Submitting a Pull Request

Intro

After following the ML developer guide to iterate on ML code, the next step is to get your updated code merged back into the repo for production use. This page walks you through the workflow for doing so via a pull request.

Opening a pull request

Add project reference to Azure devops build definition

Prior to opening a pull request for a brand new project, ensure that the project has been added to the `get_project_to_build.ps1` file. This ensures that when a pull request is submitted, only the project under review is validated and deployed. To add the reference for a new project follow these steps:

1. Open the file within VS Code
2. Within the `$editedFiles` section, add a reference to your new project:

```
"<new_project_name/*>" {
        Write-Host "<new_project_name> changed"
        AppendQueueVariable "<new_project_name>"
    }
```

Once this has been added the build pipeline will be able to pick up and build only the project that has changed within the pull request.

Open the pull request

Before opening a pull request ensure your unit tests pass and the pipelines run end to end in the PPD workspace. This will reduce the amount of time spent on rework within the PR. as well as wasted time on your part since the Azure pipelines can take some time to run.

To push your updated ML code to production, open a pull request against the remote Git repo containing the current project.

PR Description Template

A PR template should auto-populate in the PR description. IF it does not please copy and paste the below template into your request:

`This PR adds <PROVIDE A BRIEF DESCRIPTION OF THE CODE BEING ADDED IN THIS PULL REQUEST>.`

`Before submitting this PR, please make sure:`

`- [ ] Your model training and inference workflows run end to end without errors in PPS workspace`

`- [ ] You have added unit tests for your feature engineering code (at a minimum!) and these pass within the Azure Pipeline`

`- [ ] Documentation is complete and up to date: <ADD LINK TO JIRA DOCUMENTATION>`

Viewing test status and debug logs

Opening a pull request will trigger Azure DevOps Pipeline that runs unit and integration tests for the model training pipeline on Databricks. A model will be built, logged and validated on a test dataset (specified in `project_name/resources/model-workflow-resource.yml` under task_key: `ModelValidation` by default).

You can view test status and debug logs from the pull request UI, and push new commits to your pull request branch to address any test failures.

The integration test runs the model training notebook in the staging workspace, training, validating, and registering a new model version in the model registry.

Where possible use a sample of the production data to reduce the run time of the integration tests.

The fitted model along with its metrics and params will also be logged to an MLflow run. To debug failed integration test runs, click into the Databricks job run URL printed in the test logs to review the failure reasons.

Merging your pull request to main

Once the tests pass on your pull request, get your pull request reviewed by the same DAta Scientist that peer reviewed the EDA and development phase. Once they have approved the Pull request then it can be merged to the main branch.

Deploying your project

Projects can only be deployed into the PRD data science workspace from the release branch of the repo.

To do this you will need to set up another Pull Request merging the Main branch into the Release branch.

Once this is set up ask the Head of Data Science (or their stand in) to review and approve this pull request.

Approvals and Automation flow diagram

The diagram below shows the steps required to have code approved and deployed within the ML Ops framework.