

Federated ML application at edge R4 Test Document

- [Running Unit tests](#)
- [Pipeline Examples](#)
 - [Introduction](#)
 - [Quick Start](#)

Running Unit tests

A script to run all the unittests has been provided in `./python/federatedml/test` folder.

Once FATE is installed, tests can be run using:

```
sh ./python/federatedml/test/run_test.sh
```

All the unit tests shall pass if FATE is installed properly.

Pipeline Examples

Introduction

We provide some example scripts of running FATE jobs with [FATE-Pipeline](#).

Please refer to the document linked above for details on FATE-Pipeline and FATE-Flow CLI v2. DSL version of provided Pipeline examples can be found [here](#).

Quick Start

Here is a general guide to quick start a FATE job.

1. (optional) create virtual env

```
python -m venv venv
source venv/bin/activate
pip install -U pip
```

2. install `fate_client`

```
# this step installs FATE-Pipeline, FATE-Flow CLI v2, and FATE-Flow SDK
pip install fate_client
pipeline init --help
```

3. configure server information

```
# configure by conf file
pipeline init -c pipeline/config.yaml
# alternatively, input real ip address and port info to initialize pipeline
# optionally, set log directory for Pipeline
pipeline init --ip 127.0.0.1 --port 9380 --log-directory ./logs
```

4. upload data with FATE-Pipeline

```
# upload demo data to FATE data storage, optionally provide path to where deployed examples/data locates

python demo/pipeline-upload.py --base /data/projects/fate
```

If upload job is invoked correctly, job id will be printed to terminal and an upload bar is shown. If FATE-Board is available, job progress can be monitored on Board as well.

```

UPLOADING:
|||||100.00%
2020-11-02 15:37:01.030 | INFO      | pipeline.utils.invoker.job_submitter:monitor_job_status:121 - Job id
is 2020110215370091210977
Job is still waiting, time elapse: 0:00:01
Running component upload_0, time elapse: 0:00:09
2020-11-02 15:37:13.410 | INFO      | pipeline.utils.invoker.job_submitter:monitor_job_status:129 - Job is
success!!! Job id is 2020110215370091210977

```

1. run a FATE-Pipeline fit job

```
python demo/pipeline-quick-demo.py
```

This quick demo shows how to build to a heterogeneous SecureBoost job. Progress of job execution will be printed as modules run. A message indicating final status ("success") will be printed when job finishes. The script queries final model information when model training completes.

```

2020-11-02 10:45:29.875 | INFO      | pipeline.utils.invoker.job_submitter:monitor_job_status:121 - Job id
is 2020110210452959882932
Job is still waiting, time elapse: 0:00:01
Running component reader_0, time elapse: 0:00:07
Running component dataio_0, time elapse: 0:00:10
Running component intersection_0, time elapse: 0:00:14
Running component hetero_secureboost_0, time elapse: 0:00:46
Running component evaluation_0, time elapse: 0:00:50
2020-11-02 10:46:21.889 | INFO      | pipeline.utils.invoker.job_submitter:monitor_job_status:129 - Job is
success!!! Job id is 2020110210452959882932
2020-11-02 10:46:21.890 | INFO      | pipeline.utils.invoker.job_submitter:monitor_job_status:130 - Total
time: 0:00:52

```

2. (another example) run FATE-Pipeline fit and predict jobs

```
python demo/pipeline-mini-demo.py
```

This script trains a heterogeneous logistic regression model and then runs prediction with the trained model.

```

2020-11-02 15:40:43.907 | INFO      | pipeline.utils.invoker.job_submitter:monitor_job_status:121 - Job id
is 2020110215404362914679
Job is still waiting, time elapse: 0:00:01
Running component reader_0, time elapse: 0:00:08
Running component dataio_0, time elapse: 0:00:10
Running component intersection_0, time elapse: 0:00:15
Running component hetero_lr_0, time elapse: 0:00:42
2020-11-02 15:41:27.622 | INFO      | pipeline.utils.invoker.job_submitter:monitor_job_status:129 - Job is
success!!! Job id is 2020110215404362914679
2020-11-02 15:41:27.622 | INFO      | pipeline.utils.invoker.job_submitter:monitor_job_status:130 - Total
time: 0:00:43

```

Once fit job completes, demo script will print coefficients and training information of model.

After having completed the fit job, script will invoke a predict job with the trained model. Note that Evaluation component is added to the prediction workflow. For more information on using FATE-Pipeline, please refer to this [guide](#).

```

2020-11-02 15:41:28.255 | INFO      | pipeline.utils.invoker.job_submitter:monitor_job_status:121 - Job id
is 2020110215412764443280
Job is still waiting, time elapse: 0:00:02
Running component reader_1, time elapse: 0:00:08
Running component dataio_0, time elapse: 0:00:11
Running component intersection_0, time elapse: 0:00:15
Running component hetero_lr_0, time elapse: 0:00:20
Running component evaluation_0, time elapse: 0:00:25
2020-11-02 15:41:54.605 | INFO      | pipeline.utils.invoker.job_submitter:monitor_job_status:129 - Job is
success!!! Job id is 2020110215412764443280
2020-11-02 15:41:54.605 | INFO      | pipeline.utils.invoker.job_submitter:monitor_job_status:130 - Total
time: 0:00:26

```