Experiment No. : 7 Page No. : 17

DOCKER IMPLEMENTATION

Aim: To deploy a model.py using docker.

Step 1: Creation of model.py:

```
from sk-learn.datasets import load_iris
from sklearn.ensemble import RandomForestClassifier
import Pickle

def train_model():
    data = load_iris()
    X, Y = data.data, data.target
    model = RandomForestClassifier()
    model.fit(X, Y)
    with open('model.pkl', 'wb') as f:
        Pickle.dump(model, f)
    print("model trained and saved as model.pkl")

def Predict():
    with open("model.pkl", 'rb') as f:
                model = pickle.load(f)
    test_Data = [5.1, 3.5, 1.4, 0.2]
    Prediction = Model.predict([test_Data])
    print(f"Prediction for {test_Data}: {int(
                    predict[0])}")

if __name__ == '__main__':
    train_model()
    predict()
```

Teacher's Signature : _____

Step 2: Create a requirements.txt with:
scikit-learn

Step 3: Create Dockerfile with:

```
FROM python:3.11-slim
WORKDIR /app
COPY requirements.txt requirements.txt
COPY model.py model.py
RUN pip install -r requirements.txt
CMD ["python", "model.py"]
```

Step 4: Open terminal in VS code to build a ml-model
"docker build -t ml-model"

Run model container:
"docker run ml-model"

Add tag:
"docker tag ml-model shreya124/ml-model"

push container:
"docker push shreya124/ml-model"

Get a pull request
"docker pull shreya124/ml-model"

run the image/container
"docker run shreya124/ml-model"

Teacher's Signature : _____

We can access other's images if they are Public and run them with:
" docker pull bhavkorat/ml-model "
" docker run bhavkorat/ml-model "

Result:

The above implementation of Docker is executed successfully.