D602 Deployment Task 3 Write Up

By: Kevin Sandoval

A.

* Project was cloned to the IDE
* Commits with a message for parts B,C, and D. Two versions of each are included
* Copy of URL included in “Comments to Evaluator”
* Branch history submitted in a text file

B.

For Part B of the performance assessment I used the FastAPI documentation for docker containers as well as the YouTube video referenced in the course resources section. The YouTube video is called “Deploy ML models with FastAPI, Docker, and Heroku | Tutorial” by AssemblyAI. The video showcased a tutorial for setting up FastAPI documentation that I used when creating my API. For Part 1 of Part B, the code was mostly provided, however I changed the response message to “API Test: API is Operational” as recommended in Dr. Sewell’s video lecture. The message displays when the API is first run and shows in the tab that is created. The user can then add “/docs” to the end of the URL to test the rest of the API. In the new page, the user can see the “/predict/delays” which accepts inputs for “Order”, “Airport”, “DepartureTime”, and “ArrivalTime” as specified by the given API\_Python\_1.0.0.ipynb file given. The order should be always set to 1, but the user can add any valid airport, departure time, and arrival time to receive the average departure delay in minutes.

C.

For Part C, I created a python file called “test\_predictions” that contains tests for the API functionality. The user can load the test file, uncomment one of the tests, and run the code to check to see if the API responds appropriately. I added tests for the order, arrival airport, departure time, and arrival time. These checks are found in the code as exceptions and if the input that a user provides is incorrect, it will raise an exception that highlights the issue. There is also a test for the full correct functionality of the API, and in that case the API works as intended and provides the average departure delay in minutes.

D.

For Part D I used the same resources as in Part B. The FastAPI documentation helped create the functioning Dockerfile that fit the folder requirements that GitLAB has. Running everything through one folder and one file as opposed to a main.py file and a model.py was difficult to get up and running at first but the documentation helped guide the construction of the file. I also had to edit the requirements.txt file slightly to ensure that everything was in sync and running correctly. Once finished the Dockerfile and command prompts used create the image for the API and then starts the container used to run the API.

E.

I wrote my code by downloading the API\_Python\_1.0.0.ipynb file. Since the first example file I had to help with creation had a main.py and model.py file with specific folder layout, I decided to break up the given .ipynb file into two files. It made it easier to manipulate each file and find bugs or issues in the code. After I was finished, I combined the files back together in one singular file. I tried to name it the same “API\_Python\_1.0.0” file format, but when I ran the command to start the docker container, it had trouble reading the file name and it thought the file was “API\_Python\_1”. Because of this, it was unable to read the file. I decided to change the file name to “main” as that is typical in the documentation and then had no issues with file recognition. The only other issue I had is when I uploaded all of the files to the GitLAB and the pipeline failed. I included the fail message below as it seems to be an issue with GitLAB since my pipeline runs correctly on my local machine.

A screenshot of a computer program

Description automatically generated

F.

Panopto video showing functional API is included in the submission.

G.

AssemblyAI (July 30th, 2022). *Deploy ML models with FastAPI, Docker, and Heroku | Tutorial.* Retrieved January 25th, 2024,From <https://www.youtube.com/watch?v=h5wLuVDr0oc>

Dr. Sewell, W (n.d). *D602 Webinar # 4.* Retrieved January 20th, 2024,From D602 Course Search

<https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=0a593472-da20-4f64-8957-b21e00fd671a>

FastAPI (n.d.), *FastAPI in Containers – Docker* Retrieved January 24th, 2024, From <https://fastapi.tiangolo.com/deployment/docker/>

Wells, R (June 27th, 2018) *Unit Testing and Test Driven Development in Python* Retrieved January 28th, 2024 From <https://www.linkedin.com/learning/unit-testing-and-test-driven-development-in-python/what-is-unit-testing?resume=false&u=2045532>

WGU. (n.d). *Create your GitLab course specific branch OLD 1.* Retrieved December 29th, 2024 From <https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=b28d1026-3889-48de-881f-b22100338f5c>