DA5402 assgn 1 Report

1. What was the most difficult part of managing data versions manually?

The hardest part was tracking which dataset version was created by which script and at what stage. Even though I used version names like v1_raw, v2_cleaned, and stored them in separate folders, it became confusing after multiple runs.

Every time I modified preprocessing, I had to manually:

- Rename files
- Update config.yaml
- Ensure nothing was overwritten
- Update logs

2. How did you ensure that the model in production was the same one you evaluated during training?

I saved each model along with metadata that included:

- Dataset used
- Training date
- Git commit hash
- Accuracy
- Model parameters

3. How did you ensure that train.py always used the correct version of data from config.yaml?

I avoided hardcoding paths.

train.py always read file paths directly from config.yaml. So whenever I updated the version inside the config file, the training script automatically used that version.

4. Top three features I would want in an automated MLOps tool

1. Automatic data versioning – So every dataset change is tracked automatically without manual renaming.
2. Model registry with version control – To automatically link model, dataset, metrics, and code version.

5. Describe the breakdown you experienced while tracking which model was serving the API.

The breakdown happened during deployment.

I had multiple model files, and a deployment log. There was no automatic guarantee that the API was serving the exact model whose accuracy I reported.

6. How would an automated Model Registry have made Phase B easier?

An automated registry would have:

- Assigned version numbers automatically
- Linked model to data and code version
- Prevented accidental overwriting
- Allowed easy promotion to production
- Enabled rollback