the docker image and host the machine learning model using streamlit app server.



