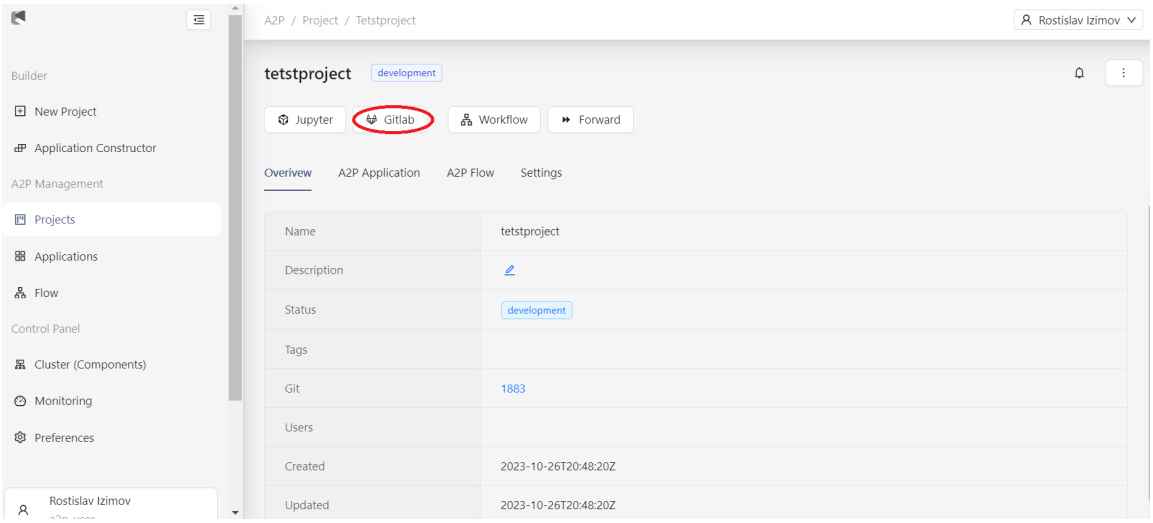
Работа с GitLab

Подключение к web-интерфейсу

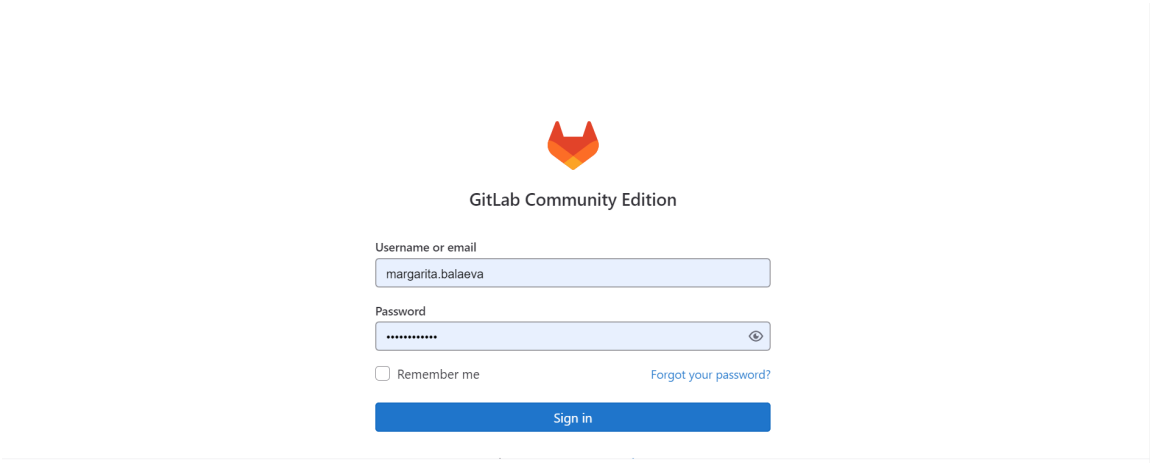
Для того, чтобы обеспечить эффективную совместную работу пользователей, а также для автоматизации CI/CD процессов используется система контроля версий Git (Gitlab SCM).

В системе контроля версий отслеживаются файлы блокнотов (ipynb), Python-файлы, конфигурационные файлы и файлы зависимостей (например, requirements.txt).

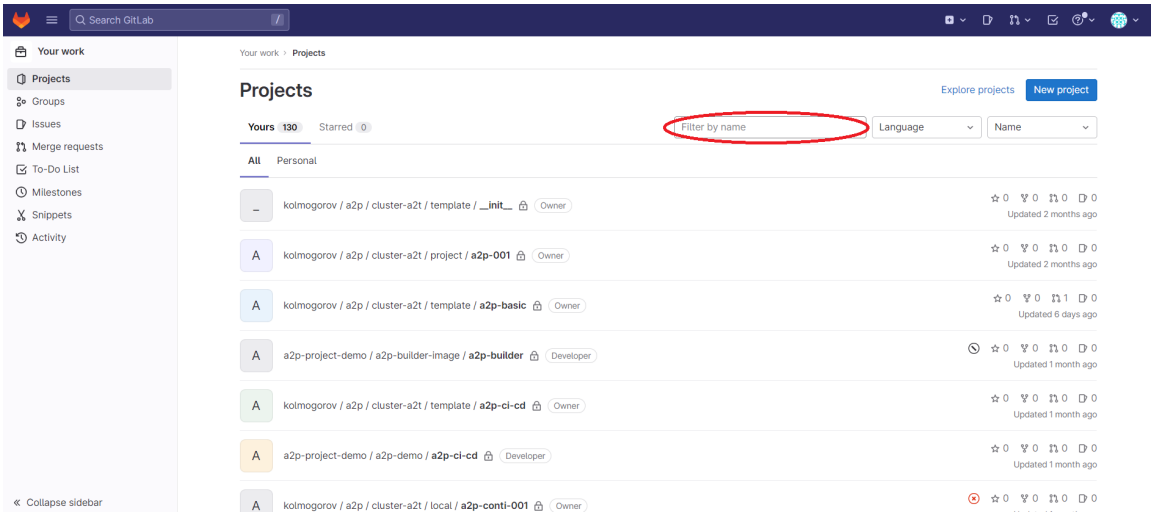
Для входа в web-интерфейс из модуля A2P требуется перейти по "GitLab" (см. рисунок ниже):



Далее ввести данные вашей учетной записи (рисунок ниже):



После входа в web-интерфейс будет отображена панель со всеми имеющимся репозиториями Пользователя, после этого нужно найти свой репозиторий своего проекта и перейти в него, если проектов много, то удобно воспользоваться окном поиска (см. рисунок ниже).



Обзор веток репозитория в GitLab

Следует напомнить, что репозиторий в GitLab, имеющий имя проекта разрабатываемой модели создается при запуске create_project.ipynb еще на этапе разработки модели в Jupyter.

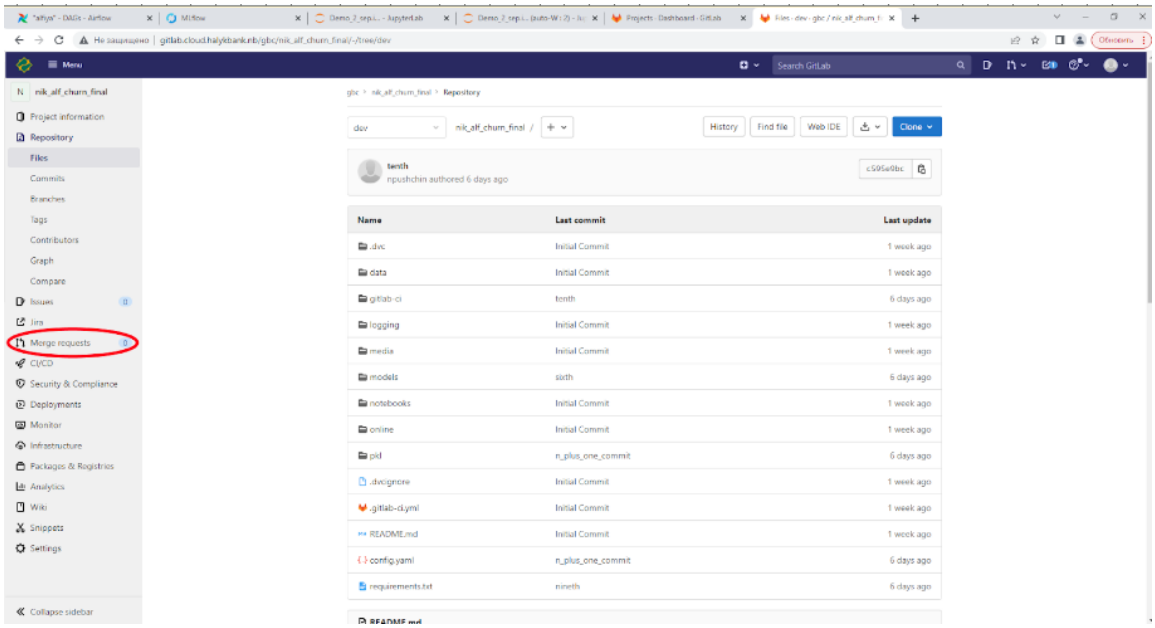
В созданном репозитории имеются необходимые конфигурационные файлы и следующие ветки:

- dev - ветка разработки модели;
- main - ветка, при успешном коммите в которую запускается CI/CD процесс в среде DEV;
- prod - ветка, при успешном коммите в которую запускается CI/CD процесс в промышленной среде (PROD) для Batch моделей;
- retrain - ветка, коммит в которую запускает процесс переобучения модели;
- online - ветка, при успешном коммите в которую, запускается CI/CD процесс в промышленной среде (PROD) для Online моделей.

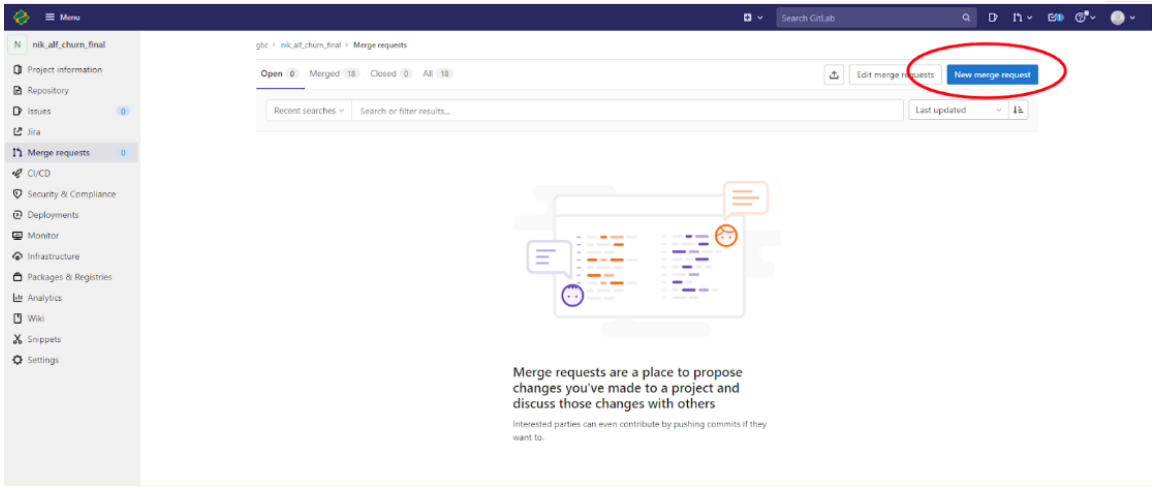
CI/CD процесс для продуктивизации модели в среде DEV

После того как Пользователь запушил коммит из контейнера разработки модели (контейнер, в котором велась работа в Jupyter), необходимо перейти в удаленный репозиторий в GitLab в ветку dev и проверить наличие данного коммита.

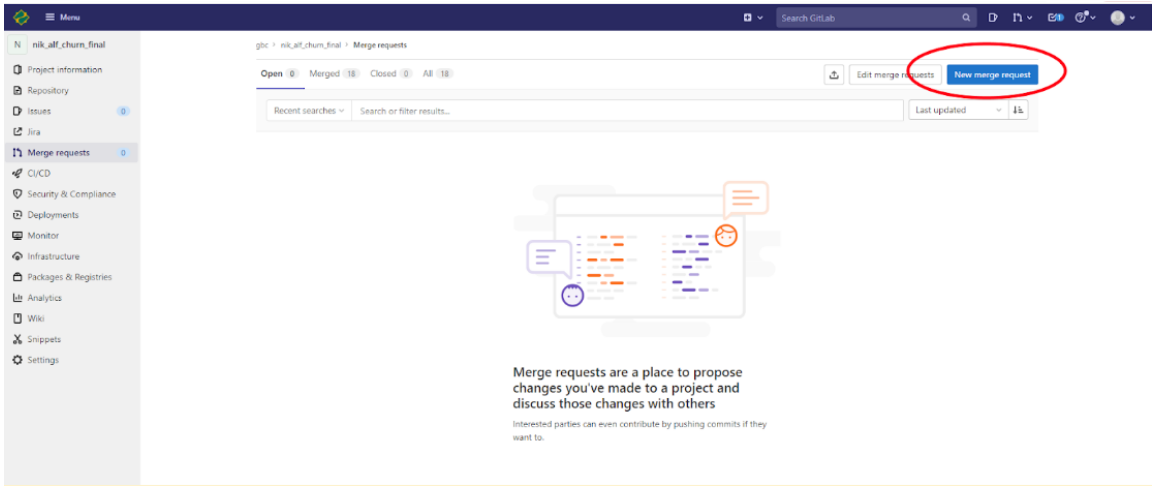
Далее, запуск автоматического CI/CD процесса продуктивизации модели в среде DEV инициируется с любого коммита в ветку main, для этого требуется нажать на кнопку «Merge Request» из панели пользователя, рисунок ниже.



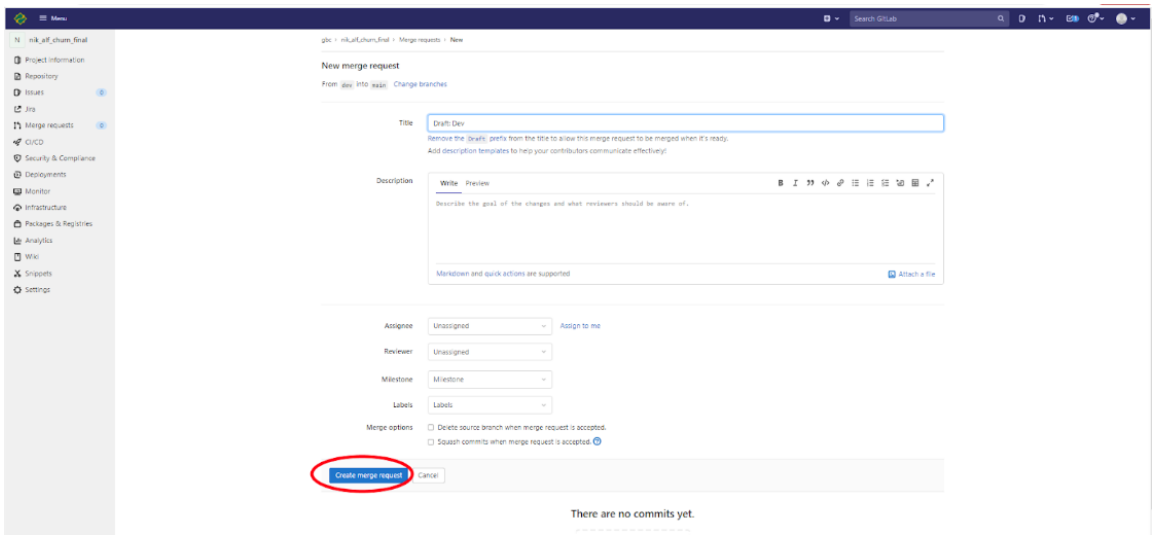
Далее нажать на “New merge request” (рисунок ниже):



Далее в поле Source branch необходимо выбрать dev, а в поле Target branch выбрать main и нажать “Compare branches and continue” (см. рис. ниже):

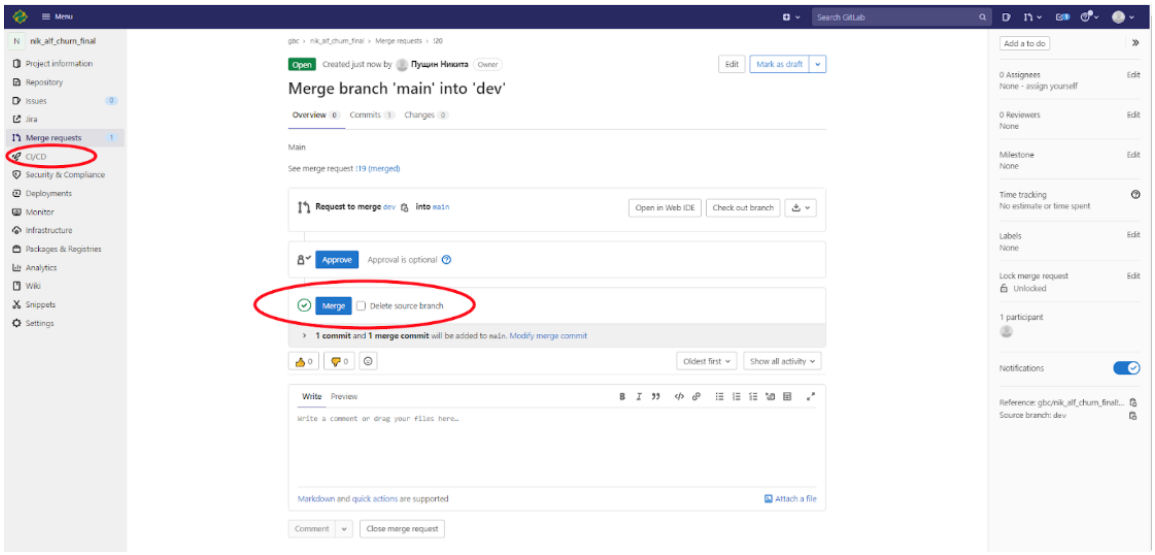


Далее, если необходимо, то заполнить поля Description, Assignee, Reviewer, Milestone, Labels и нажать “Create merge request” (рисунок ниже и описание Description, Assignee, Reviewer, Milestone, Labels указано ниже).

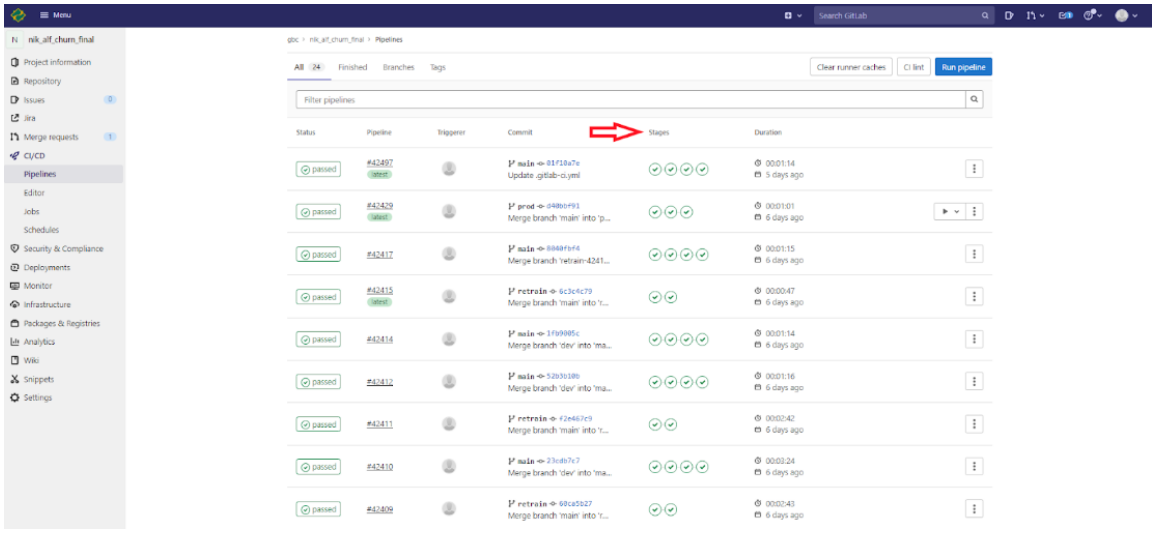


- Description - Описание Пользователем разработанной модели;
- Assignee - Ответственный за запрос;
- Reviewer - Рецензент запроса;
- Milestone - Временные рамки, специально установленные Пользователем, для конкретных целей;
- Labels - Указываются runner или группы runners, которые используются для различных задач.

На открывшейся странице нужно нажать на кнопку “Merge” и перейти на вкладку CI/CD слева (рисунок ниже):



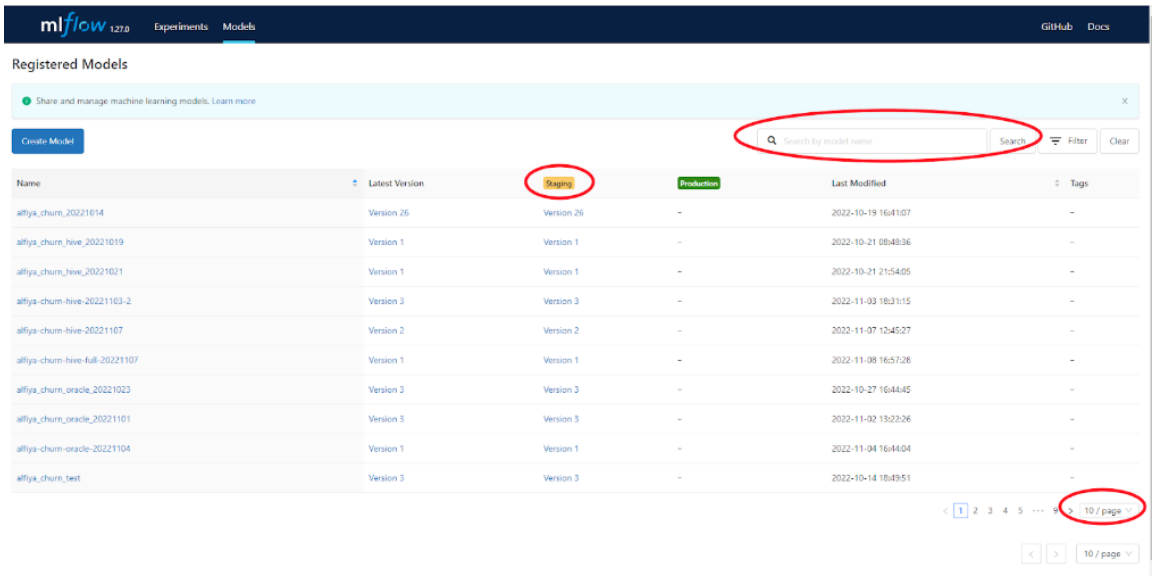
На рисунке ниже представлено страница с CI/CD пайплайнами для текущей модели.



Пайплайн продуктивизации модели в среде DEV состоит из 4 стадий (stages):

- build (сборка образа с моделью);
- create-model (создание модели в реестре моделей в MLflow);
- deploy (создание DAG-файла и загрузка в репозиторий airflow-dev);
- stage-model (переводит модель из стадии Latest в стадию Staging, см. рисунок ниже).

Таким образом, при успешном прохождение всех 4 стадий процесса можно перейти в MLflow и на вкладке Models увидеть последнюю версию и стадию своей модели. Если модель не найдена на экране, то можно воспользоваться поиском; еще полезно увеличить число отображаемых на одной странице моделей справа внизу, рисунок (MLFlow модель).

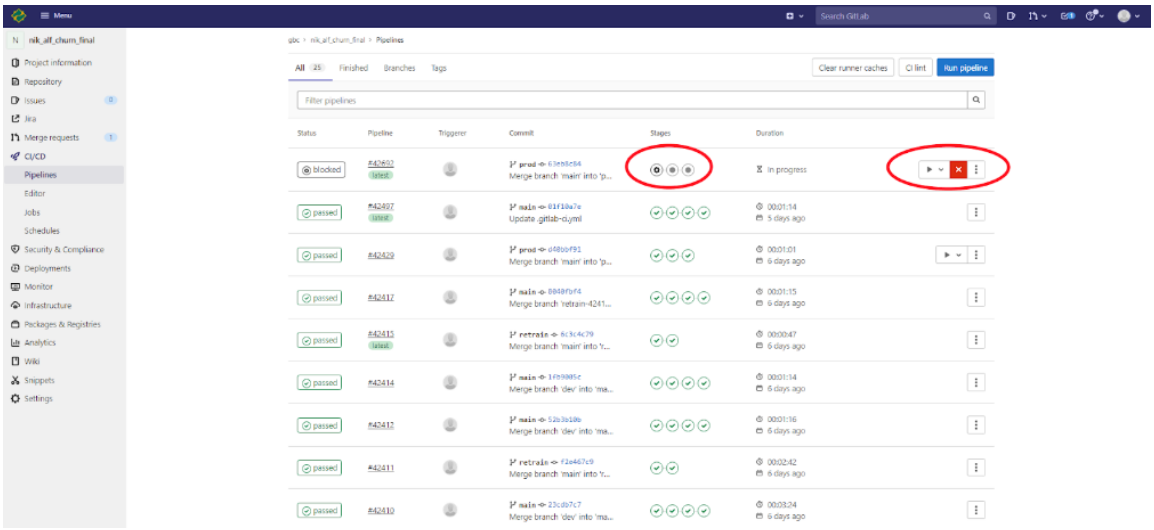


После окончания CI/CD процесса можно переходить в Airflow, где должен появиться DAG файл с названием проекта. Время появления DAG-файла в Airflow от окончания CI/CD процесса может занимать до нескольких минут. Далее можно активировать DAG-файл, более подробно про активацию DAG-файла в Airflow написано в разделе «Оркестрация рабочих процессов машинного обучения».

CI/CD процесс для продуктивизации модели в среде PROD

После того, как проверена работа модели в контуре DEV, она может быть переведена в контур PROD, для этого требуется выполнить Merge Request по аналогии с CI/CD процессом продуктивизации модели в контуре DEV, только в качестве Source Branch должна быть main, а в качестве Target Branch - prod. После успешного выполнения запроса слияния следует перейти в меню слева на вкладку CI/CD, на экране будет пайплайн, состоящий из 3 стадий. На первой стадии выполняется сборка docker образа с моделью, на второй стадии создается DAG-файл для модели и размещается в репозитории airflow-prod, на третьей стадии в реестре моделей MLflow производится перевод данной модели из стадии Staging на стадию Production.

При выводе в PROD каждая стадия процесса требует ручного запуска (значок шестеренка), для этого в колонке справа следует нажать на запуск (рисунок ниже).



После успешного окончания CI/CD процесса продуктивизации модели в контуре PROD можно переходить в Airflow-prod и выполнить установку DAG-файла на запуск по регламенту (расписанию), более подробно про работу в Airflow можно прочитать в разделе «Оркестрация рабочих процессов машинного обучения». Обратите внимание, что, в соответствии с ролевой моделью (см. Приложение А. «Ролевая Модель»), не все пользователи имеют возможность активации DAG-файлов модели в контуре PROD.

CI/CD процесс для переобучения моделей

В случае, если модель подразумевает переобучение, то на этапе разработки модели должен быть подготовлен файл retarin.py и dag-retrain.py. Логика процесса переобучения построена следующим образом: необходимо сделать коммит в ветку retrain, как правило, таким коммитом является merge request из ветки main в ветку retrain. Коммит автоматически запустит CI/CD процесс для переобучения модели, который состоит из двух стадий. На первой стадии формируется docker образ для применения кода переобучения (упрощенно, это контейнер, в котором будет запущен скрипт retrain.py). На второй стадии формируется DAG-файл для переобучения, имеющий имя проекта модели с приставкой “retrain” в конце, данный файл загружается в репозиторий airflow-dev. Далее, спустя обычно несколько минут, данный файл становится виден в интерфейсе Airflow, где он доступен для активации. Нужно его там активировать (более подробно про работу с Airflow смотрите в разделе «Оркестрация рабочих процессов машинного обучения»), после активации DAG-файл должен встать на расписание. Далее, по наступлению условия активации, данный DAG-файл должен отработать и, в случае отсутствия ошибок, в репозитории с моделью должна появиться новая ветка с названием типа retrain-pipeline_id-date, где data — это дата запуска DAG-файла переобучения, а pipeline_id — это номер CI/CD пайплайна. В этой ветке в папке pkl будет находиться свежеобученная модель в файле model.pkl.dvc. Далее нужно выполнить merge request из данной ветки retrain-pipeline_id-date в ветку main (для продуктивизации в среде dev) или в ветку prod (для продуктивизации в среде PROD). После слияния (“merge”) автоматически запустится

CI/CD продуктивизации модели для среды dev (подробнее см. п. «CI/CD процесс для продуктивизации модели в среде DEV») или prod (подробнее см. п. «CI/CD процесс для продуктивизации модели в среде PROD»).

Важно отметить, если переобучаемая модель стояла на расписании в Airflow и, например, проводила скоринг ежедневно, то после окончания CI/CD процесса, о котором шла речь выше (после мерджа retrain-pipeline_id-data в main или prod), будет автоматически будет обновлен DAG-файл с помощью, которого происходит скоринг моделью. То есть, на следующий день данная модель будет делать предсказания с помощью свежеобученной модели.

Таким образом, порядок действий Пользователя при переобучении, следующий:

1. Сделать merge request из ветки main в ветку retrain.
2. Перейти в Airflow и активировать DAG-файл переобучения.
3. В репозитории модели должна появиться новая ветка retrain-pipeline_id-date, где в папке pkl должен быть файл model.pkl.dvc, который и является свежеобученной моделью.
4. Выполнить merge request из ветки retrain-pipeline_id-date в ветку main (для переобучения модели, продуктивизированной в среде dev) или prod (для переобучения модели, продуктивизированной в среде prod).
5. После этого в Airflow соответствующей среды DAG-файл модели будет автоматически обновлен и при его следующем запуске, скоринг будет проводиться уже с помощью свежеобученной модели.