

Docker Tutorial

Step 1: Create Project Directory

Create a folder for your app. Open VSCode or Spyder and set created folder as the working directory. Make sure Docker Desktop is running.

Step 2: Create the Python App

Create a file called app.py with the following content:

```
# ml_app.py

import streamlit as st
import pandas as pd
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier

# Load the Iris dataset
iris = load_iris()
X = pd.DataFrame(iris.data, columns=iris.feature_names)
y = pd.Series(iris.target)

# Split the dataset into training and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3,
random_state=42)

# Train a RandomForestClassifier
clf = RandomForestClassifier()
clf.fit(X_train, y_train)

# Predict the test set results
y_pred = clf.predict(X_test)

# Streamlit app
st.title("Iris Flower Prediction App")

# Create input fields for user to enter feature values
sepal_length = st.slider("Sepal Length (cm)", float(X["sepal length (cm)"].min()), float(X["sepal length (cm)"].max()), float(X["sepal length (cm)"].mean()))
sepal_width = st.slider("Sepal Width (cm)", float(X["sepal width (cm)"].min()), float(X["sepal width (cm)"].max()), float(X["sepal width (cm)"].mean()))
petal_length = st.slider("Petal Length (cm)", float(X["petal length (cm)"].min()), float(X["petal length (cm)"].max()), float(X["petal length (cm)"].mean()))
petal_width = st.slider("Petal Width (cm)", float(X["petal width (cm)"].min()), float(X["petal width (cm)"].max()), float(X["petal width (cm)"].mean()))

# Create a DataFrame for the input values
input_data = pd.DataFrame([[sepal_length, sepal_width, petal_length, petal_width]], columns=X.columns)

# Make predictions
prediction = clf.predict(input_data)
prediction_proba = clf.predict_proba(input_data)
```

```
# Display the prediction results
st.subheader("Prediction")
st.write(iris.target_names[prediction][0])
st.subheader("Prediction Probability")
st.write(prediction_proba)
```

Step 3: Create the Dockerfile

Create a file called Dockerfile (no extension!) with the following content:

```
# Use the official Python 3.9 slim image as the base for a lightweight
environment
FROM python:3.9-slim

# Set the working directory inside the container to /app
WORKDIR /app

# Copy the requirements.txt file into the container
COPY requirements.txt .

# Upgrade pip and install all Python dependencies listed in requirements.txt
RUN pip install --upgrade pip && pip install -r requirements.txt

# Copy all remaining application code and files from the current directory to
/app in the container
COPY . .

# Expose port 8501, which is the default port used by Streamlit
EXPOSE 8501

# Set the default command to run the Streamlit app on container start,
# binding it to all network interfaces on port 8501
ENTRYPOINT ["streamlit", "run", "ml_app.py", "--server.port=8501", "--server.address=0.0.0.0"]
```

Step 4: Create requirements.txt

```
streamlit
scikit-learn
pandas
```

Step 5: Build the Docker Image

Run this command in the same folder as the Dockerfile in terminal:

```
docker build -t iris-streamlit-app .
```

Step 5: Run the Docker Container

```
docker run -p 8501:8501 iris-streamlit-app
```

Your app will be accessible at <http://localhost:8501>

Export and Share as a File

Save the image to a file:

```
docker save iris-streamlit-app > iris-streamlit-app.tar
```

Send the file (via cloud storage, USB, etc.).

Recipient loads the image:

```
docker load < iris-streamlit-app.tar  
docker run -p 8501:8501 iris-streamlit-app
```

Common Docker Commands:

List running containers: `docker ps`

Stop a container: `docker stop <container_id>`

Remove a container: `docker rm <container_id>`

Remove an image: `docker rmi <image_id>`

List images: `docker images`