**Steps to follow to execute the pipeline**

**Note:**

* Please ask for latest public access url to access the platform.
* These url will also be used in repo-info.py file as API\_ENDPOINT.
* The Github repository should be public.

**Follow the below steps to run the end-to-end demo:**

* Login to your GitHub account.
* Create a new repository with name “regression\_wine\_algo”. It should be “public”

Graphical user interface, text, application, email

Description automatically generated

* Get the http url and clone the repository in your system



* Open the cloned repository folder inside your IDE(VS Code, Sublime):
  + Create a folder name “.github”, at the root of your repository, create a folder “workflows”.



* + Create a new file in “workflows” folder, name that file “ci-cd.yaml”
  + To create file, use below command



* + Copy the below content from wordpad document in the ci-cd.yaml file and save.



* + Note: Ensure that syntax and indentation is correct.
  + Create a file and name it “MLproject” at the root, specify entry point and configurable hyper parameters.
  + To create file, use below command



* + Copy the content and save file “MLproject”



* + Create a conda.yaml file.



* + Mention all the libraries required and copy the content



* + Create a python file name “repo-info.py” which will use to start the training pipeline.

touch repo-info.py

* + - API\_ ENDPOINT will be given by the system. (**USE ABOVE GIVEN URL AS API\_ENDPOINT**)
    - Add “/train” in new endpoint.
    - Change the values of key “url”,”branch\_name” and “algo\_type” in data section with your repository url, branch\_name.



* + Create a train.py file and copy the content from below file.
    - touch train.py



* + Download the dataset file from

<http://archive.ics.uci.edu/ml/machine-learning-databases/wine-quality/>

* + Download winequality-red.csv and save file with name “wine-quality.csv”
* At end you should a have structure like this in regression\_wine folder:
  + .git
  + .github
    - workflows
      * ci-cd.yaml
  + MLproject
  + Conda.yaml
  + train.py
  + Data files “wine-quality.csv”
  + repo-info.py
* Check the indentation and spacing in all the config files and python files.
* Now commit from the terminal, using



* Note: Either use ssh/personal access token for pushing.
* Pipeline will be triggered

**Check Actions**

* Go to your repository, click on the actions tab
* Move to the actions tabs and Check whether the pipeline is triggered

Graphical user interface, text, application

Description automatically generated

* Click on build, and check the details, there will be section “execute py script”, click on execute py script and check if process completed successfully.

Text

Description automatically generated with medium confidence

Graphical user interface, text

Description automatically generated

* Once the pipeline is completed successfully open the MLOps platform and click on procced button.
* Use the same url which added in “repo-info.py” without “/train”
* EG: <http://c71b-115-119-250-30.ngrok.io>

Graphical user interface, application, email, Teams

Description automatically generated

* When model is trained, it will redirect to deployment UI/ Model Tracking.

Graphical user interface, application, email

Description automatically generated

**To retrain the model with different hyper parameters**

* + Change parameter values in “MLproject” file, make a commit and push the code
  + Return to MLOps Platform Home Page, and click on proceed.
* On the model tracking page, all the model trained are listed with logged hyper-parameters and metrics.

**Model Deployment**

* + Choose the best model and click the deploy button.
  + Background Process:
    - Model will be fetched from Minio for deployment
    - Environment will be created with required libraries.
    - Pod will be deployed, and model will be ready to predict the value.

Graphical user interface, table

Description automatically generated

* The above process will take approx. 3 min to complete.
* Once above steps are completed, prediction page will be loaded.

**Test the Model**

* Upload the prediction .csv file, and click on predict button

Graphical user interface, application, website

Description automatically generated

* Predicted result will be displayed on dashboard.

Graphical user interface, application, website

Description automatically generated

**Monitor Deployment Metrics**

* Click on the Prometheus button, to monitor the metrics.

Chart, line chart

Description automatically generated

* Click on Home button to start new project/ retrain the model.

-------------------------------------------------------------------------------------------------------------------------------

**Use Case: To train algorithms: Regression, Classification, Clustering**

* To train any algorithm/model user need to add the training code and data files in the git repository along with ci-cd.yaml, Mlproject, conda.yaml and repo\_info.py.
* Train.py file should be in same hierarchy of config files. i.e: main directory of github repo
* Make sure below mentioned library and code is there in training python file:
  + Reference python train file, check the code blocks and add required mlflow components in your training code.



* + MlFlow should be imported
  + Add mlflow.start\_run section, training code should be added and in this section
  + Log the parameters which needs to be tracked
  + Log the model to track
* Configure MLproject file, according to hyper parameters and entry point for training code file
  + After setting hyper parameter commit the file, either through UI/CLI
  + On commit, pipeline will trigger and python API file will send post request for training.
  + Once done, move to MLOps UI, click on proceed button and once model is trained, it will redirect to deployment UI/ Model Tracking.
  + To retrain the model with different hyper parameters, change value in “MLproject” file and commit the code, and return to MLOps Platform and click on proceed.
* On the model tracking page, all the model trained are listed with logged hyper-parameters and metrics.
* Click the deploy button, to start the deployment process.
* Once above steps are completed, prediction page will be loaded.
* Upload the prediction .csv file, and click on predict button
* Predicted result will be displayed on dashboard.
* Click on the Prometheus button, to monitor the metrics.
* Click on Home button to start new project/ retrain the model.

**USE ABOVE DEMO AS REFERENCE FOR ALL CONFIG FILES AND TRAIN FILE, TO RUN ANY ALGORITHM**