

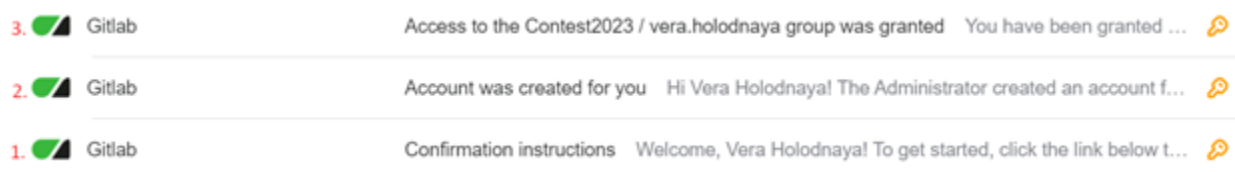
Вводная инструкция для участников

Вам понадобится

- Доступ к системе GS Labs Gitlab: <https://git-place.gs-labs.tv>.
- Базовые навыки работы с GIT (например, можно посмотреть здесь: <https://htmlacademy.ru/blog/git/git-basic-command>).
- Обучающая выборка (набор данных, предоставляемый участникам для выполнения задания конкурса) - ссылка на нее размещена в шаблоне конкурсного репозитория (в README).

Получение доступов в Gitlab

После того как мы добавим вас в систему, вам придет 3 автоматических письма от Gitlab. Необходимо открывать их по очереди, начиная с самого первого (нижнего).

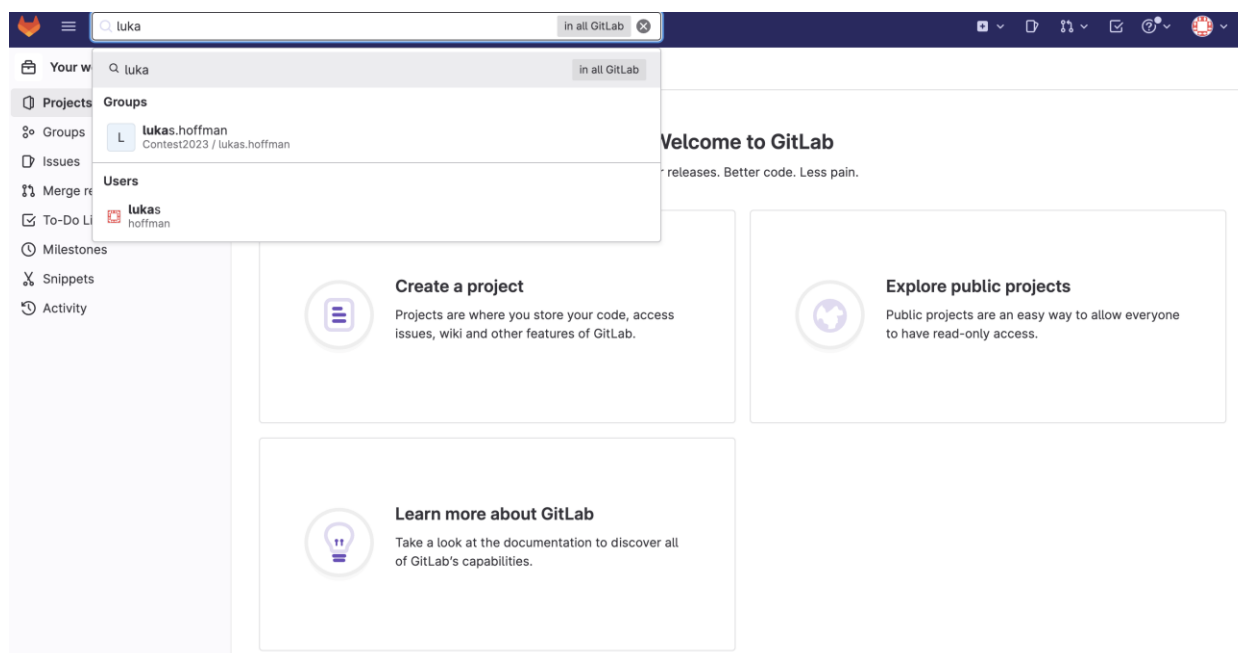


В первом письме «Confirmation instructions» необходимо подтвердить свою учетную запись, во втором «Account was created for you» – задать пароль. В третьем – ссылка в вашу директорию.

Важно! Не меняйте в личном кабинете Gitlab ваши данные (эл. почта, имя), они используются для идентификации вашей конкурсной работы.

Начало работы

Вы уже получили данные для доступа. Для примера здесь будет использоваться пользователь lukas.hoffman




1. Для работы вам необходимо клонировать шаблон конкурсного репозитория. Ссылка: https://git-place.gs-labs.tv/contest_public/contest2023_template. Сделать это можно командой: **git clone https://git-place.gs-labs.tv/contest_public/contest2023_template.git**

2. Создать новый проект в Gitlab в отведенной для вас директории (выглядит она следующим образом <https://git-place.gs-labs.tv/contest2023/name.surname> , где "name.surname" ваше имя в Gitlab GS Labs). Проект должен быть пустым, т.е. при создании необходимо снять галочку "Initialize repository with a README".

Важно! Проект в вашем окружении должен быть один!

Your work / Projects / New project / Create blank project



Create blank project

Create a blank project to store your files, plan your work, and collaborate on code, among other things.

Project name

Must start with a lowercase or uppercase letter, digit, emoji, or underscore. Can also contain dots, pluses, dashes, or spaces.

Project URL

Project slug

Visibility Level ⓘ

☒ Private
Project access must be granted explicitly to each user. If this project is part of a group, access is granted to members of the group.
Other visibility settings have been disabled by the administrator.

Project Configuration

☐ Initialize repository with a README
Allows you to immediately clone this project's repository. Skip this if you plan to push up an existing repository.

☐ Enable Static Application Security Testing (SAST)
Analyze your source code for known security vulnerabilities. [Learn more.](#)

Create project

Cancel

3. Далее для разработки вам необходимо сменить месторасположение клонированного ранее репозитория на вновь созданный репозиторий. При создании Gitlab вам выдаст подсказку как это сделать. Ниже на скриншоте три варианта на разные случаи. Если вы все делаете по инструкции, то ваш будет нижний.

Также необходимо переименовать папку contest2023_template, например: `mv contest2023_template repo1` (чтобы имя папки совпадало с названием созданного вами репозитория).

Почему ваши наработки надо складывать в правильное место? На группу contest2023 выданы права организаторам конкурса, чтобы они видели все работы в данной группе.

The repository for this project is empty

You can get started by cloning the repository or start adding files to it with one of the following options.

[Clone](#) [Upload File](#) [New file](#) [Add README](#) [Add LICENSE](#) [Add CHANGELOG](#) [Add CONTRIBUTING](#) [Add Wiki](#)

[Configure Integrations](#)

Command line instructions

You can also upload existing files from your computer using the instructions below.

Git global setup

```
git config --global user.name "Lukas"
git config --global user.email "Lukas.hoffman55@gmail.com"
```

Create a new repository

```
git clone https://git-place.gs-labs.tv/contest2023/lukas.hoffman/repo1.git
cd repo1
git switch -c master
touch README.md
git add README.md
git commit -m "add README"
git push -u origin master
```

Push an existing folder

```
cd existing_folder
git init --initial-branch=master
git remote add origin https://git-place.gs-labs.tv/contest2023/lukas.hoffman/repo1.git
git add .
git commit -m "Initial commit"
git push -u origin master
```

Push an existing Git repository

```
cd existing_repo
git remote rename origin old-origin
git remote add origin https://git-place.gs-labs.tv/contest2023/lukas.hoffman/repo1.git
git push -u origin --all
git push -u origin --tags
```

Важно! У вас есть клонированный репозиторий, вы поменяли в нем origin на правильный, выполнили push на сервер. Убедитесь в том, что репозиторий появился на сервере. Он должен отображаться в виде списка файлов.

R repo1

Project Information

Repository

Issues 0

Merge requests 0

CI/CD

Security and Compliance

Deployments

Packages and registries

Infrastructure

Monitor

Analytics

Wiki

Snippets

Settings

Contest2023 > lukas.hoffman > repo1

R repo1

Project ID: 73

1 Commit

1 Branch

0 Tags

0 Bytes Project Storage

Initial commit

Dmitry Stolyarov authored 1 minute ago

e8f33edd

master repo1

Find file Web IDE Clone

README

CI/CD configuration

Add LICENSE

Add CHANGELOG

Add CONTRIBUTING

Auto DevOps enabled

Add Kubernetes cluster

Add Wiki

Configure Integrations

Name	Last commit	Last update
gitlab-ci.yml	Initial commit	1 minute ago
Dockerfile	Initial commit	1 minute ago
README.md	Initial commit	1 minute ago
docker-compose.yml	Initial commit	1 minute ago
main.py	Initial commit	1 minute ago
requirements.pip	Initial commit	1 minute ago

Работа с репозиторием

1. Состав шаблона репозитория:

.gitlab-ci.yml - Файл настройки Gitlab CI, его нельзя редактировать. С помощью CI выполняется автоматическая проверка результатов выполнения вашей разработки.

output/result.csv - результат работы модели со списком предсказаний.

train - автоматически монтируемая папка с обучающими данными.

Dockerfile - Сборка и запуск ваших наработок. Редактировать можно с сохранением путей.

docker-compose.yml - не используется в CI, можно использовать на своем ПК для отладки.

main.py и **requirements.pip** - ваши файлы с наработками.

README.md – описание + ссылка на конкурсный датасет (Обучающую выборку).

2. Порядок работы

Разработка и отладка ПО на локальном ПК -> Коммит изменений в ветку для разработки (dev, stage, test и т.д) или master. Если разработка проходила не в ветке master, необходимо сделать Merge в ветку master.

Зачем нужен Merge? Запуск CI/CD настроен на тэг, его необходимо делать из ветки master, подразумевается, что в данной ветке находится работоспособное приложение. При успешном выполнении CI/CD вы сможете посмотреть результат выполнения вашего ПО, он будет находиться в артефактах, выполненных JOB, либо в меню Artifacts.

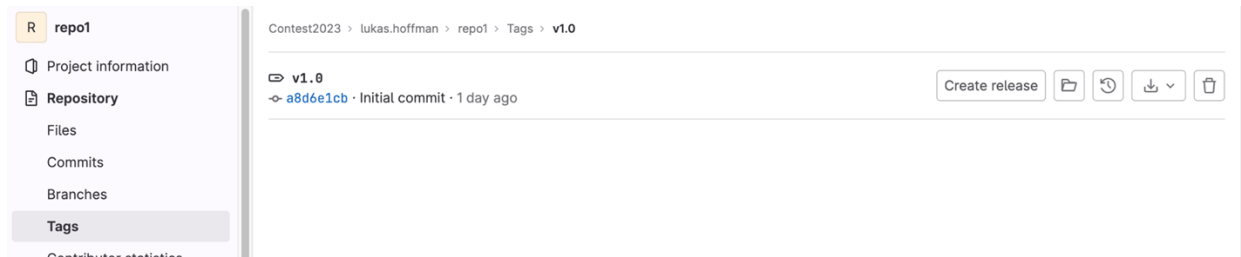
! На этапе "Разработка" показатель в артефактах неинформативный. Пришлем дополнительную информацию, если будут изменения (на рассмотрении варианты реализации проверки MAP по обучающей выборке на этапе "Разработка").

Status	Job	Pipeline	Stage	Name	Duration	Coverage
passed	#310 master → a8d6e1cb python	#122 created by	appraiser	Load_Data	00:00:05 1 day ago	Download artifacts
passed	#309 master → a8d6e1cb python	#122 created by	build	run	00:00:05 1 day ago	
passed	#308 master → a8d6e1cb python	#122 created by	build	inside	00:07:05 1 day ago	
manual	#307 master → a8d6e1cb python allowed to fail manual	#122 created by	NvidiaT...	TestNvidia		

3. Передача на проверку.

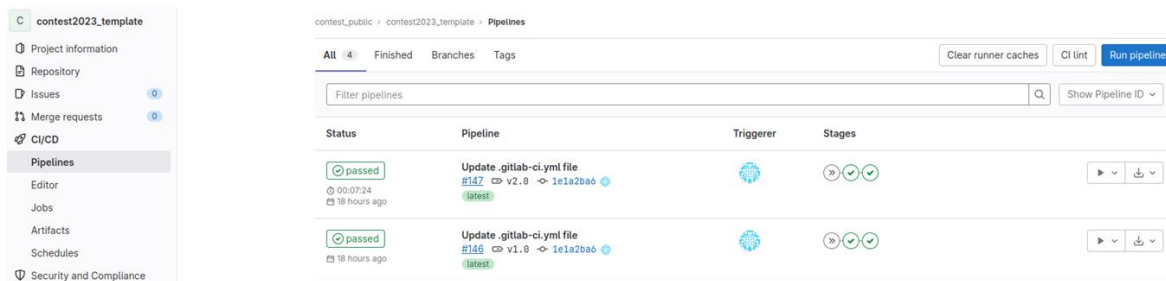
После оповещения Организатора в ветке master сделать tag (Например: git tag v1.0).

Важно! Если вы используете графическую карту, то в названии тэга обязательно должно присутствовать "GPU" (в начале тэга и большими буквами), например, GPU.v01.



При создании тэга у вас запустится Pipeline.

Важно! Необходимо убедиться в его успешном завершении. Если этого не произошло и перезапуск не помог, то необходимо написать организатору на contest@gs-labs.ru, приложив ссылку на Pipeline.



Рекомендации: Тэгом помечаются только работоспособные версии, проверенные локально. На проверку работ обычно уходит значительное время и желательно не откладывать публикацию работ на последние дни.

Дополнительно

Приложение можно разрабатывать с использованием графической карты, для этого ничего дополнительно делать не нужно. CI/CD по умолчанию запускает все приложения с возможностью использования графической карты.

При необходимости можно проверить работоспособность графической карты, запустив Job TestNvidia, вывод консоли должен быть примерно как на изображении ниже.

! Если будет большое количество работ на GPU (и соответственно большая нагрузка на систему), возможна корректировка в данной части.

passed Job TestNvidia triggered just now by lukas

Search job log

```

1 Running with gitlab-runner 16.0.2 (85586bd1)
2 on ml-runner-server KjbxmSuS, system ID: s_9c3a868eb4de
3 Preparing the "shell" executor 00:00
4 Using Shell (bash) executor...
5 Preparing environment 00:00
6 Running on ml-runner-server...
7 Getting source from Git repository 00:01
8 Fetching changes with git depth set to 20...
9 Переинициализирован существующий репозиторий Git в /home/gitlab-runner/builds/KjbxmSuS/0/contest2023/lukas.hoffman/rep
  o1/.git/
10 Checking out a8d6e1cb as detached HEAD (ref is v1.0)...
11 Удаление output/
12 Skipping Git submodules setup
13 Executing "step_script" stage of the job script 00:01
14 $ echo -n $CI_REGISTRY_PASSWORD | docker login -u $CI_REGISTRY_USER --password-stdin $CI_REGISTRY
15 WARNING! Your password will be stored unencrypted in /home/gitlab-runner/.docker/config.json.
16 Configure a credential helper to remove this warning. See
17 https://docs.docker.com/engine/reference/commandline/login/#credentials-store
18 Login Succeeded
19 $ docker run --rm --runtime=nvidia --gpus all nvidia/cuda:11.6.2-base-ubuntu20.04 nvidia-smi
20 Thu Jul 13 11:46:01 2023
21 +-----+
22 | NVIDIA-SMI 530.41.03                Driver Version: 530.41.03    CUDA Version: 12.1      |
23 |-----+-----+-----+
24 | GPU   Name                               Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
25 | Fan  Temp  Perf            Pwr:Usage/Cap|  Memory-Usage | GPU-Util  Compute M. |
26 |                                           | MIG M.       |
27 |=====+=====+=====+
28 |  0  NVIDIA GeForce RTX 3070              Off| 00000000:01:00:00 Off |         N/A |
29 |  0%   45C   P8              18W / 220W |  66MiB /  8192MiB |      0%   Default |
30 |                                           |               |
31 |-----+-----+-----+
32 |                                     |               |
33 |                                     |               |
34 +-----+-----+-----+
35

```

Описание метрики качества модели (MAP)

Для оценки качества моделей используется метрика MAP. С общей информацией можно ознакомиться по ссылке: <https://sdsawtelle.github.io/blog/output/mean-average-precision-MAP-for-recommender-systems.html>

К особенностям данной метрики относится то, что порядок предсказанных фильмов в предсказании играет роль в подсчете и первый фильм для предсказания имеет вес больший чем второй, и так далее.

При расчёте метрики на **тестовой** выборке следует исключать фильмы, которые смотрел пользователь в обучающей выборке (при включении фильмов, используемых для обучения модели, в список рекомендаций они будут считаться как промах рекомендации). При запуске функции расчёта метрики MAP на **обучающей** выборке ранее просмотренные фильмы могут попадать в рекомендации.

Также для расчёта метрики на **тестовой** выборке выбираются только те пользователи, данные о которых есть как в обучающей, так и в тестовой выборке.

Кратко расчёт метрики можно описать следующим образом :

По каждому фильму из списка предсказаний для пользователя рассчитаем величину $W = P/N * R$

P - количество угаданных фильмов(начинается с 1 и возрастает при каждом следующем найденном фильме)

N - номер фильма в списке предсказаний

R - признак нахождения события с просмотром данного фильма в тестовой выборке у выбранного пользователя (1 или 0)

далее складываем W всех фильмов и делим на размер списка предсказаний.

Полученные на предыдущем шаге значения усредняются по всем пользователям, т.е. вычисляется сумма значений для всех пользователей и делится на число пользователей в тестовой выборке.

Пример:

Допустим мы делаем предсказание для 3 пользователей и размер списка предсказаний равен 4.

Пользователь 1 - фильмы в позиций 1 и 3 из списка рекомендаций были просмотрены в тестовом периоде

Пользователь 2 - фильмы в позиции 2 из списка рекомендаций был просмотрены в тестовом периоде

Пользователь 3 - ни один фильм из списка рекомендаций не был просмотрен в тестовом периоде (но были просмотры других фильмов)

Тогда:

Пользователь 1: $W1 = 1/1 * 1$, $W2 = 1/2 * 0$, $W3 = 2/3 * 1$, $W4 = 2/4 * 0$. $W_{sum} = 1 + 0 + 2/3 + 0 = 5/3$

Пользователь 2: $W1 = 0/1 * 0$, $W2 = 1/2 * 1$, $W3 = 1/3 * 0$, $W4 = 1/4 * 0$. $W_{sum} = 0 + 1/2 + 0 + 0 = 1/2$

Пользователь 3: $W1 = 0/1 * 0$, $W2 = 0/2 * 0$, $W3 = 0/3 * 0$, $W4 = 1/4 * 0$. $W_{sum} = 0 + 0 + 0 + 0 = 0$

далее

Пользователь 1: $AP = W_{sum}/4 = (5/3)/4 = 5/12$

Пользователь 2: $AP = W_{sum}/4 = (1/2)/4 = 1/8$

Пользователь 3: $AP = W_{sum}/4 = 0/4 = 0$

Ну и в заключении складываем и делим на число пользователей 3:

$MAP = AP_{sum} / 3 = (5/12 + 1/8 + 0)/3$ приблизительно равно 0.18

Описание данных обучающей выборки

Обучающая выборка содержит информацию о просмотрах фильмов около 200 000+ пользователей за 70 дней. Участникам потребуется предсказать ТОП-20 наиболее релевантных фильмов каждому из пользователей, просмотренных им в следующие 30 дней.

movies.csv - файл с данными фильмов

id - id фильма

name - название

year - год выпуска

date_publication - дата публикации

genres - список id жанров

countries - список id стран производства

description - описание фильма

staff - список id staff

genres.csv - файл с данными жанров

id - id жанра

name - имя жанра

countries.csv - файл с данными стран

id - id страны

name - имя страны

staff.csv - файл с данными сочетания персона и должность

id - id записи

name - имя персоны

role - актер/режиссёр/ и т.д

logs.csv - файл с данными логов событий о просмотрах фильмов

id - id записи

user_id - id пользователя

datetime - время события (окончание просмотра фильма)

duration - длительность просмотра фильма

movie_id - id фильма