QBN1 — QBN1 TASK 2: DATA PRODUCTION PIPELINE

DEPLOYMENT — D602
PRFA — QBN1

Preparation Task Overview Submissions Evaluation Report

COMPETENCIES

4162.1.2 : Implements a Data Product Pipeline

The learner implements a data product pipeline to address organizational needs.

INTRODUCTION

As a data analyst, you may be responsible for designing and executing a data pipeline that effectively addresses specific organizational requirements.

In this task, you will write code in Python or R that downloads and filters a selected airport's data. You will then run an MLFlow experiment capturing the features listed in the comments within the poly_regressor file. Next, you will write an MLProject file that links the two scripts. Finally, you will provide an explanation of how you wrote your code.

SCENARIO

You are an analyst at a major airline whose work involves predicting delays from various airports that your airline serves. To capture both current and future needs, a previous analyst developed a polynomial regression to predict the average departure delay in minutes between a given departure airport and all the airports that are served from that departure airport. They tested the model on a single airport but then left the company abruptly, leaving you with the task of finishing their work and deploying the model to other business units within the airline.

You have been provided with the previous analyst's regression model and the Bureau of Transportation Statistics website they used to download the initial data that was used to train the model. You will need to download data for one of the airports listed at the end of this section and test the regression model on that airport. To capture the performance of the model, you will implement MLFlow experiments to record the various parameters with which you test the model and the performance under each set of conditions. You will then combine three steps of model testing into a single ML pipeline such that other analysts can extend your work in other business units.

Your final code and all updates must be stored in a GitLab repository so users in other business units can track the changes you made to the code over time.

Help

Airport list:

- ATL: Atlanta, Georgia
- LAX: Los Angeles, California
- JFK: New York, New York
- MIA: Miami, Florida
- DFW: Dallas, Texas
- ORD: Chicago, Illinois

REQUIREMENTS

Your submission must represent your original work and understanding of the course material. Most performance assessment submissions are automatically scanned through the WGU similarity checker. Students are strongly encouraged to wait for the similarity report to generate after uploading their work and then review it to ensure Academic Authenticity guidelines are met before submitting the file for evaluation. See Understanding Similarity Reports for more information.

Grammarly Note:
Professional Communication will be automatically assessed through Grammarly for Education in most performance assessments before a student submits work for evaluation. Students are strongly encouraged to review the Grammarly for Education feedback prior to submitting work for evaluation, as the overall submission will not pass without this aspect passing. See Use Grammarly for Education Effectively for more information.

Microsoft Files Note:
Write your paper in Microsoft Word (.doc or .docx) unless another Microsoft product, or pdf, is specified in the task directions. Tasks may not be submitted as cloud links, such as links to Google Docs, Google Slides, OneDrive, etc. All supporting documentation, such as screenshots and proof of experience, should be collected in a pdf file and submitted separately from the main file. For more information, please see Computer System and Technology Requirements.

You must use the rubric to direct the creation of your submission because it provides detailed criteria that will be used to evaluate your work. Each requirement below may be evaluated by more than one rubric aspect. The rubric aspect titles may contain hyperlinks to relevant portions of the course.

A. Create your subgroup and project in GitLab using the provided web link and the "GitLab How-To" web link by doing the following:

- Clone the project to the IDE.
- Commit with a message and push when you complete each requirement listed in parts B, C, D, and E.

Note: You may commit and push whenever you want to back up your changes, even if a requirement is not yet complete.

- Submit a copy of the GitLab repository URL in the "Comments to Evaluator" section when you submit this assessment.
- Submit a copy of the repository branch history retrieved from your repository, which must include the commit messages and dates.

B. Write a script in either Python or R to import the data you downloaded and format it according to the criteria required by the model script, demonstrating a progression of work on your code. You must run a DVC command to create a metafile for your dataset and submit the metafile to the GitLab repository. Submit *at least* **two** versions of your code to the GitLab repository demonstrating a progression of work on your code.

Note: Include the original CSV file you downloaded in your GitLab repository. Although data files are not typically stored in GitLab repositories, including this file in your submission is necessary to allow evaluators to fully assess your work.

C. Write a script in either Python or R to filter data to only departures from the chosen airport, then implement *at least* **two** other data cleaning steps. Submit *at least* **two** versions of your code to the GitLab repository demonstrating a progression of work on your code.

D. Using the code template provided in the GitLab repository, implement an MLFlow experiment that captures the features listed in the comments within the poly_regressor file for either Python or R. Submit *at least* **two** versions of your code to the GitLab repository demonstrating a progression of work on your code.

E. Using the provided YAML file for either Python or R, write an MLProject file that links the two scripts you wrote in parts B and C with the modified poly_regressor script from part D.

Note: Your MLProject file must be submitted to the GitLab repository.

F. Provide a written explanation of how you wrote the code and MLProject pipeline, including any challenges you encountered and how you addressed these challenges. Include a screenshot of your MLProject pipeline running successfully.

G. Acknowledge sources, using in-text citations and references, for content that is quoted, paraphrased, or summarized.

H. Demonstrate professional communication in the content and presentation of your submission.

File Restrictions

File name may contain only letters, numbers, spaces, and these symbols: ! - _ . * ' ( )
File size limit: 200 MB
File types allowed: doc, docx, rtf, xls, xlsx, ppt, pptx, odt, pdf, csv, txt, qt, mov, mpg, avi, mp3, wav, mp4, wma, flv, asf, mpeg, wmv, m4v, svg, tif, tiff, jpeg, jpg, gif, png, zip, rar, tar, 7z

RUBRIC

A:GITLAB REPOSITORY

NOT EVIDENT	APPROACHING COMPETENCE	COMPETENT
A GitLab repository is not provided.	The subgroup and project are created in GitLab, but 1 or more of the given actions are not completed, or they are completed incorrectly.	The subgroup and project are created in GitLab correctly, and all of the given actions are completed correctly.

B:IMPORT AND FORMAT SCRIPT

NOT EVIDENT	APPROACHING COMPETENCE	COMPETENT