Gaby Masak

D602 – Deployment

Task 2: Data Production Pipeline

10/26/2024

Explanation: Code for Data Production Pipeline

**Requirement A: Gitlab Subgroup and Project**

Link to GitLab Repository: <https://gitlab.com/wgu-gitlab-environment/student-repos/gmasak/d602-deployment-task-2/-/tree/working_branch>

A screenshot of a computer

Description automatically generated

Figure 1: Screenshot of Repository Branch History

**Requirement B: Python Script to Import and Format Airport Data**

Gitlab Link to Version 1: <https://gitlab.com/wgu-gitlab-environment/student-repos/gmasak/d602-deployment-task-2/-/commit/043e06fabaaa4abea0b7de26cc7d73cfae3dc9fd>

GitLab Link to Version 2: <https://gitlab.com/wgu-gitlab-environment/student-repos/gmasak/d602-deployment-task-2/-/commit/118a3154358916df5b5bab272db9bf17de6575e4>

GitLab Link to Final Version: <https://gitlab.com/wgu-gitlab-environment/student-repos/gmasak/d602-deployment-task-2/-/commit/e904c6003152c1ff5087734f31ab9b86f522677d#d406fb1394122462ad4731706bd8d2a7d4a7725d>

Screenshot of History:

A screenshot of a computer

Description automatically generated

Figure 2: Screenshot of Commits for Requirement B

A screenshot of a computer

Description automatically generated

Figure 3: Screenshot Showing Changes

**Requirement C: Python Script to Filter Data**

Gitlab Link to Version 1: <https://gitlab.com/wgu-gitlab-environment/student-repos/gmasak/d602-deployment-task-2/-/commit/118a3154358916df5b5bab272db9bf17de6575e4>

GitLab Link to Version 2: <https://gitlab.com/wgu-gitlab-environment/student-repos/gmasak/d602-deployment-task-2/-/commit/e904c6003152c1ff5087734f31ab9b86f522677d#d406fb1394122462ad4731706bd8d2a7d4a7725d>

Screenshot of History:

A screenshot of a computer

Description automatically generated

Figure 4: Screenshot Showing Commits for Requirement C

A screenshot of a computer

Description automatically generated

Figure 5: Screenshot Showing Changes for Requirement C

**Requirement D: MLFlow Experiment Implementation**

Gitlab Link to Version 1: <https://gitlab.com/wgu-gitlab-environment/student-repos/gmasak/d602-deployment-task-2/-/commit/d25e9337453053d4b69e7184ebe706c1662b66c1>

GitLab Link to Version 2: <https://gitlab.com/wgu-gitlab-environment/student-repos/gmasak/d602-deployment-task-2/-/commit/e904c6003152c1ff5087734f31ab9b86f522677d#d406fb1394122462ad4731706bd8d2a7d4a7725d>

Screenshot of History:

A screenshot of a computer

Description automatically generated

Figure 6: Screenshot Showing Commits for Requirement D

A screenshot of a computer

Description automatically generated

Figure 7: Screenshot Showing Changes for Requirement D

**Requirement E: MLProject File to Link Previous Scripts**

Gitlab Link to Version 1: <https://gitlab.com/wgu-gitlab-environment/student-repos/gmasak/d602-deployment-task-2/-/commit/e904c6003152c1ff5087734f31ab9b86f522677d#d406fb1394122462ad4731706bd8d2a7d4a7725d>

GitLab Link to Version 2: <https://gitlab.com/wgu-gitlab-environment/student-repos/gmasak/d602-deployment-task-2/-/commit/a8702c9db86d87f047863674a69faf0103a91b9a>

Screenshot of History:

A screenshot of a phone

Description automatically generated

Figure 8: Screenshot Showing Commits for Requirement E

A screenshot of a computer

Description automatically generated

Figure 9: Screenshot Showing Changes for Requirement E

**Requirement F: Results**

A screen shot of a computer

Description automatically generated

Figure 10: Screenshot Showing Successful Pipeline Run in Terminal

A screenshot of a computer

Description automatically generated

Figure 11: Screenshot Showing MLFlow UI After Successful Run

A screenshot of a computer

Description automatically generated

Figure 12: Screenshot Showing "Finished" Status After Successful Run

The code was written in segments. First libraries and modules are imported. Next, for requirement B, the downloaded CSV file is imported. I then looked at the provided poly\_regressor file to determine which columns would be needed. I used this information to filter the dataframe to reflect only these columns and subsequently updated the names of the columns to match those in the poly\_regressor file. Regarding challenges encountered, there was some confusion regarding which delay column to use. Both departure and arrival delay columns featured two similar columns: DepDelay/DepDelayMinutes and ArrDelay/ArrDelayMinutes respectively. The main difference between the columns was the presence of negative values if the flight left early. The latter option would zero out any negative numbers, considering early flights to be on time. Ultimately, it was clear that the minutes variations should be used as only the delays should be tracked. Both variations of these files are exported into CSV files. I then moved on to the DVC requirements, where I ran into issues as simple terminal commands like pip threw errors. After troubleshooting, I had to install Python, then make sure to add it as an environmental variable. I then used the terminal to install and initialize DVC then add the dataset to the DVC using the following commands:

pip install dvc

dvc --version

dvc add C:\Users\gabri\PycharmProjects\d602-deployment-task-2\cleaned\_data.csv

git add cleaned\_data.csv.dvc

git commit -m "Add DVC metafile for cleaned\_data.csv"

git push origin working\_branch

I then added code to log the artifact for the mlflow pipeline. To check, I needed to execute the following commands in the terminal:

pip install mlflow

python import\_data.py

mlflow ui

Requirement C called for several manipulations of the dataset. First the data is filtered to only include rows that have ‘ATL’ in the origin airport column. Two data cleaning steps were taken: removal of duplicates and removal of missing values. The latter removed canceled flights from the dataframe, as these rows lacked departure and arrival times. I then added logging to the script for next requirement.

Requirement D involved modifying the supplied poly\_regressor\_Python\_1.0.0.py file. I added several lines inside of the relevant mlflow experiment to log the relevant artifacts, metrics, and models. I had several issues and troubleshooted, wherein I installed Anaconda, added its environmental variables to the path, moved the clean\_data() and import\_data() function calls out of the MLFlow experiment in Requirement D, and forced a commit after issues with Git.

A screenshot of a computer

Description automatically generated

Figure 13: Screenshot of 404 Error

I also had issues with cloning the project to my IDE, as the provided instructions led to a 404 error, seen in Figure 13. I was able to fix this by troubleshooting how to create a personal access token, which had recently expired, and copying this new token into the box instead of pressing the generate button as suggested on the instructions. Once this was completed, the project was cloned to my IDE and I was able to complete the requirements.