QFlow: Solution for Managing ML Projects

Yuriy Gankin
Chief Scientific Officer

In recent years, the rapid growth of AI technologies has resulted in an increasing number of organizations adopting machine learning techniques to solve complex business problems. However, the process of building and deploying ML models at scale can be challenging, involving numerous complex steps such as data preparation, feature engineering, model training, and deployment.

Quantori introduces the QFlow framework that provides a structured approach to managing ML projects, enabling teams to collaborate more efficiently, improve code quality, and automate various tasks. This framework provides a structure to develop, maintain, and scale ML projects. QFlow emphasizes modularity, reproducibility, and versioning, enabling teams to track changes easily and reproduce results. Compared to other pipeline management solutions, QFlow offers a more flexible, scalable, and intuitive approach.

QFlow provides numerous benefits for life science projects:

The easy-to-use Python-based DSL allows for straightforward pipeline management without the need for extensive engineering expertise. This is particularly important for life science projects, where researchers may not have a strong background in software engineering.

QFlow enables the deployment of specialized tools often used to process life science data. This allows for greater flexibility and customization in the pipeline management process, essential for life science projects that often require unique and complex workflows.

As the volume of data grows, QFlow allows for easy scaling to meet the project's demands without significant infrastructure changes. This scalability also enables companies to more efficiently handle larger datasets and accelerate the research process.

Typical Workflow When Using QFlow

1 A typical workflow of using QFlow starts with creating a project and configuring it according to the specific needs of the ML pipeline. This can be done using a simple configuration wizard that guides the user through the process and helps set up the project's environment, dependencies, and resource allocation.

```
(qflow-py3.10) ➜  ml-pipelines git:(dev) ✗ qflow new --project-name example
Tracker to configure (mlflow, wandb, filetracker) [filetracker]: mlflow
SUCCESS: Project 'example' is created successfully.
Configuring the component: 'mlflow':
Name of the new configuration [mlflow]:
Address of local or remote tracking server [.qflow/tracker/mlflow]:
Address of local or remote model registry server [null]:
SUCCESS: Component mlflow configured
SUCCESS: Project 'example' is configured.
```

Fig. 1. Creating a New Project

2 Once the project is set up, the user can prototype different pipeline steps using Jupyter notebooks or scripts. As soon as they are ready, these notebooks/scripts can be converted into pipeline steps; the pipeline is written using QFlow's DSL-based approach, which allows for intuitive and readable pipeline definition. The pipeline definition and the steps are version-controlled using GitHub.

```
(qflow-py3.10) ➜  example git:(dev) ✗ qflow workflow run --workflow-name logistic_regression
                    Workflow
 Workflow                    60% 0:00:05

                     Steps
 0_factory_load             100% 0:00:00
 1_factory_load             100% 0:00:00
 2_factory_load             100% 0:00:00
 3_FitLogisticRegression      0% 0:00:05
 4_Predict                    0% -:--:--
```

Fig. 2. Creating Pipelines from CLI

3 With QFlow, it is also possible to deploy specialized tools and libraries needed for the data processing phase (e.g., it allows deploying a service for fMRI processing), which is often a critical part of life science projects. This is achieved by leveraging containerization and the support of plugins.

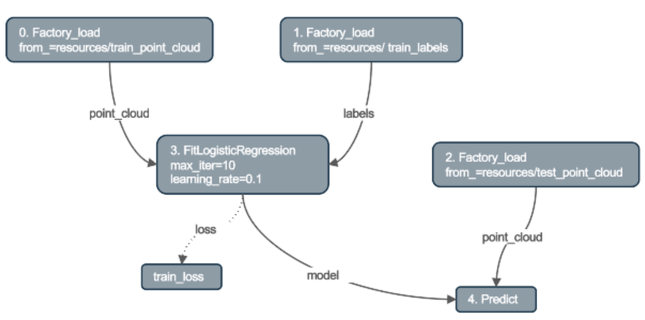

Fig 3. Example of a Pipeline Graph

4 As the project grows in size and complexity, the QFlow project can be configured to run the pipelines in the cloud to handle larger datasets and more complex models.

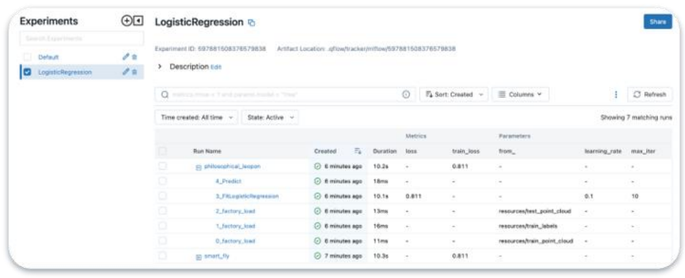

Fig 4. Tracked Pipeline

5 To expand QFlow's functionality, it uses pluggy, a lightweight plugin manager for Python. This feature allows developers to create custom plugins and add new functionality to QFlow as needed.
QFlow integrates with popular machine learning tracking solutions such as MLFlow and Weights & Biases for tracking purposes. Custom tracking solutions can also be added by creating plugins.
For infrastructure provisioning, QFlow uses Terraform. This enables developers to provision new resources and infrastructure quickly and easily in a scalable and reproducible.
Finally, QFlow provides automation for GitHub workflows, enabling users to run and test pipelines from pull requests.

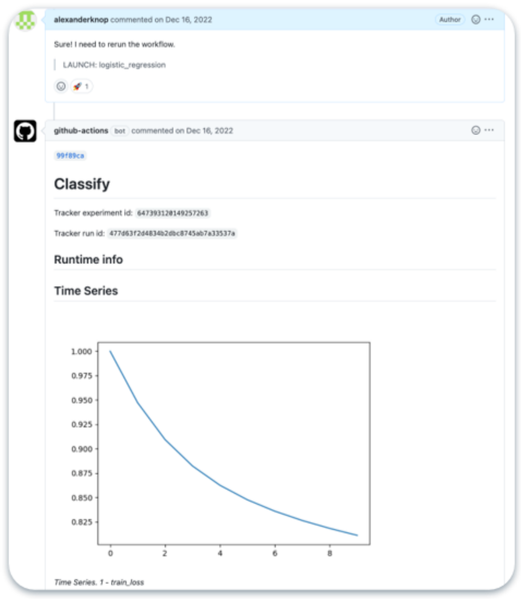

Fig 6. Pipelines can be Executed from PR

contact@quantori.com
+1(617) 225-4353