

Task 2:

Data Production Pipeline

Keniyah Chestnut

Deployment — D602

SID: 012601305

A: GITLAB REPOSITORY

I created a subgroup and project in GitLab using the provided link. I cloned the repo to my computer, worked in it locally, and pushed my changes as I completed each part of the task. At first, Git wasn't recognized in my terminal, so I had to install it and set up my Git config with my name and email. Once I got that working, I was able to commit and push everything without issues.

The screenshot displays the GitLab interface for a repository named 'd602-deployment-task-2'. The top navigation bar includes links for 'NEW ORIGIN', 'COMMIT', 'PUSH', 'CHECKOUT', 'DIFF', 'PIPELINE', 'JOB', 'WIKI', and 'ISSUES'. Below the navigation bar, the repository name and a dropdown menu for 'main' are visible. A search bar for 'Search by message' is also present. The main content area shows a list of commits, each with a commit icon, a message, the author, the time ago, and a commit hash. The commits are grouped by date: May 26, 2025 and May 25, 2025. The commit messages include 'Track T_ONTIME_REPORTING.csv using DVC', 'Dataset', 'This is Step B version 1 for Task 2.', 'This is my Code (Second Script) for E in Task 2.', 'This is my code for E in Task 2.', 'This is my code through step D in Task #2.', 'This is my code through step B in Task #2.', and 'This is my code through step B in Task #2.'. The commit hashes are displayed in a box next to each commit. Below the commit list, there is a section for 'Pipeline' and 'Update .gitlab-ci.yml file', both of which are marked as successful with green checkmarks. The pipeline status is shown as '26f8d6d2' and '9ba616e6'. The 'Update .gitlab-ci.yml file' section shows a commit hash of 'd018de92'. Below the pipeline section, there is a section for 'model', 'JSON file', 'Track T_ONTIME_REPORTING.csv using DVC', 'Dataset', 'This is Step B version 1 for Task 2.', 'This is my Code (Second Script) for E in Task 2.', 'This is my code for E in Task 2.', 'This is my code through step D in Task #2.', 'This is my code through step B in Task #2.', and 'This is my code through step B in Task #2.'. The commit hashes are displayed in a box next to each commit. Below the commit list, there is a section for 'moved table structure for clarity' and 'Merge branch 'working_branch' into 'main''. The commit hashes are displayed in a box next to each commit.

Commit Message	Author	Time Ago	Commit Hash
Track T_ONTIME_REPORTING.csv using DVC	Keniyah Chestnut	44 minutes ago	5cf6b2c8
Track T_ONTIME_REPORTING.csv using DVC	Keniyah Chestnut	47 minutes ago	13672492
Dataset	Keniyah Chestnut	49 minutes ago	f39fb98c
This is Step B version 1 for Task 2.	Keniyah Chestnut	1 hour ago	d9cdf82c
This is my Code (Second Script) for E in Task 2.	Keniyah Chestnut	1 hour ago	b5fe92cd
This is my code for E in Task 2.	Keniyah Chestnut	1 hour ago	c7736d94
This is my code through step D in Task #2.	Keniyah Chestnut	1 hour ago	57465a67
This is my code through step B in Task #2.	Keniyah Chestnut	7 hours ago	e1fac3f8
This is my code through step B in Task #2.	Keniyah Chestnut	7 hours ago	4876fb52
Pipeline	Keniyah Chestnut	13 hours ago	26f8d6d2
Update .gitlab-ci.yml file	Keniyah Chestnut	13 hours ago	9ba616e6
this is Step D for Task #2.	Keniyah Chestnut	13 hours ago	d018de92
model	Keniyah Chestnut	13 hours ago	e3e383a8
JSON file	Keniyah Chestnut	14 hours ago	9708d343
Track T_ONTIME_REPORTING.csv using DVC	Keniyah Chestnut	15 hours ago	5cf6b2c8
Track T_ONTIME_REPORTING.csv using DVC	Keniyah Chestnut	15 hours ago	13672492
Dataset	Keniyah Chestnut	15 hours ago	f39fb98c
This is Step B version 1 for Task 2.	Keniyah Chestnut	16 hours ago	d9cdf82c
This is my Code (Second Script) for E in Task 2.	Keniyah Chestnut	16 hours ago	b5fe92cd
This is my code for E in Task 2.	Keniyah Chestnut	16 hours ago	c7736d94
This is my code through step D in Task #2.	Keniyah Chestnut	17 hours ago	57465a67
This is my code through step B in Task #2.	Keniyah Chestnut	22 hours ago	e1fac3f8
This is my code through step B in Task #2.	Keniyah Chestnut	22 hours ago	4876fb52
moved table structure for clarity	Emilio Miller	5 months ago	c90e6199
Merge branch 'working_branch' into 'main'			c90e6199

B: IMPORT AND FORMAT SCRIPT

I wrote two versions of my import script. The first version just loads the CSV and filters down to the required columns. The second version cleans the data and renames the columns to match what the model script expects. I also used DVC to track the original CSV file and committed the .dvc file to GitLab.

Version 1:

```
[9]: import pandas as pd

# Load the raw data
df = pd.read_csv("T_ONTIME_REPORTING.csv")

# Select relevant columns only
columns_to_keep = [
    'YEAR', 'MONTH', 'DAY_OF_MONTH', 'DAY_OF_WEEK',
    'ORIGIN_AIRPORT_ID', 'DEST_AIRPORT_ID',
    'CRS_DEP_TIME', 'DEP_TIME', 'DEP_DELAY',
    'CRS_ARR_TIME', 'ARR_TIME', 'ARR_DELAY'
]

df = df[columns_to_keep]

# Preview
df.head()
```

```
[9]:  YEAR  MONTH  DAY_OF_MONTH  DAY_OF_WEEK  ORIGIN_AIRPORT_ID  DEST_AIRPORT_ID  CRS_DEP_TIME  DEP_TIME  DEP_DELAY
0   2025      1             1             3             10135             11057             606      556.0        -10
1   2025      1             1             3             10135             11057             1219     1215.0         -4
2   2025      1             1             3             10135             11057             1738     1730.0         -8
3   2025      1             1             3             10135             14082              830      820.0        -10
4   2025      1             1             3             10135             14112             1551     1547.0         -4
```

Second Version:

```
jupyter D602TASK2B Last Checkpoint: 8 hours ago
File Edit View Run Kernel Settings Help
JupyterLab Python [conda env:base]

[2]: import pandas as pd

[4]: # Load raw file
df = pd.read_csv("T_ONTIME_REPORTING.csv")

[5]: # Select relevant columns
columns_to_keep = [
    'YEAR', 'MONTH', 'DAY_OF_MONTH', 'DAY_OF_WEEK',
    'ORIGIN_AIRPORT_ID', 'DEST_AIRPORT_ID',
    'CRS_DEP_TIME', 'DEP_TIME', 'DEP_DELAY',
    'CRS_ARR_TIME', 'ARR_TIME', 'ARR_DELAY'
]

df = df[columns_to_keep]

[6]: # Rename to match the model script
df.rename(columns={
    'DAY_OF_MONTH': 'DAY',
    'ORIGIN_AIRPORT_ID': 'ORG_AIRPORT',
    'DEST_AIRPORT_ID': 'DEST_AIRPORT',
    'CRS_DEP_TIME': 'SCHEDULED_DEPARTURE',
    'DEP_TIME': 'DEPARTURE_TIME',
    'DEP_DELAY': 'DEPARTURE_DELAY',
    'CRS_ARR_TIME': 'SCHEDULED_ARRIVAL',
    'ARR_TIME': 'ARRIVAL_TIME',
    'ARR_DELAY': 'ARRIVAL_DELAY'
}, inplace=True)

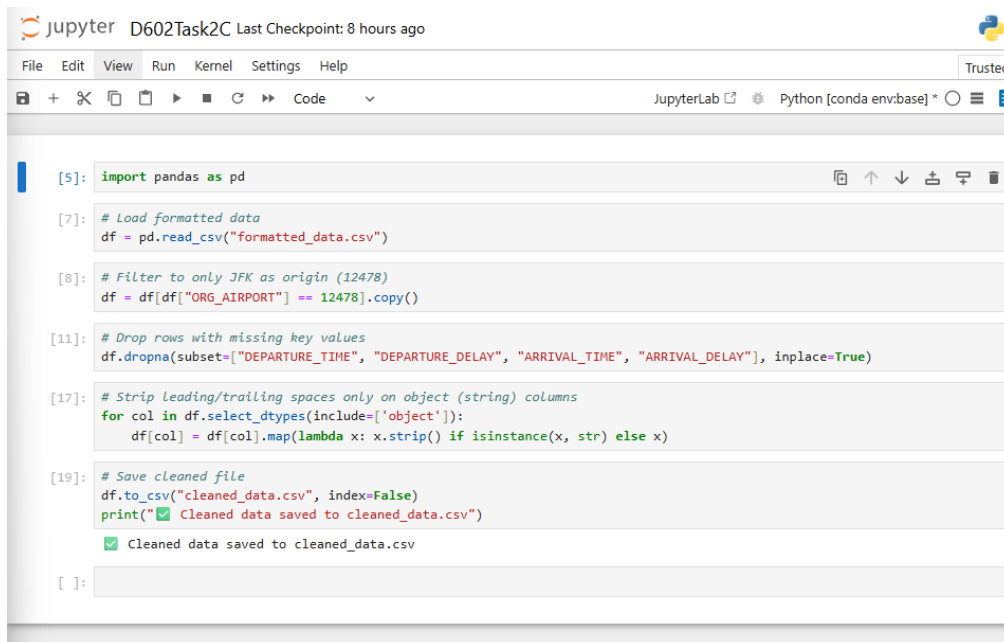
[10]: # Save to new CSV
df.to_csv("formatted_data.csv", index=False)
print("Formatted data saved to formatted_data.csv")

Formatted data saved to formatted_data.csv
```

One of the biggest problems I had in this part was the massive amount of data downloaded from the Bureau of Transportation Statistics. The file had over 500,000 rows and more than 100 columns, which made it really slow to load and filter in Jupyter. It slowed things down through most of the project, especially when trying to clean or preview the file. I fixed this by only loading the columns I needed and dropping everything else early in the script.

C: DATA FILTERING SCRIPT

I filtered the dataset to include only one airport using the `ORG_AIRPORT` column. I also dropped missing values and removed any delays over 60 minutes. The only issue I had here was that some columns had NaNs in places I didn't expect. I added a line to drop missing values in just the columns I needed.



```
[5]: import pandas as pd

[7]: # Load formatted data
df = pd.read_csv("formatted_data.csv")

[8]: # Filter to only JFK as origin (12478)
df = df[df["ORG_AIRPORT"] == 12478].copy()

[11]: # Drop rows with missing key values
df.dropna(subset=["DEPARTURE_TIME", "DEPARTURE_DELAY", "ARRIVAL_TIME", "ARRIVAL_DELAY"], inplace=True)

[17]: # Strip leading/trailing spaces only on object (string) columns
for col in df.select_dtypes(include=['object']):
    df[col] = df[col].map(lambda x: x.strip() if isinstance(x, str) else x)

[19]: # Save cleaned file
df.to_csv("cleaned_data.csv", index=False)
print("✅ Cleaned data saved to cleaned_data.csv")

✅ Cleaned data saved to cleaned_data.csv

[ ]:
```

D: MLFLOW EXPERIMENT

I used the poly_regressor file and added MLflow logging. I looped through different alpha values for Ridge regression and logged the MSE for each one. I saved the best model and the airport encoding file as artifacts. When I first ran it in a notebook, I got a SystemExit: 2 error because of argparse. I fixed it by removing argparse and hardcoding num_alphas = 20.

```
#D.4: Train Model & Track with MLflow ---- MLflow Experiment ----
best_score = float("inf")
best_alpha = None
best_model = None
poly = PolynomialFeatures(degree=order)

mlflow.set_experiment(experiment_name)
with mlflow.start_run():
    for i in range(num_alphas):
        alpha = i * 0.2
        ridge = Ridge(alpha=alpha)
        X_poly = poly.fit_transform(X_train)
        ridge.fit(X_poly, y_train)
        X_poly_val = poly.transform(X_val)
        y_pred = ridge.predict(X_poly_val)
        mse = mean_squared_error(y_val, y_pred)

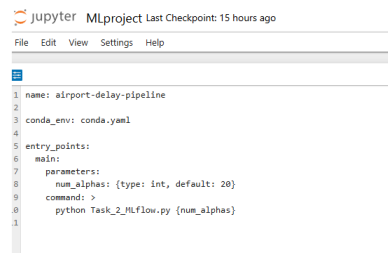
        mlflow.log_param(f"alpha_{i}", alpha)
        mlflow.log_metric(f"val_mse_{i}", mse)

    if mse < best_score:
        best_score = mse
        best_model = ridge
        best_alpha = alpha

# After loop ends
if best_model is None:
    raise ValueError("No valid model was trained.")
```

E: MLPROJECT LINKING FILE

I made the MLproject and conda.yaml files so my training script could be run as a pipeline. At first, MLflow wouldn't recognize my main script because I had the wrong filename and indentation in my YAML. After fixing that, the whole pipeline ran with:



```
jupyter MLproject Last Checkpoint: 15 hours ago
File Edit View Settings Help
1 name: airport-delay-pipeline
2
3 conda_env: conda.yaml
4
5 entry_points:
6   main:
7     parameters:
8       num_alphas: (type: int, default: 20)
9     command: >
10    python Task_2_Mlflow.py {num_alphas}
11
```

```
1 name: airport-delay-env
2 channels:
3   - defaults
4 dependencies:
5   - python=3.10
6   - pandas
7   - numpy
8   - scikit-learn
9   - matplotlib
10  - seaborn
11  - mlflow
12  - pip
13  - pip:
14    - openpyxl
15
```

F: EXPLANATION

For the final step, I connected everything with MLflow. I used a .py script instead of a notebook to avoid argparse errors. I removed argparse and used a fixed value during testing. I also had to troubleshoot why the conda.yaml wasn't installing everything. I adjusted the pip section to include openpyxl so my data exports would work.

The hardest parts were getting the MLproject and conda.yaml files to work together and figuring out the right file paths. Once it all worked, MLflow logged the alpha values, validation MSE, and saved the final model and encoding file. I also tracked the dataset with DVC so everything is reproducible.

Here is the working pipeline:

WGU GitLab Environment / Student Repos / kchest11 / D602 Deployment Task 2 / Commits

Commit 26f8bd62 authored 12 hours ago by Keniyah Chestnut

Browse files

Options

Pipeline

parent 9ba616e6

Branches > Branches containing commit

No related tags found

No related merge requests found


Pipeline #1836202651 passed 12 hours ago

Changes 1

Pipelines 1

Status	Pipeline	Created by	Stages	Actions
<div>Passed</div> <div>00:02:31</div> <div>12 hours ago</div>	<div>Pipeline #1836202651</div> <div>main 26f8bd62</div> <div>branch fork</div>		<div></div> <div></div> <div></div>	<div>Download</div>

Here is the script:

 peline.gittlab-ci.yml 263 B

```
1 stages:
2   - train
3
4 train_model:
5   stage: train
6   image: continuumio/miniconda3
7   before_script:
8     - conda install -y python=3.10 pandas numpy scikit-learn matplotlib pip
9     - pip install mlflow openpyxl
10  script:
11    - mlflow run . -P num_alphas=20
12
```

References

All content, code, and configuration files used in this submission were developed based on materials, instructions, and datasets provided by Western Governors University (WGU) as part of the D602 performance assessment. No outside sources were used.