

D602 - Deployment

QBN1 Task 2: Data Production Pipeline














































Prepared by Shanay Murdock


































A. GitLab Repository

GitLab URL:

https://gitlab.com/wgu-gitlab-environment/student-repos/smurd32/d602-deployment-task-2/-/tree/task-2-working-branch?ref_type=heads

Screenshot of repository branch commit history with dates and comments

task-2-working-bra...	d602-deployment-task-2	Author ▾	Browse files	Search by message	🔍
Jul 29, 2025					
	Add main.py and modify MLProject file Shanay Murdock authored 40 seconds ago	5ba5a08c			
	Update run_pipeline command Shanay Murdock authored 23 minutes ago	3500c66b			
	Commit output of MLflow run Shanay Murdock authored 58 minutes ago	172078e9			
	D. Log Scikit-Learn Ridge Regression model Shanay Murdock authored 2 hours ago	e14dbd10			
	Create MLProject file with entry points Shanay Murdock authored 2 hours ago	6f93417c			
	Remove unused versions of poly regressor file Shanay Murdock authored 2 hours ago	e3a7d90c			
	D. Log the model performance metrics Shanay Murdock authored 2 hours ago	e181e1e4			
	D. Log the performance plot Shanay Murdock authored 2 hours ago	6346827b			
	D. Add log of input parameters Shanay Murdock authored 2 hours ago	7d160537			
	D. Add informational log Shanay Murdock authored 2 hours ago	48f6be86			
	Track CSV with DVC Shanay Murdock authored 2 hours ago	a13cec5e			
	C. Filter for departures out of Atlanta (ATL) and export cleaned dataset Shanay Murdock authored 2 hours ago	539b5da0			
	C. Filter for Atlanta (ATL) departures Shanay Murdock authored 2 hours ago	a7488fe2			
	C. Check for missing value types, fill missing values, and make necessary dtype conversions Shanay Murdock authored 3 hours ago	1d7f905e			
	Update .gitignore to ignore intermediary CSV file Shanay Murdock authored 3 hours ago	6057dd82			

 C. Load CSV to DataFrame and check for duplicate rows Shanay Murdock authored 3 hours ago	7398cb3b		
 Update README.md with new file Shanay Murdock authored 3 hours ago	48d6312a		
 B. Fix output in file docstring Shanay Murdock authored 3 hours ago	4c17f673		
 B. Export formatted DataFrame to CSV Shanay Murdock authored 3 hours ago	19724702		
 B. Export formatted DataFrame to CSV Shanay Murdock authored 3 hours ago	80534baa		
 B. Rename columns to prepare for polynomial regression model Shanay Murdock authored 18 hours ago	f44dd261		
 Update README.md with list of files Shanay Murdock authored 18 hours ago	65cbb3a9		
 B. Select columns to keep from raw data file Shanay Murdock authored 19 hours ago	bb5538de		
Jul 28, 2025			
 B. Create import_and_format.py file and load raw data as DF from CSV Shanay Murdock authored 22 hours ago	4d1be28e		
 Add "pass" to incomplete function to prevent runtime errors Shanay Murdock authored 23 hours ago	050b758b		
 Remove R files unnecessary for Python project Shanay Murdock authored 23 hours ago	14096212		

B. Import and Format Script

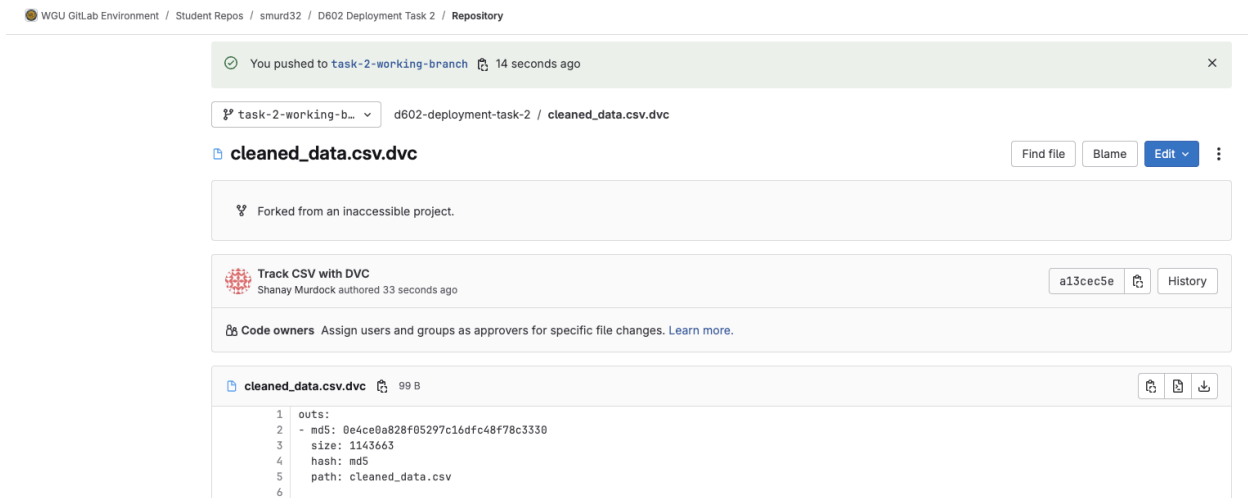
For Part B, I wrote a Python script called `import_and_format.py` that imported the raw CSV data: `T_ONTIME_REPORTING.csv` from the Bureau of Transportation Statistics. This script inspected the raw data, selected which columns to keep (as anyone downloading the data can choose any of the columns to add to the generated CSV) and filtered strictly to keep the columns necessary for the polynomial regression model that comes in Part D.

From there, I renamed the columns to match what the polynomial regression file asked for and exported the data from a DataFrame format to a CSV and called it `formatted_flight_data.csv` because it's an intermediary cleaning step and I didn't want it getting confused as the fully cleaned data.

See screen shots below for the script:

```
1  #!/usr/bin/env python
2  # coding: utf-8
3
4  """
5  Import and format flight data from the original CSV file.
6
7  This script reads the raw flight data, selects necessary columns,
8  renames them according to the expected format, and saves the cleaned
9  data to a new CSV file.
10
11  Output:
12      formatted_flight_data.csv: CSV file containing the formatted data.
13  """
14
15  import pandas as pd
16
17  df = pd.read_csv('T_ONTIME_REPORTING.csv')
18  #print(df.head().T) # Observe first five rows
19  #print(df.shape) # (54562, 19) - 12 columns expected by poly_regressor
20  # print(df.info()) # Observe data types and non-null counts
21
22  # Select required columns
23  # Note: This provides flexibility for reproducibility if someone downloads
24  # extra data in the raw dataset.
25  columns_to_keep = [
26      'YEAR',
27      'MONTH',
28      'DAY_OF_MONTH',
29      'DAY_OF_WEEK',
30      'ORIGIN',
31      'DEST',
32      'CRS_DEP_TIME',
33      'DEP_TIME',
34      'DEP_DELAY',
35      'CRS_ARR_TIME',
36      'ARR_TIME',
37      'ARR_DELAY'
38  ]
39  # Filter to keep only required columns
40  df = df[columns_to_keep]
41  # print(df.shape) # (54562, 12) - 12 columns expected by poly_regressor
42
43  # Define column mapping from original to expected format
44  column_mapping = {
45      'YEAR': 'YEAR',
46      'MONTH': 'MONTH',
47      'DAY_OF_MONTH': 'DAY',
48      'DAY_OF_WEEK': 'DAY_OF_WEEK',
49      'ORIGIN': 'ORG_AIRPORT',
50      'DEST': 'DEST_AIRPORT',
51      'CRS_DEP_TIME': 'SCHEDULED_DEPARTURE',
52      'DEP_TIME': 'DEPARTURE_TIME',
53      'DEP_DELAY': 'DEPARTURE_DELAY',
54      'CRS_ARR_TIME': 'SCHEDULED_ARRIVAL',
55      'ARR_TIME': 'ARRIVAL_TIME',
56      'ARR_DELAY': 'ARRIVAL_DELAY'
57  }
58
59  # Replace column titles with column_mapping
60  df.rename(columns=column_mapping, inplace=True)
61
```

Screenshot of `cleaned_data.csv.dvc` . This is the metadata file generated by DVC, which was done after Part C where the data was cleaned:



C. Data Filtering Script

For Part C, I wrote a Python script called `clean_and_filter.py` that imported the output from Part B, `formatted_flight_data.csv`, and loaded it back into a DataFrame. From there, I further inspected the data, and checked for duplicate rows by dropping any duplicates or logging that no duplicates were found. I checked for missing data and found that data was missing from four numeric columns, so I filled the values with `0` in order to allow for type conversions in the next step. I converted any float columns to integer columns as appropriate for the data. Then I filtered the data for any flights originating out of Atlanta (ATL). Lastly, I exported the DataFrame to `cleaned_data.csv` for it to be picked up by the modified `poly_regressor_Python_1.0.0.py` file.

See screenshot of script below:

 clean_and_filter.py  1.47 KiB

```
1  #!/usr/bin/env python
2  # coding: utf-8
3
4  """
5  clean_and_filter.py - Flight data filtering and cleaning script
6
7  This script imports the formatted flight data CSV file and filters and cleans it
8  according to the requirements of the polynomial regression model.
9
10 Input: formatted_flight_data.csv (formatted flight data)
11 Output: cleaned_data.csv (formatted for model consumption)
12         cleaned_data.csv.dvc (DVC metafile)
13 """
14
15 # Load libraries
16 import pandas as pd
17
18 # Load the formatted flight data CSV file
19 df = pd.read_csv('formatted_flight_data.csv')
20
21 # Check for duplicate rows
22 duplicates = df.duplicated().sum()
23 if duplicates > 0:
24     print(f"Found {duplicates} duplicate rows. Removing duplicates.")
25     df = df.drop_duplicates()
26 else:
27     print("No duplicate rows found.")
28
29 # Check for missing values
30 # print(df.isnull().sum())
31 # Fill missing float or integer values with 0
32 df.fillna(0, inplace=True)
33
34 # Check data types of each column
35 # print(df.info())
36 # Convert floats to integers where appropriate
37 float_columns = ['DEPARTURE_TIME', 'DEPARTURE_DELAY', 'ARRIVAL_TIME', 'ARRIVAL_DELAY']
38 df[float_columns] = df[float_columns].astype('int64')
39
40 # Filter for Atlanta (ATL) departures
41 # Print formatted data shape
42 # print(f"Formatted data shape:\n\tRows: {df.shape[0]}\n\tColumns: {df.shape[1]}")
43 df = df[df['ORG_AIRPORT'] == 'ATL']
44 # Print formatted data shape
45 # print(f"Filtered data shape:\n\tRows: {df.shape[0]}\n\tColumns: {df.shape[1]}")
46
47 # Save filtered data
48 df.to_csv('cleaned_data.csv', index=False)
```

D. MLflow Experiment

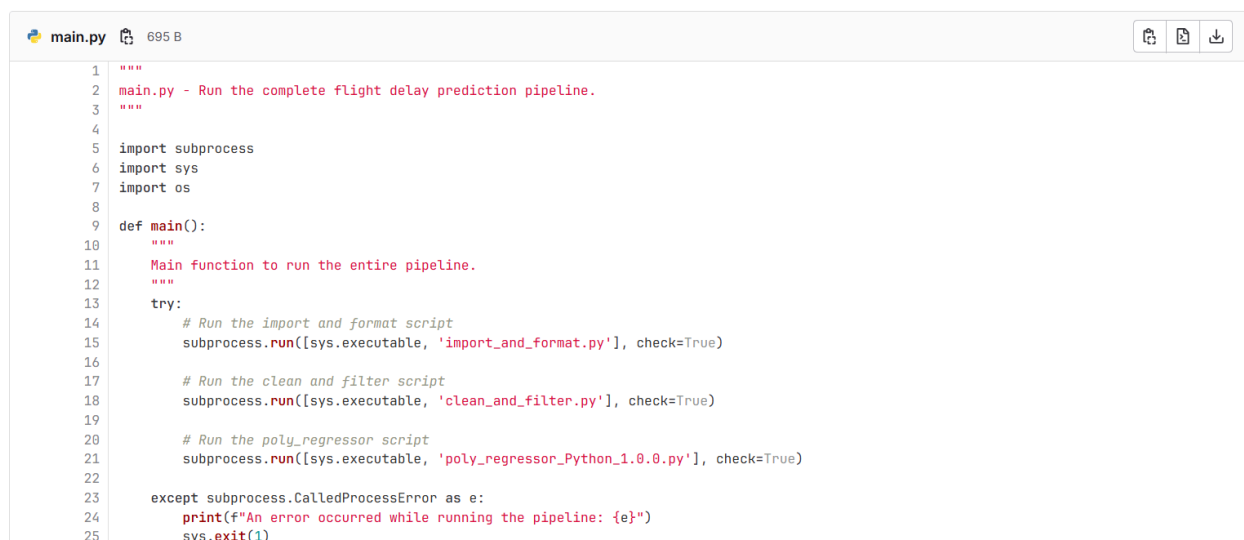
For `poly_regressor_Python_1.0.0.py`, I simply added the tracking of the artifacts, parameters, and metrics as per the instructions.

```

354
355
356 # TO DO: create an MLFlow run within the current experiment that logs the following as artifacts, parameters,
357 # or metrics, as appropriate, within the experiment:
358 # 1. The informational log files generated from the import_data and clean_data scripts
359 # 2. the input parameters (alpha and order) to the final regression against the test data
360 # 3. the performance plot
361 # 4. the model performance metrics (mean squared error and the average delay in minutes)
362
363 with mlflow.start_run(experiment_id = experiment.experiment_id, run_name = "Final Model - Test Data"):
364     # YOUR CODE GOES HERE
365     # 1. The informational log files generated from the import_data and clean_data scripts
366     mlflow.log_artifact("polynomial_regression.txt")
367     mlflow.sklearn.log_model(ridge, "final_model")
368
369     # 2. the input parameters (alpha and order) to the final regression against the test data
370     mlflow.log_param("alpha", parameters[0])
371     mlflow.log_param("order", parameters[1])
372
373     # 3. the performance plot
374     mlflow.log_artifact("model_performance_test.jpg")
375
376     # 4. the model performance metrics (mean squared error and the average delay in minutes)
377     mlflow.log_metric("mean_squared_error", score)
378     mlflow.log_metric("average_delay_minutes", np.sqrt(score))
379
380 mlflow.end_run()
381
382 logging.shutdown()

```

Below is a screenshot of `main.py`, which links `import_and_format.py`, `clean_and_filter.py`, and `poly_regressor_Python_1.0.0.py` for the `MLProject` file.



```

main.py 695 B
1 """
2 main.py - Run the complete flight delay prediction pipeline.
3 """
4
5 import subprocess
6 import sys
7 import os
8
9 def main():
10     """
11     Main function to run the entire pipeline.
12     """
13     try:
14         # Run the import and format script
15         subprocess.run([sys.executable, 'import_and_format.py'], check=True)
16
17         # Run the clean and filter script
18         subprocess.run([sys.executable, 'clean_and_filter.py'], check=True)
19
20         # Run the poly_regressor script
21         subprocess.run([sys.executable, 'poly_regressor_Python_1.0.0.py'], check=True)
22
23     except subprocess.CalledProcessError as e:
24         print(f"An error occurred while running the pipeline: {e}")
25         sys.exit(1)

```

E. MLProject Linking File

The `MLProject` file provides the project structure with entry points for the main pipeline and individual components.

See the screenshot below:



Add main.py and modify MLProject file

Shanay Murdock authored 6 hours ago



Code owners Assign users and groups as approvers for specific file changes. [Learn more.](#)



MLProject 410 B

```

1 name: flight_delay_prediction
2 conda_env: pipeline_env.yaml
3
4 entry_points:
5     main:
6         command: 'python main.py'
7
8     import_data:
9         command: 'python import_and_format_data.py'
10
11     clean_data:
12         command: 'python clean_and_filter.py'
13
14     train_model:
15         parameters:
16             num_alphas: { type: int, default: 20 }
17         command: 'python poly_regressor_Python_1.0.0.py {num_alphas}'
18

```

See the below screenshot to see the successful run of MLflow:

```

● (pipeline_env) shanaymurdock@wc-dhcp25d029 d602-deployment-task-2 % mlflow run . -e main
/opt/anaconda3/envs/pipeline_env/lib/python3.12/site-packages/mlflow/utils/requirements_utils.py:20: UserWarning: pkg_resources is deprecated as an API. See https://setuptools.pypa.io/en/latest/pkg_resources.html. The pkg_resources package is slated for removal as early as 25-11-30. Refrain from using this package or pin to Setuptools<81.
  import pkg_resources # noqa: TID251
2025/07/29 14:59:23 INFO mlflow.utils.conda: Conda environment mlflow-c751e9444d9934631bb32d0bcefb3e7fe6d6a109 already exists.
2025/07/29 14:59:23 INFO mlflow.projects.utils: == Created directory /var/folders/mm/_tnhnc3525q9gjbq0kn3d74w0000gn/T/tmp2fhy15dw for nloading remote URIs passed to arguments of type 'path' ==
2025/07/29 14:59:23 INFO mlflow.projects.backend.local: == Running command 'source /opt/anaconda3/bin/./etc/profile.d/conda.sh && con activate mlflow-c751e9444d9934631bb32d0bcefb3e7fe6d6a109 1>&2 && python main.py' in run with ID '3a71d66552d84d16bd71cd297810e0ec' ==
2025/07/29 14:59:23 INFO mlflow.projects: == Run (ID '3a71d66552d84d16bd71cd297810e0ec') succeeded ==
○ (pipeline_env) shanaymurdock@wc-dhcp25d029 d602-deployment-task-2 %

```

F. Explanation (Challenges and Solutions)

Challenge: Understanding DVC as a new concept and where to implement it. I had committed the file using Git and had to learn how to remove the tracking from Git.

Solution: Install and import DVC (Tran 2025, pp. 185-187). Remove the tracked file from Git, upload the original file to Git but commit the cleaned data file to DVC so that the cleaned data can be tracked during the model run.

Challenge: Formatting MLProject file

Solution: I had to understand it's a yaml file with no file extension. I formatted the file contents based on a lesson from *Introduction to MLflow* course on DataCamp (Bassler, n.d.).

Challenge: MLflow server already in use.

Solution: The instructions provided didn't cover what to do in the event that I had to walk away from the project and come back to it. I kept getting errors that the server was already in use but the MLflow UI wasn't working if I just went back to the site. I found an article with a command line script for stopping the server but the prompt didn't work. I had to restart the computer and reactivate environment.

G. Resources

Bassler, W. (n.d.) *Introduction to MLflow*. DataCamp.

<https://app.datacamp.com/learn/courses/introduction-to-mlflow>

Tran, K. (2025) *Production Ready Data Science: From Prototyping to Production with Python*. Self published.

WGU Official Course Resources