

Введение в современный MLOps

Владислав Гончаренко

girafe.ai



Владислав Гончаренко

- Автор курсов по машинному обучению и магистерской программы в МФТИ
- Исследователь МЛ (аспирант Физтеха)
- Руководитель команды ранжирования видео в Дзене
- ex-руководитель команды восприятия в беспилотных грузовиках
- фанат open source



Что такое MLOps?

girafe
ai

01

Зачем нужен MLOps?

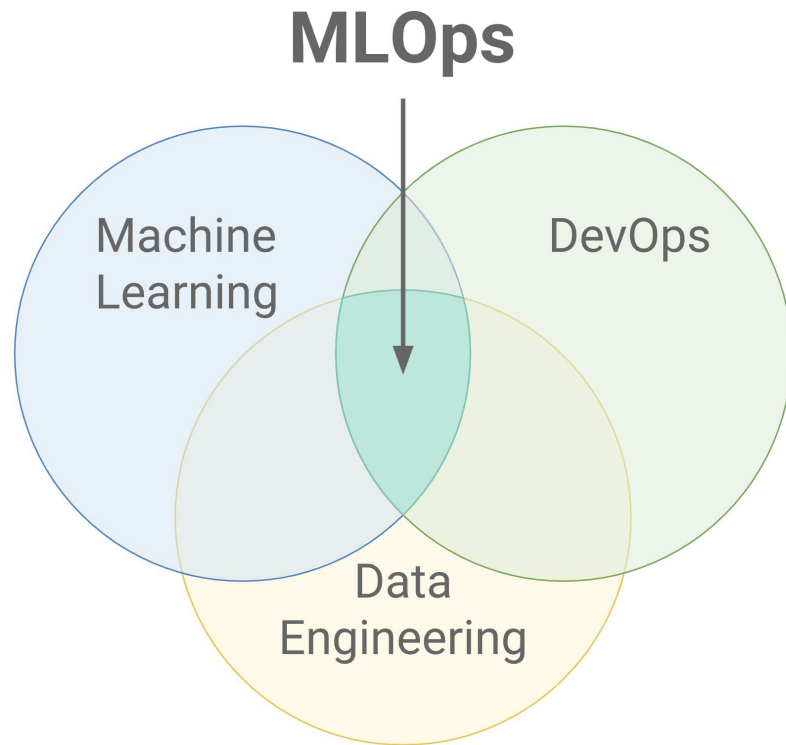
- Ресурсы, затрачиваемые на разработку моделей, всё растут
 - майнинг данных (cpu)
 - разметка данных (люди)
 - обучение моделей (gpu)
- Воспроизводимость тренировок
 - не только в индустрии, но и в исследованиях
- Организация доставки
 - сократить time-to-market
 - исключить рутину
- Декомпозиция компетенций
 - более глубокое разделение труда

Определение

MLOps is a paradigm that aims to deploy and maintain machine learning models in production reliably and efficiently.

MLOps seeks to increase automation and improve the quality of production models, while also focusing on business and regulatory requirements

[Well-known common knowledge site](#)



Зачем MLOps МЛщикам?

потому что никто другой для вас её не построит!!!

Для приобщения к теме проектирования систем рекомендую [system design primer](#)

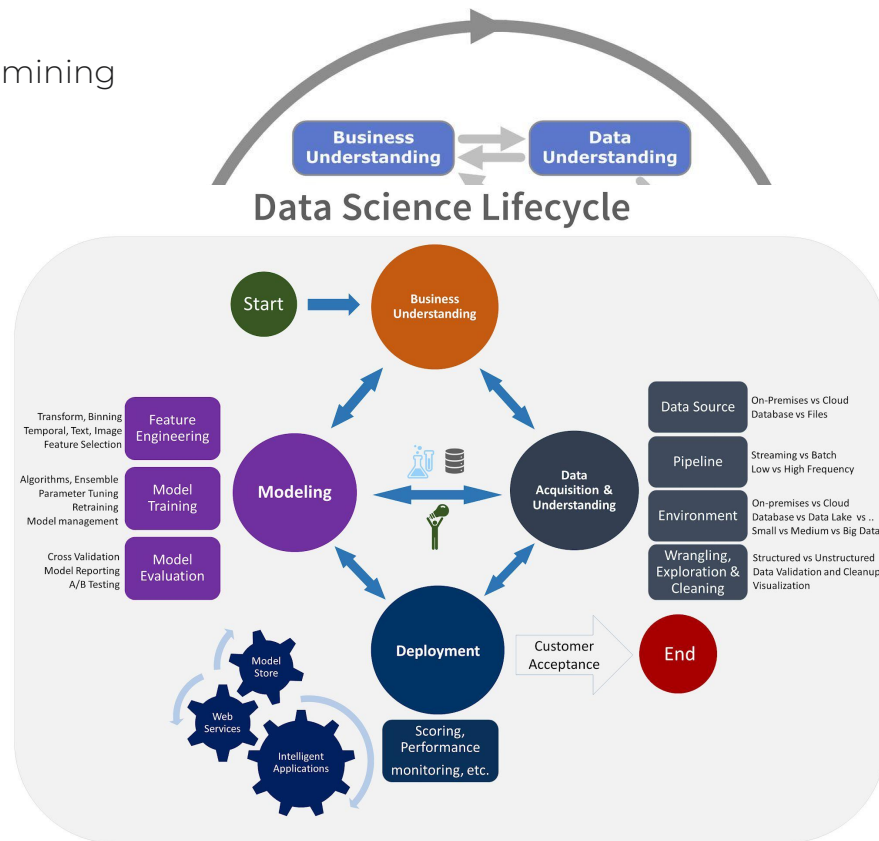
Стандарты разработки МЛ проектов

- **CRISP DM**

- Cross-industry standard process for data mining
- proposed in 1999
- upgraded to ASUM-DM in 2015

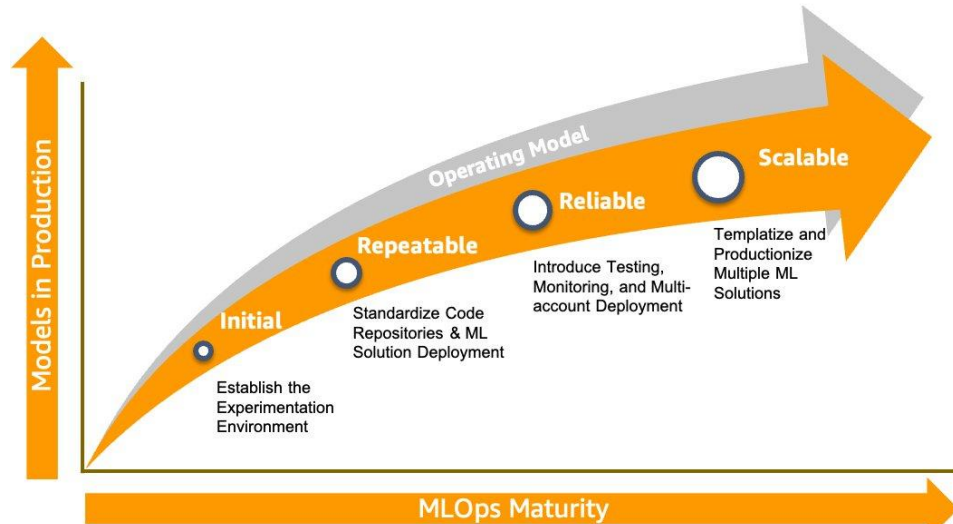
- **TDSP**

- Team standard Data Science Process by Microsoft



MLOps в крупных компаниях

- Amazon <https://aws.amazon.com/sagemaker/mlops/>
- Google <https://cloud.google.com/architecture/mlops-continuous-delivery-and-automation-pipelines-in-machine-learning>
- Nvidia <https://blogs.nvidia.com/blog/2020/09/03/what-is-mlops/>
- Яндекс: YTSaurus + Nirvana + Toloka




Даже Гуччи!!!

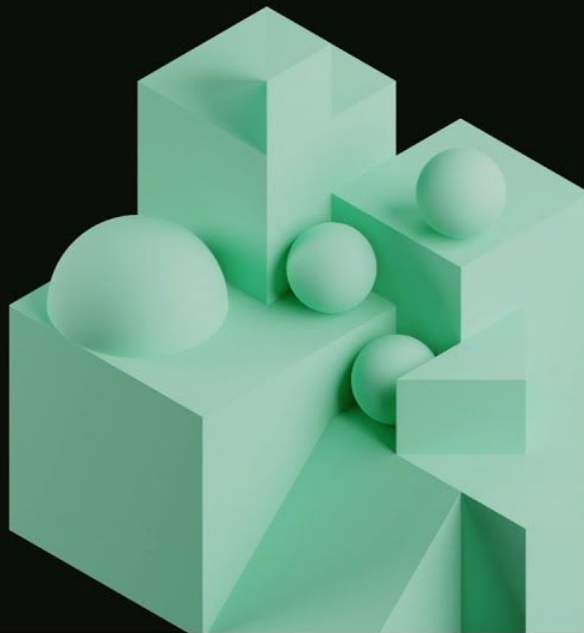
[Youtube video](#)

**DATA+AI
SUMMIT**
BY  databricks

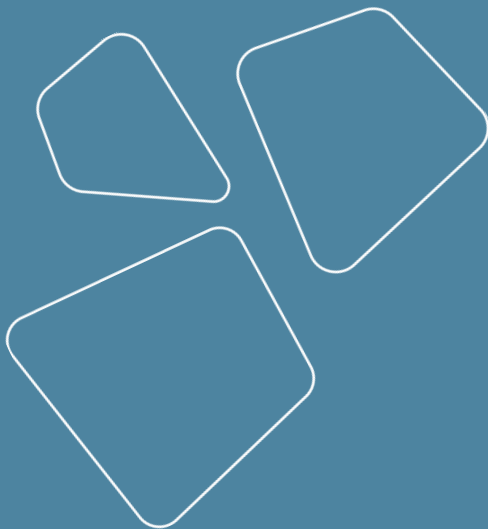
MLOps at Gucci: from zero to hero

An overview on implementing an MLOps
solution from scratch


Databricks
2023



Темы на сегодня



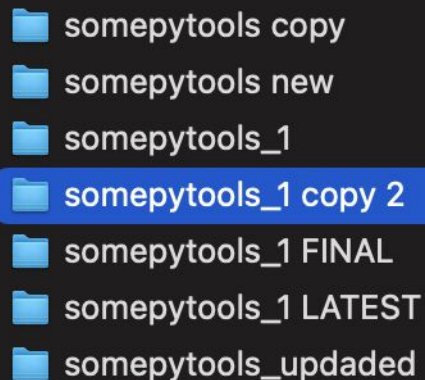
- Хранение кода
- Хранение данных
- Модель вычислений
- Логирование + визуализация
- Регулярные запуски кода

https://t.me/girafe_ai

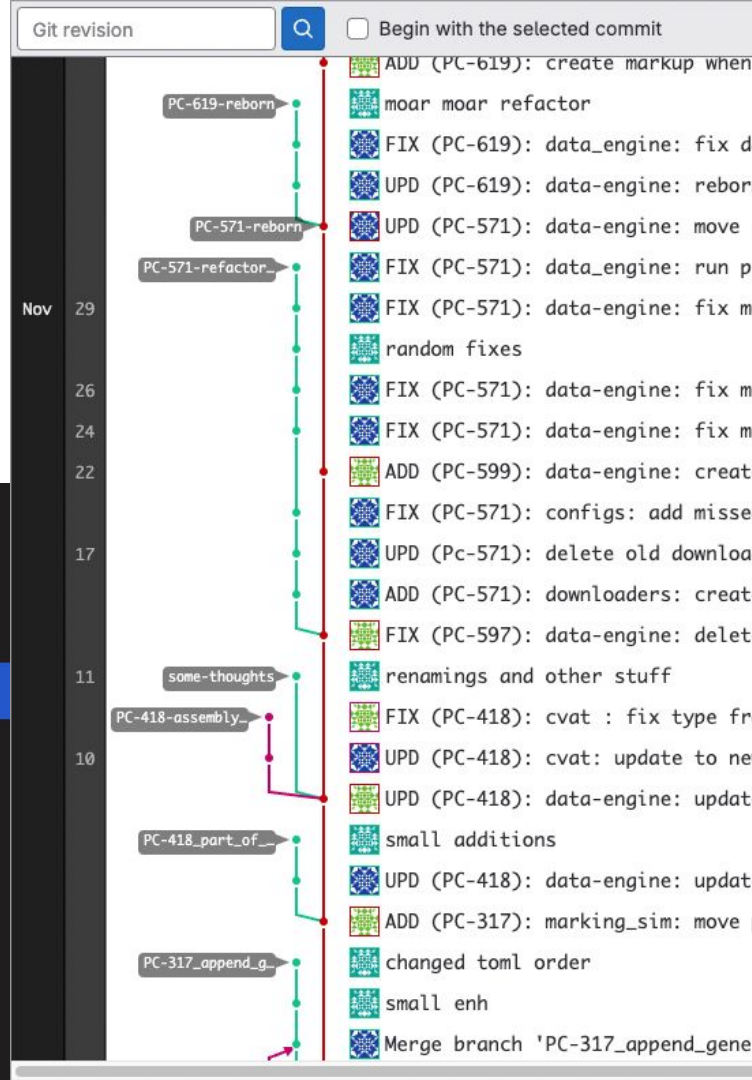


Хранение кода

- Одна локальная копия
- Много локальных копий
- Удалённые копии
- Система контроля версий (git, svn, etc.)

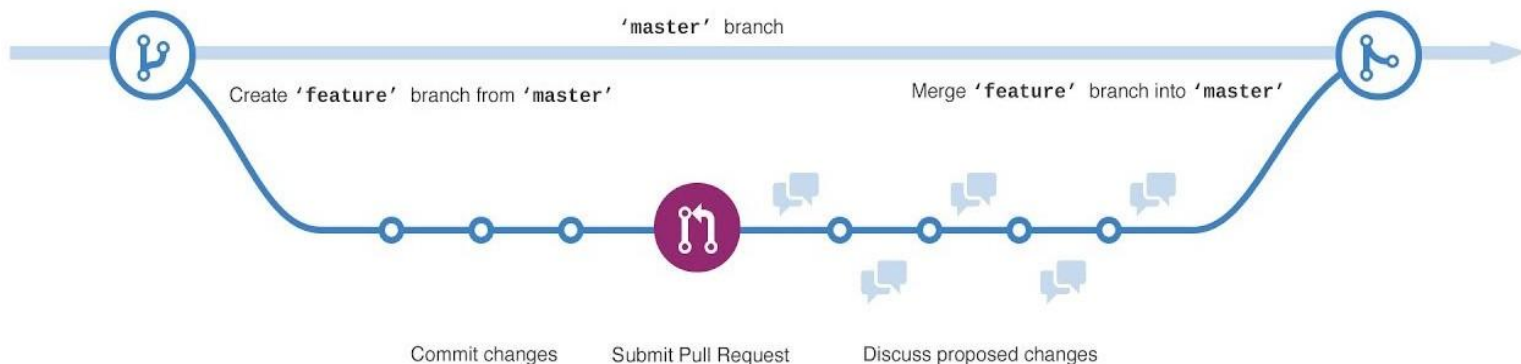


somepytools copy
somepytools new
somepytools_1
somepytools_1 copy 2
somepytools_1 FINAL
somepytools_1 LATEST
somepytools_updated



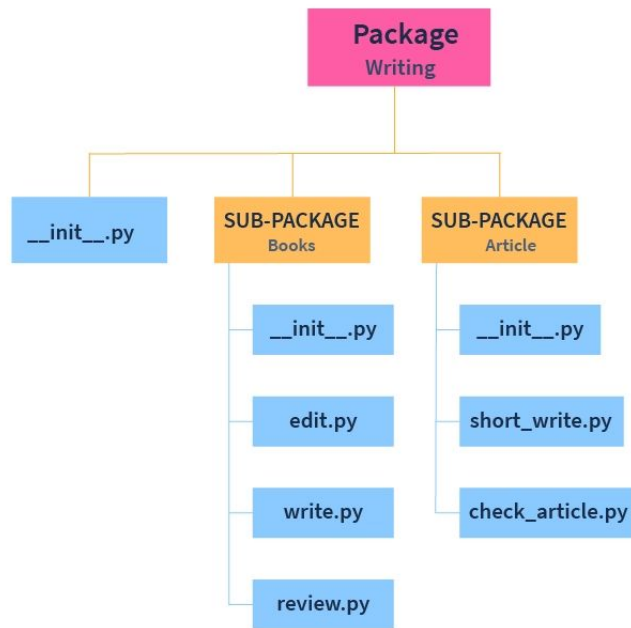
Version Control System (VCS)

- Source code - [git](#)
- Cloud remote - gitlab or github
- Best practice - [merge requests \(pull requests\)](#)



Дистрибуция кода

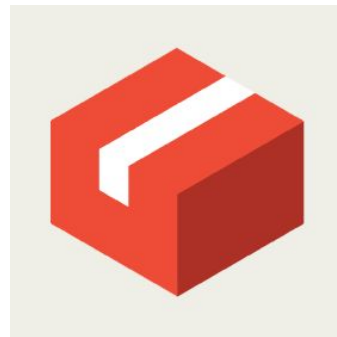
- Source code
 - клонируем проект на финальную машину
 - обновляем с помощью git pull
 - не требует структуры
- Пакеты
 - устанавливаем с помощью pip install
 - обновляем через pip update
 - предполагает структуру
 - для простого пакетирования подходит [poetry](#)



Хранение данных

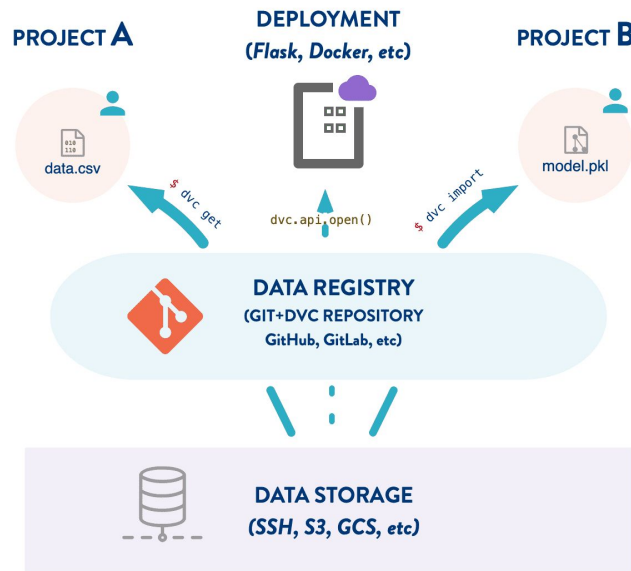
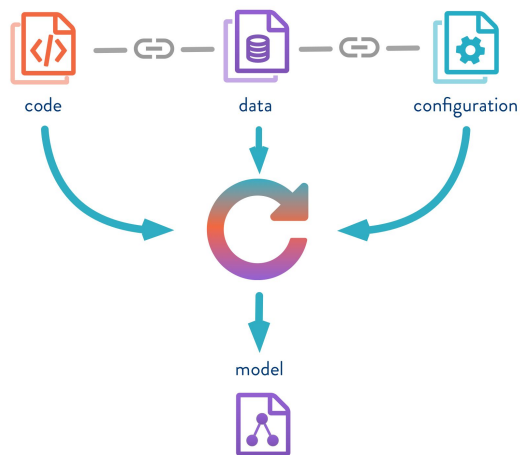
Актуально как для датасетов, так и для готовых моделей

- Локально
- Удалённо
- Распределённо
 - небольшие данные
 - dvc
 - git LFS
 - большие данные
 - парадигма MapReduce
 - стек hadoop или система YTsaurus
 - дата каталог datahubproject.io



Data Version Control (DVC)

- git for data is [DVC](#) (tutorials: [one](#), [two](#))
- Versioning and Access submodules



Модель вычислений

Типы вычислительных мощностей

- Железные
- Виртуальные
 - classical VMs: KVM, vmware
 - docker
- Создаваемые под задачу
 - Очереди задач (slurm, clearml)
 - MapReduce
 - Kubernetes, k8s (Kubeflow)
 - Serverless computing (Amazon Lambda)



docker



kubernetes

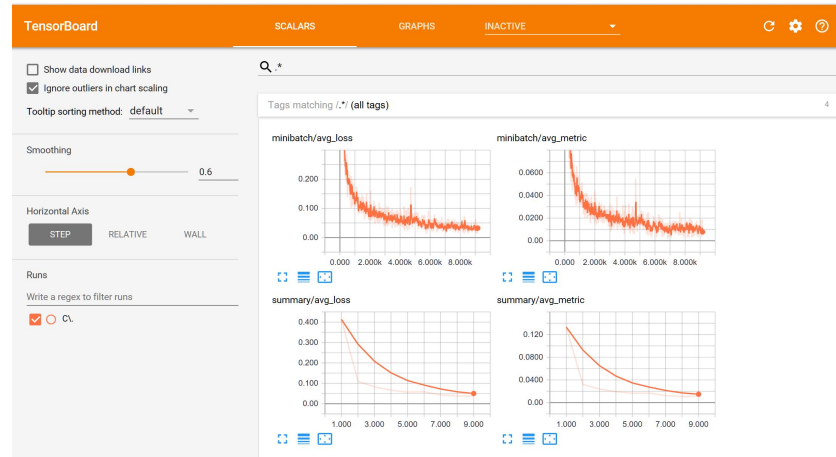


Логирование и визуализация

Тем важнее, чем больше экспериментов и крупнее каждый из них

Подходит для экспериментов, не для логирования прода

- отсутствие
- print
- local service
 - tensorboard
- remote service
 - ml-flow
 - clear-ml
 - kubeflow
 - w&b, neptune и прочая проприетарщина



Experiments Tracker

- Tensorboard
- [MLFlow](#)
- [ClearML](#)
- [sacred](#)
- [Kubeflow](#)
- [neptune.ai](#)
- [weights and biases](#)
-



Регулярные запуски кода

Дефакто стандарт индустрии это airflow

Для других стеков применяются свои инструменты



Apache
Airflow

Разметка данных

- Self-hosted
 - cvat
 - and many more
- Cloud
 - toloka
 - mturk



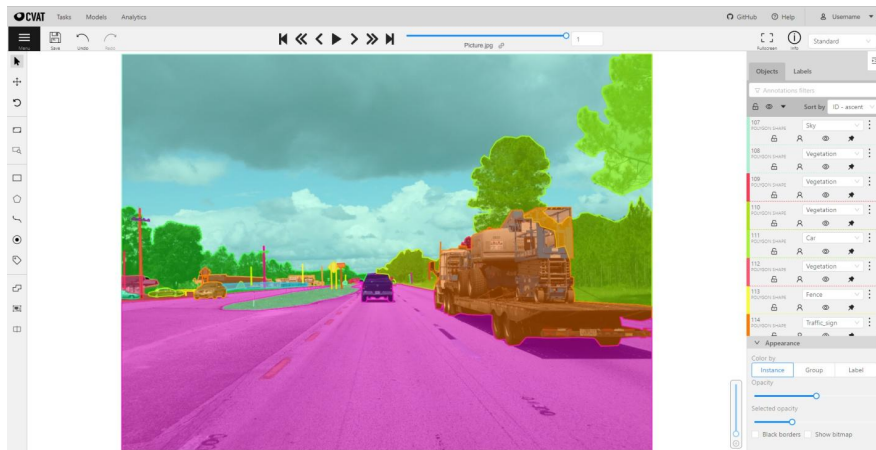
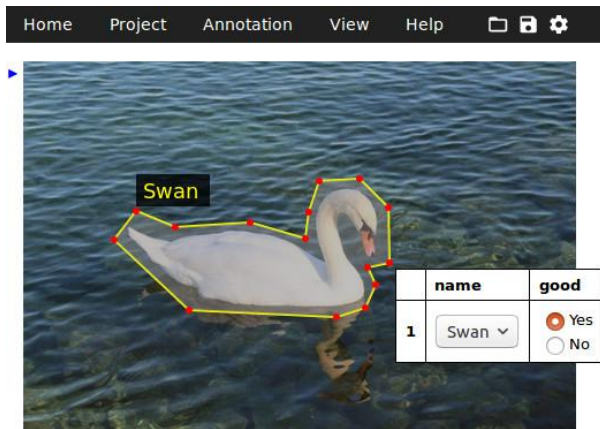
Self-hosted

Solutions specific to Computer Vision

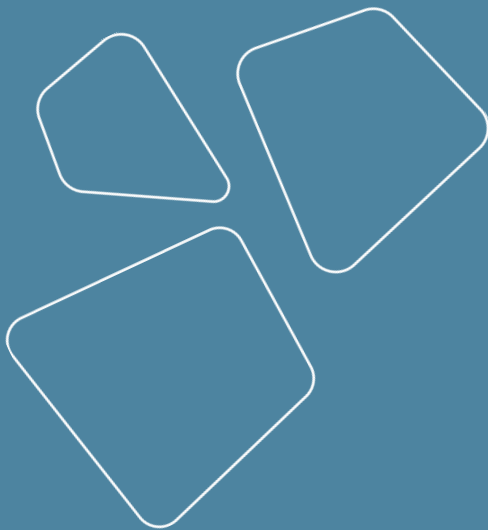
- simple cases - [VIA](#) (free software, standalone)
- scalable solution - [CVAT](#) (free software, server based)
- special cases - [hasty.ai](#) (proprietary, server based)

All of them are web based

Suggest your favorite tools in comments!
(especially for other tasks)



О чём поговорили



- Определение и важность MLOps
- Хранение кода
- Хранение данных
- Модель вычислений
- Логирование + визуализация
- Регулярные запуски кода

Не вошло сегодня:

- Параллелизация GPU
- Разметка данных
- Инференс
- Сохранение артефактов + восстановление

Спасибо за внимание!

Жду вопросов и обсуждений

