Big Data Assignment 6

Srivas S ME20b175

3 April 2024

1 Introduction

FastAPI is a modern,fast framework for building APIs with python.It is designed to be high performance and uses asynchronous programming which helps it handle many concurrent connections effectively. It also has a UI called Swagger UI which we can use on the local server to run our programs. Here,we perform the experiments on the MNIST data and try to set up an API for it. We use the second model from the previous assignment, as it gave one of the highest test accuracies.

2 Using FastAPI for the MNIST dataset:

We apply the functions as described in the question document. On running the commands, and loading Swagger UI, we get the following layout:

2.1 layout

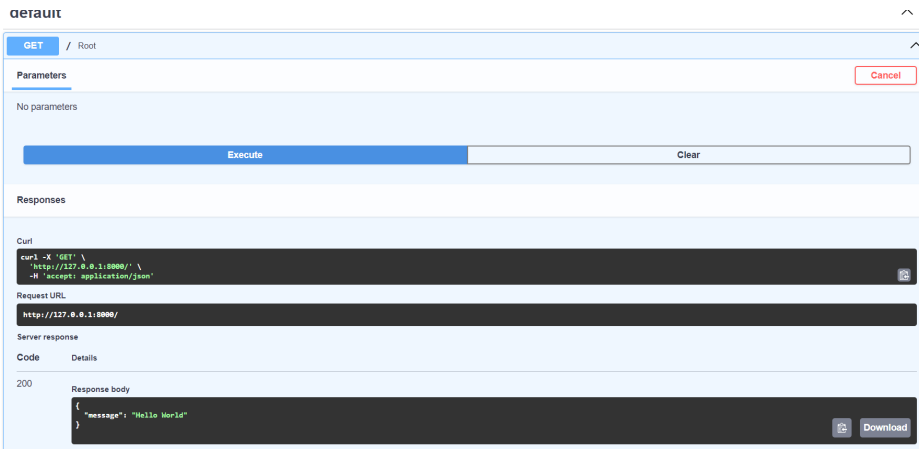

Figure 1: The layout of the system

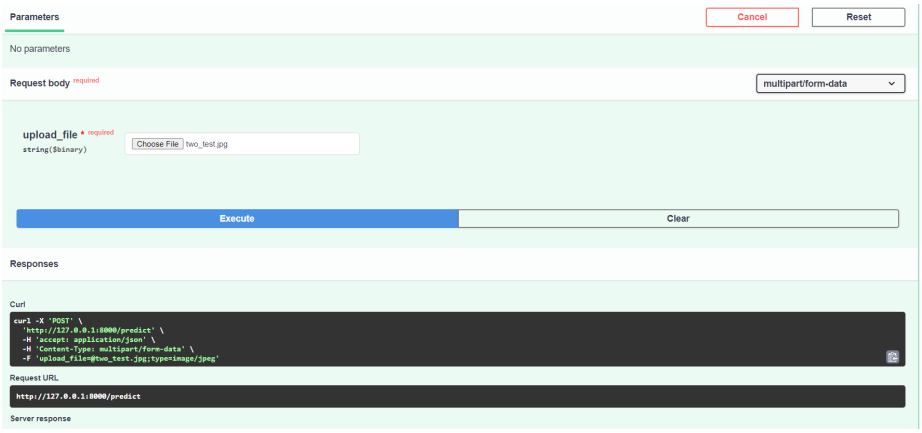

Figure 2: The second set of layout below

Figure 3: The API layout

Here, I have used the hello world program to test the system.

2.2 Testing the API initially with 28x28

We initally provide the api with a 28x28 image from MNIST. This is because we havent yet made the code for using arbitrary image size. The output is shown in the next figure:

2.3 Using resizing and entering arbitrary images from MS paint:

Now we resize the images by using image.resize((28,28)) to get arbitrarily sized images to normal size. Also we notice that the background here must be black just like the MNIST dataset. We draw in MS Paint according to that and get the following 10 results from 10 experiments done: First we try 3: Next we

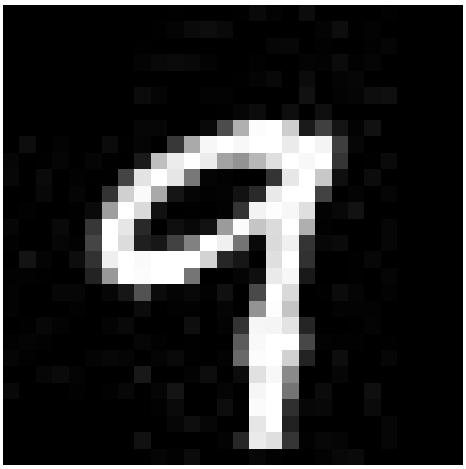

Figure 4: The image for 9 from MNIST

try 1: However, this gives us a wrong answer of 2 The third trial is on five Here the output matches correctly. Next we try four The model gets this too right Next we try 7 We see that the output for 7 is 2 which is wrong We try 8 now: We see that even this is obtained wrongly as 3 We try 2 now: We try zero next: The output erraneously comes as 5 for 0 We try 6 now: The output for an input of six is erraneously put as 8. Finally, we try for 9: This also gives a wrong answer of 3.

3 Conclusion:

We have studied various commands in mlflow such as start_run,autolog, mlflow.log_params, mlflow.log_metrics as well as nested runs. We have observed the various use cases and obtained the comparison between various models used for MNIST. We get that Model 8 which has slightly better overall test accuracy as per the figure 20. We also got an idea into variability of various metrics across various experiments.This shows the usefulness of MLflow in making various aspects of model analysis more streamlined and quicker.

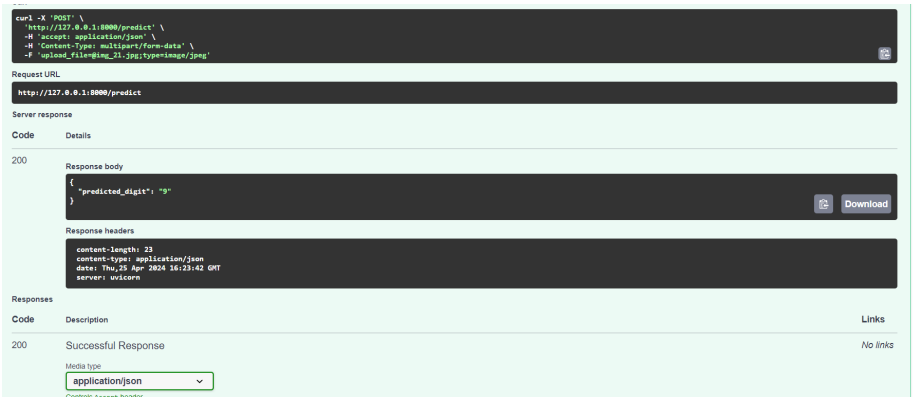

Figure 5: The output generated in the API

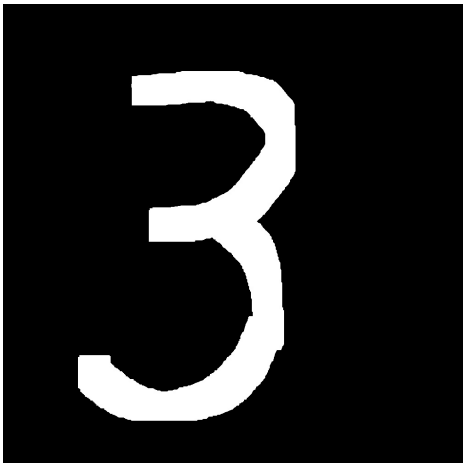

Figure 6: The input(3)

Figure 7: The output generated

Figure 8: The input(1)

Figure 9: The output given for the 1 input image

Figure 10: The input(5)

Figure 11: The output for the 5 input image

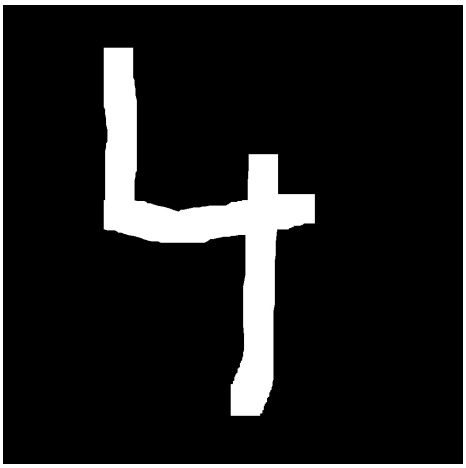

Figure 12: The input(4)

Figure 13: The output for the input 4

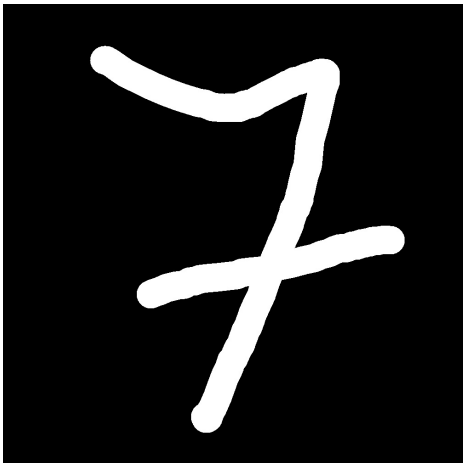

Figure 14: Input seven

Figure 15: The output for the input 7

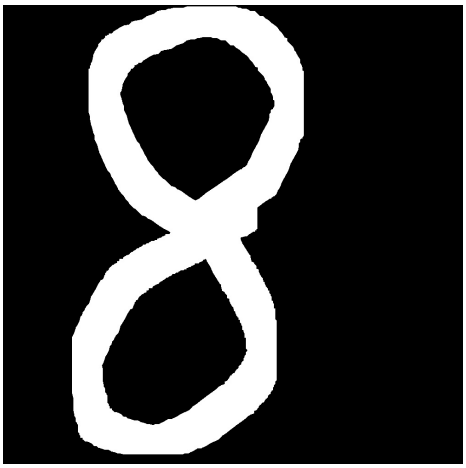

Figure 16: Input eight

Figure 17: The output for the input 8

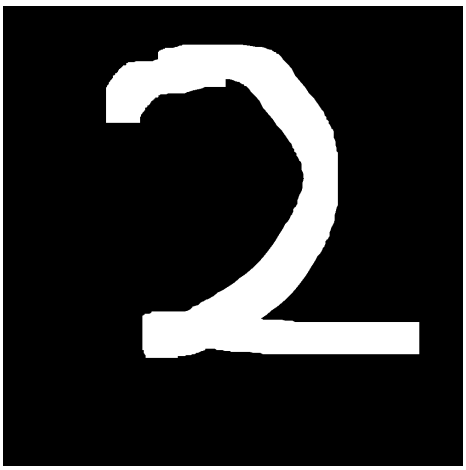

Figure 18: Input two

Figure 19: The output for the input 2

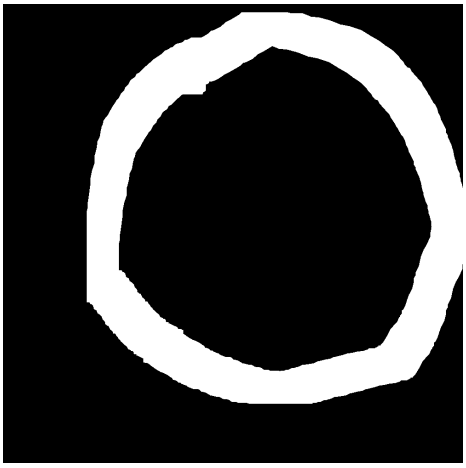

Figure 20: Input zero

Figure 21: The output for the input 0

Figure 22: Input six

Figure 23: The output for the input six(given as 8 here)

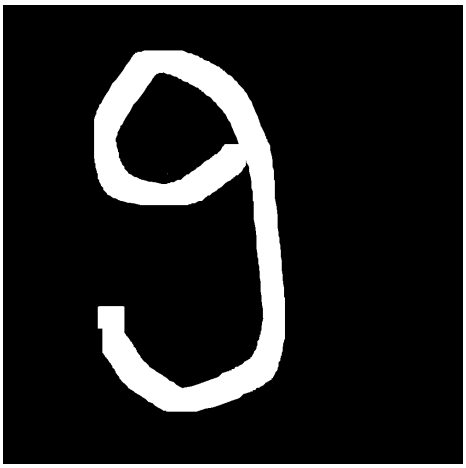

Figure 24: Input six

Figure 25: The output for the input nine