

УДК 004

АНАЛИЗ И РАЗРАБОТКА КОНВЕЙЕРА MLOPS ДЛЯ РАЗВЕРТЫВАНИЯ МОДЕЛЕЙ МАШИННОГО ОБУЧЕНИЯ

Р. Р. Ямиков^{1, [0000-0001-9240-5168]}, **К. А. Григорян**^{2, [0000-0001-6470-1832]}

^{1, 2}Казанский (Приволжский) федеральный университет, ул. Кремлевская, 35, г. Казань, 420008

¹jamrustem@yandex.ru, ²karigri@yandex.ru

Аннотация

Рост числа IT-продуктов с внедренными элементами машинного обучения (Machine Learning – ML) обуславливает повышение актуальности автоматизации процессов машинного обучения. Использование методов MLOps направлено на обеспечение обучения и эффективного развертывания приложений с производственной среде, автоматизируя решение побочных инфраструктурных вопросов слабо связанных с непосредственно разработкой модели.

Мы рассматриваем компоненты, принципы и подходы MLOps и анализируем существующие платформы и решения для построения конвейеров машинного обучения. Кроме того, предлагаем подход к построению конвейера машинного обучения на основе основных инструментов DevOps и библиотек с открытым исходным кодом.

Ключевые слова: *MLOps, DevOps, CI/CD, CT, ML, конвейер машинного обучения.*

ВВЕДЕНИЕ

На текущий момент растет сложность полноценной разработки приложения с машинным обучением. Полностековому специалисту необходимо изучать достаточно широкую область знаний, требуется обладать не только навыками в области науки о данных, но и иных областях, таких как инфраструктура машинного обучения, развертывание приложения. Поэтому появляется тенденция к росту спроса на услуги MLOps-инженеров. Таким образом, сегодня сфера MLOps имеет высокую актуальность и востребованность среди организаций, связанных с управлением и обработкой данных.

Согласно проведенному опросу «The State of ML 2020» 331 специалиста по машинному обучению из 63 различных стран, до 40% опрошенных занимаются как непосредственно работами с проработкой моделей, так и инфраструктурными вопросами. Одной из частых проблем, с которой сталкивались респонденты в ходе работы, являлись процессы, связанные с развертыванием моделей в производственной среде [1]. В виду этого множество проектов с использованием машинного обучения терпит неудачу на стадиях проверки концепции (proof of concept) и экспериментирования еще до внедрения в производство [2]. Провалы обуславливается тем, что специалисты машинного обучения уделяют основное внимание проектированию и построению ML-модели, а не созданию готового продукта с машинным обучением. Кроме того, системы машинного обучения довольно сложны, и их бывает трудно соединить с инфраструктурой для использования в производственной среде [3].

MLOps помогает решить проблему внедрения моделей машинного обучения в производственную среду путем автоматизации как процессов машинного обучения, так и процессов развертывания ML-моделей в производстве.

ОБЗОР ЛИТЕРАТУРЫ

MLOps – это сборник различных методов, практик и инструментов для развертывания моделей машинного обучения в производстве [4]. MLOps можно рассматривать как пересечение практик машинного обучения и DevOps. DevOps – это методология, которая включает в себя практики автоматизации процессов сборки, настройки и развёртывания программного обеспечения (ПО) и объединяет рабочие процессы разработки ПО с процессами тестирования и эксплуатации для минимизации времени выпуска ПО [5].

В основе методов MLOps лежат принципы методологии DevOps. Основными из них являются непрерывная интеграция (CI) и непрерывное развертывание (CD). Непрерывная интеграция – это практика разработки ПО, которая заключается в постоянной интеграции программного кода и выполнении автоматизированной сборки продукта через частые промежутки времени для постоянного тестирования текущего состояния кода и скорейшего исправления ошибок [6]. Непрерывная доставка – подход к разработке ПО, при котором разработка проходит короткими итерациями, когда постоянно и автоматизировано выпускается новая

стабильная версия продукта для тестирования [7]. MLOps кроме непрерывной доставки и непрерывного развертывания включает в себя еще и непрерывное обучение (СТ) – переобучение ML-модели при необходимости.

Уровни зрелости MLOps

В зависимости от степени автоматизации процессов продукта с машинным обучением принято относить его к одному из уровней зрелости процесса MLOps [8]. Компании Google и Microsoft выделяют две основные классификации по уровню зрелости [6].

В Google выделяют три уровня зрелости процесса MLOps-проектов по степени автоматизации шагов процесса доставки модели машинного обучения в производство [9]:

1. Ручной процесс;
2. Автоматизация конвейера машинного обучения;
3. Автоматизация конвейера CI/CD.

Microsoft в свою очередь выделяет в классификации пять уровней [10]:

1. Отсутствие процессов MLOps;
2. Наличие DevOps, но без MLOps;
3. Автоматизация обучения ML-модели;
4. Автоматизация развертывания ML-модели;
5. Полная автоматизация процессов MLOps.

Принципы MLOps

В MLOps выделяют девять принципов или наилучших практик для разработки продуктов с машинным обучением [11]:

1. Автоматизация CI/CD;
2. Оркестрация рабочих процессов – координация порядка выполнения задач конвейера машинного обучения;
3. Воспроизводимость;
4. Версионирование – отслеживание данных, модели, кода в системах контроля версий для обеспечения воспроизводимости и аудита;
5. Коллаборация – возможность совместной работы над проектом машинного обучения;

6. Непрерывное обучение и оценка ML-модели;
7. Отслеживание метаданных – каждой итерации обучения модели, ее параметров, метрик;
8. Мониторинг;
9. Обратная связь – по результатам оценок качества модели, мониторинга развернутой модели.

Компоненты MLOps

Для реализации принципов MLOps используются следующие компоненты системы MLOps [11], [12]:

1. Компонент CI/CD (реализует принципы 1, 6, 9);
2. Репозиторий программного кода (принципы 4, 5);
3. Компонент оркестрации рабочих процессов (принципы 2, 3, 6);
4. Система хранения функций и данных (feature store, принципы 3, 4);
5. Инфраструктура для обучения моделей (принцип 6);
6. Реестр моделей (принципы 3, 4);
7. Хранилище метаданных машинного обучения (принципы 4, 7);
8. Компонент обслуживания моделей (принцип 1);
9. Компонент мониторинга (принципы 8, 9).

Итеративно-инкрементный процесс MLOps

Согласно сайту MLOps полный итеративно-инкрементный процесс MLOps включает в себя три основных этапа [12]:

1. Проектирование приложения на базе машинного обучения – происходят сбор требований, определение проблематики бизнеса и дальнейшее проектирование модели машинного обучения для решения проблемы пользователя и повышения его производительности. Дополнительно идут оценка данных для обучения алгоритма и разработка архитектуры модели машинного обучения.

2. Эксперименты и разработка машинного обучения – проверка применимости алгоритма машинного обучения, происходит проверка концепции. Целью этапа будет получение стабильной модели требуемого качества для производственной среды.

3. Операции машинного обучения – непосредственно само развертывание обученной модели в производство с использованием методов DevOps.

ОБЗОР СУЩЕСТВУЮЩИХ ПЛАТФОРМ

Yandex DataSphere – это облачный сервис для разработки и дальнейшей эксплуатации моделей машинного обучения, который предоставляет все необходимые инструменты и ресурсы для полного цикла разработки машинного обучения [13].

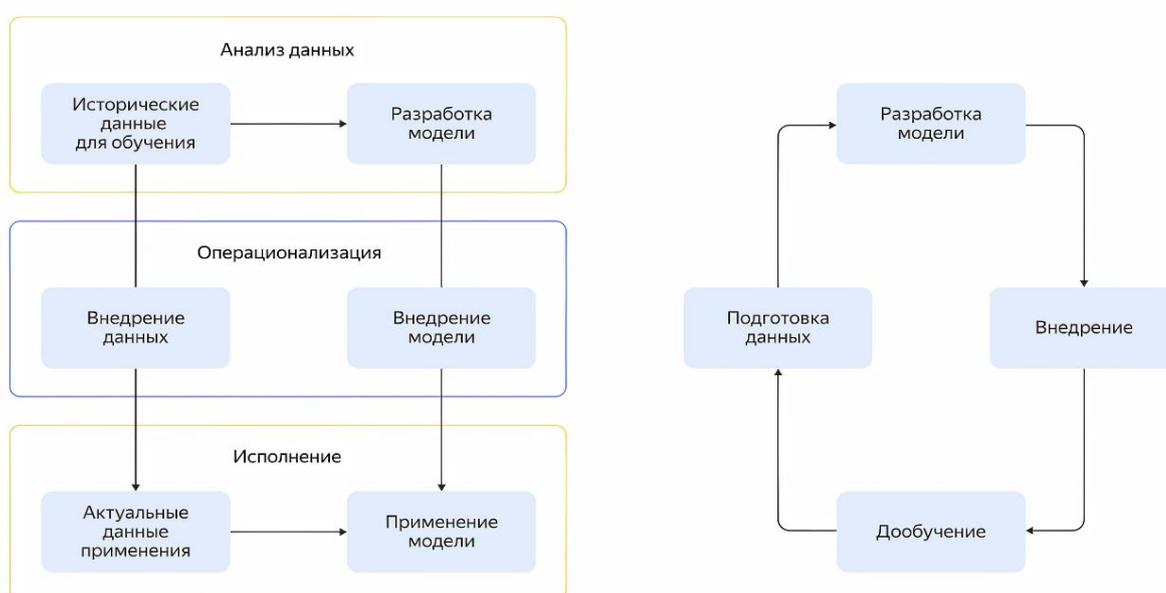


Рисунок 1. Полный цикл машинного обучения [13]

Разработка ML-моделей происходит в среде для интерактивных вычислений Jupyter Notebook. Блокнот содержит набор из множества ячеек, в каждой из которых независимо выполняется код [14].

Проекты в Yandex DataSphere представляют собой блокнот Jupyter. В них происходит сохранение полного состояния блокнота, включая переменные, установленные пакеты и др. [14].

В проект Yandex DataSphere данные можно загружать как через интерфейс (при небольшом объеме), так и через сетевые хранилища и базы данных [14].

Yandex DataSphere сохраняет состояния ноутбука в проекте при помощи системы контрольных точек, в которых хранятся как данные блокнота (код ячеек, вывод и значения переменных в определенный момент времени), так и данные хранилища проекта [14].

После разработки моделей их можно легко развернуть в виде микросервисов. Предобученные модели разворачиваются на инстансах — виртуальных машинах, где зафиксированы состояние интерпретатора и код модели. Затем эти инстансы объединяются в ноды (группы виртуальных машин). Для доступа к ним используется API [15].

Главный недостаток сервиса заключается в том, что это коммерческое ПО. Кроме того, сервис скорее рассчитан на экспериментирование с машинным обучением, чем на полноценный конвейер с оркестрацией и развертыванием модели.

MLFlow — это платформа с открытым исходным кодом, для управления целиком всем жизненным циклом машинного обучения [16]. Фреймворк включает в себя решение задач с экспериментами, воспроизводимостью и развертыванием, так же содержит центральный реестр моделей.

Платформа MLFlow поддерживает интеграцию с Docker и Kubernetes и интегрирована с основными фреймворками машинного обучения. Содержит четыре основных модуля, которые можно использовать независимо друг от друга:

- MLflow Tracking – необходим для записи экспериментов, параметров моделей, версионирования кода, логирования метрик моделей для дальнейшей визуализации в пользовательском интерфейсе [17];
- MLflow Projects – упаковка кода проекта для дальнейшего использования и воспроизведения конвейеров машинного обучения; проект описан в файле MLProject в формате yaml [18];
- MLflow Models – упаковка, хранение и развертывание моделей машинного обучения в различных средах; поддерживает развертывание в виде REST API эндпоинтов и упаковку в Docker-образы [19];
- Model Registry – хранение, аннотирование и управление жизненным циклом модели в центральной репозитории моделей [20].

Среди крупных недостатков MLFlow выделяется проблема отсутствия ролей

пользователей и функционала безопасности, что приводит к затруднениям совместной работы над моделями [21]. Кроме того, еще одни из проблем MLFlow – сложности с развёртыванием моделей на различных платформах и отсутствие мониторинга производительности моделей [21], [22].

KubeFlow — это бесплатная платформа для машинного обучения с открытым исходным кодом, которая делает развёртывание конвейеров рабочих процессов машинного обучения в контейнерах Kubernetes простым и легко масштабируемым [23]. Платформа предоставляет полный набор инструментов для оркестрации стека машинного обучения для использования при развёртывании, масштабировании и управлении системами машинного обучения.

Особенности платформы Kubeflow [24]:

- сервисы для создания интерактивных блокнотов Jupyter и управления ими при процессе обработке данных;
- операторы для обучения моделей машинного обучения TensorFlow, настройки гиперпараметров и управления нагрузками;
- обслуживание моделей ML – экспорт обученных моделей в Kubernetes через TensorFlow Serving; также есть интеграция с Seldon Core для развёртывания моделей;
- конвейер Kubeflow Pipelines — платформа для создания и развёртывания масштабируемых рабочих процессов машинного обучения на основе контейнеров Docker;
- поддержка различных фреймворков машинного обучения: TensorFlow, PyTorch, Apache MXNet, MPI, XGBoost, Chainer и др.

Kubeflow содержит множество инструментов и может быть представлен как платформа для размещения компонентов системы машинного обучения поверх Kubernetes [25].

К недостаткам Kubeflow относится отсутствие версионирования данных и конвейера [22]. Кроме этого, Kubeflow сложно настроить, у него высокий порог входа, а для его использования необходимы глубокие знания в Kubernetes [26].

DVC и CML

DVC (data version control) – инструмент с открытым исходным кодом для

управления версиями данных проектов машинного обучения, основанный на уже существующем инструментарии Git, CI/CD и т. д. Система DVC позволяет создавать воспроизводимые конвейеры машинного обучения; отслеживает и обрабатывает файлы, наборы данных, модели машинного обучения и их метрики в виде кода [27].

DVC – это инструмент, работающий совместно с Git. DVC использует файл `dvc` для контроля артефактов машинного обучения, а Git отвечает за контроль версий кода и файла `dvc` [28].

Этапы конвейера машинного обучения описаны в файле `dvc.yaml` с указанием используемых скриптов Python для каждого шага, его параметров, используемых и генерируемых файлов [29].

CML (continuous machine learning) – библиотека с открытым исходным кодом для реализации непрерывной интеграции и непрерывной доставки (CI/CD) в проектах машинного обучения. Библиотека используется для автоматизации процесса разработки, обучения и оценки моделей, сравнения экспериментов с моделями машинного обучения и отслеживания изменений наборов данных [30].

CML может применяться совместно с DVC, при этом DVC будет управлять данными и моделями, а CML – оркестровкой инфраструктуры, тестированием и мониторингом [31].

CML интегрируется в GitHub Actions или GitLab CI/CD и работает совместно с ними, создавая совместный конвейер.

CML использует GitFlow на основе рабочего процесса Git для управления экспериментами через версионирование DVC данных и моделей. CML может создавать отчеты в пулл-реквестах с метриками и графиками оценок моделей, позволяя команде принимать информированные решения [32].

Среди недостатков связки DVC и CML выделяется то, что это неполноценная платформа, а лишь часть конвейера MLOps, для работы которым дополнительно требуются инструменты CI/CD, развертывания и оркестрации модели, мониторинга модели. Однако это в свою очередь обеспечивает гибкость подхода.

МЕТОДОЛОГИЯ

Для разработки конвейера MLOps нами использовался подход, базирующийся на основных инструментах DevOps и библиотек с открытым исходным кодом. Были использованы следующие технологии для реализации компонентов MLOps:

1. Gitlab в качестве компонента репозитория программного кода;
2. Gitlab CI/CD и CML для компонент CI/CD;
3. Gitlab CI/CD с несколькими gitlab runner для выполнения заданий конвейера обучения модели на высокопроизводительной системе и развертывания на сервере; компоненты оркестрации рабочих процессов и инфраструктура для обучения моделей;
4. DVC и удаленное хранилище в Google Drive для компонентов системы хранения функций и данных, реестра моделей, хранилище метаданных машинного обучения;
5. Микросервис FastApi как компонент обслуживания моделей;
6. Сбор и хранение метрик в Prometheus и их визуализация в виде графиков в Grafana в качестве компонента мониторинга.

Был построен конвейер машинного обучения с указанной ниже блок-схемой для проведения экспериментов (рис. 2). Эксперименты проводились с использованием средств Git и Gitlab на удаленном сервере. Разработчик вносит локально правки к модели, создает коммит и отправляет его в репозиторий Gitlab, где в дальнейшем открывает pull request. Следующие процессы конвейера MLOps по обучению модели происходят удаленно. После завершения работ конвейера результаты публикуются в виде комментария к коммиту в PR. Затем разработчик может закрыть PR и слить ветку в главную, тем самым запустив дальнейшие шаги конвейера по интеграции и развертыванию модели.

Схема конвейера MLOps представлена на рис. 3.

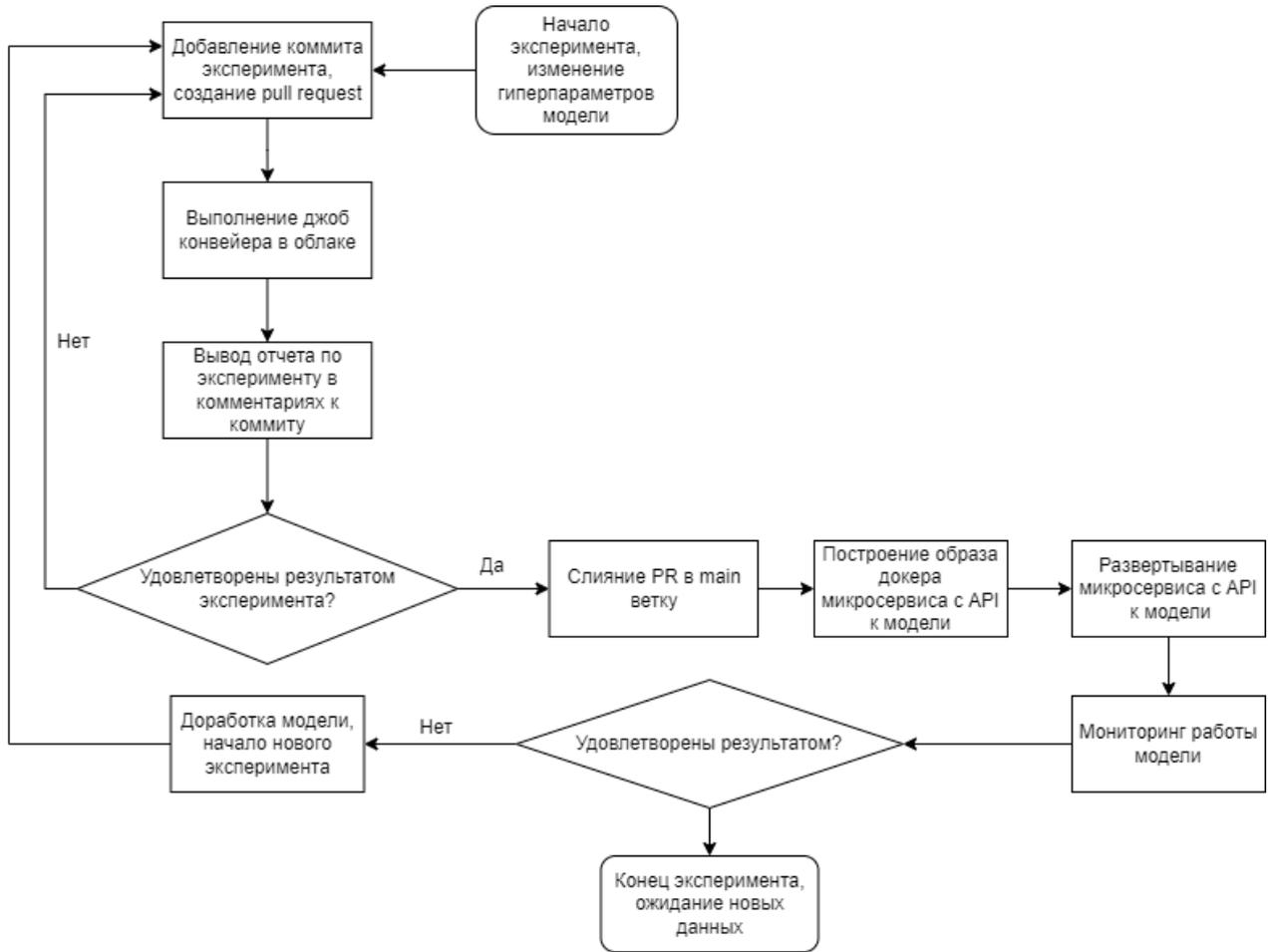


Рисунок 2. Блок схема процесса экспериментирования

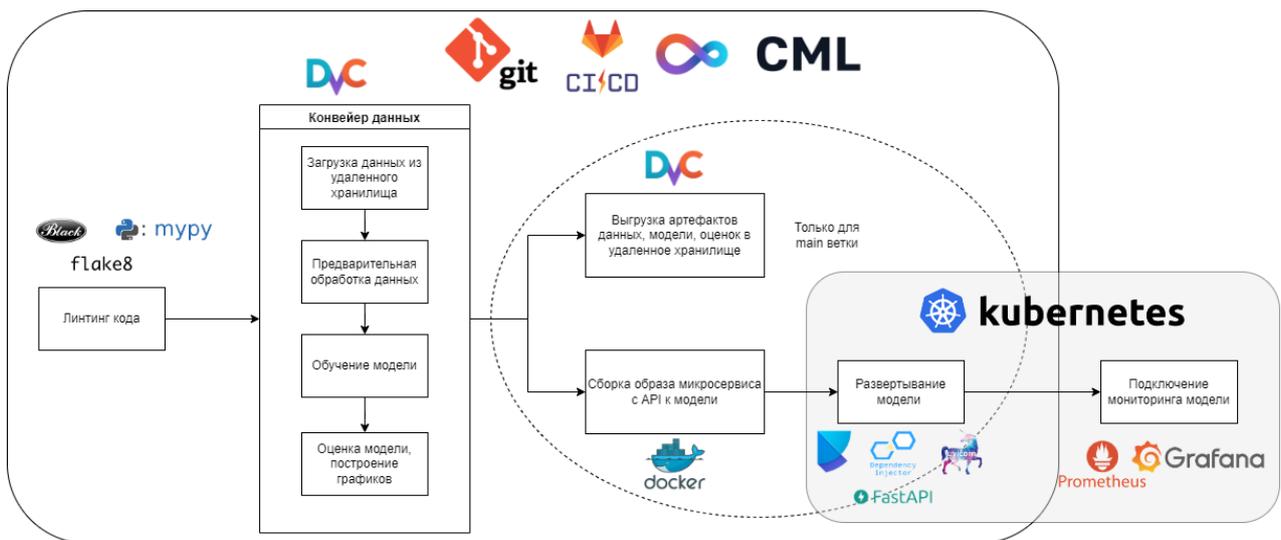


Рисунок 3. Схема конвейера

Конвейер процессов MLOps был разработан с использованием инструментов для интеграции Git, Gitlab CI/CD и CML и включает в себя 4 основных блока:

1. Линтинг программного кода файлов Python процессов обработки данных и построения модели машинного обучения; кода приложения микросервиса к API данной модели. Для синтаксического анализа и форматирования кода были использованы библиотеки: формater кода black под стандарты PEP8, статический анализатор типов туру, статический анализатор flake8;

2. Конвейер данных, состоящий из 4 шагов:

- загрузка данных;
- предварительная обработка данных;
- обучения модели на основе этих данных;
- оценка модели с построением графиков.

Для построения конвейера данных использовалась библиотека DVC.

3. Процессы интеграции:

- выгрузка артефактов в удаленное облачное хранилище, использовался DVC;

- сборка образа микросервиса с API к модели с использованием Docker;

4. Процессы развертывания на сервере в кластере оркестратора Docker контейнеров Kubernetes [33]:

- развертывание микросервиса на асинхронном фреймворке FastApi, с веб-сервером Uvicorn, с использованием библиотеки для внедрения зависимостей Dependency Injector и инструмента для управления библиотеками Poetry;

- подключение к ранее развернутым сервисам для мониторинга; использовался инструмент Prometheus для сбора и хранения событий, метрик и Grafana – для аналитики и визуализации собранных метрик [34] [35].

Стадии конвейера с процессами интеграции и процессами развертывания выполняются только в главной ветке.

РЕЗУЛЬТАТЫ

Был успешно реализован прототип конвейера MLOps. API микросервиса с моделью содержит три эндпоинта (рис. 4):

1. эндпоинт /predict с загрузкой файла для классификации моделью –

возвращает спрогнозированный класс;

2. эндпоинт `/get_experiment_metas`, возвращающий json с метаданными экспериментов: тег коммита эксперимента, метрики и параметры модели машинного обучения;

3. эндпоинт `/replace_ml_model` для «горячей» замены используемой модели машинного обучения без перезапуска сервиса; замена производится по тегу эксперимента, который можно посмотреть через эндпоинт `/get_experiment_metas`.

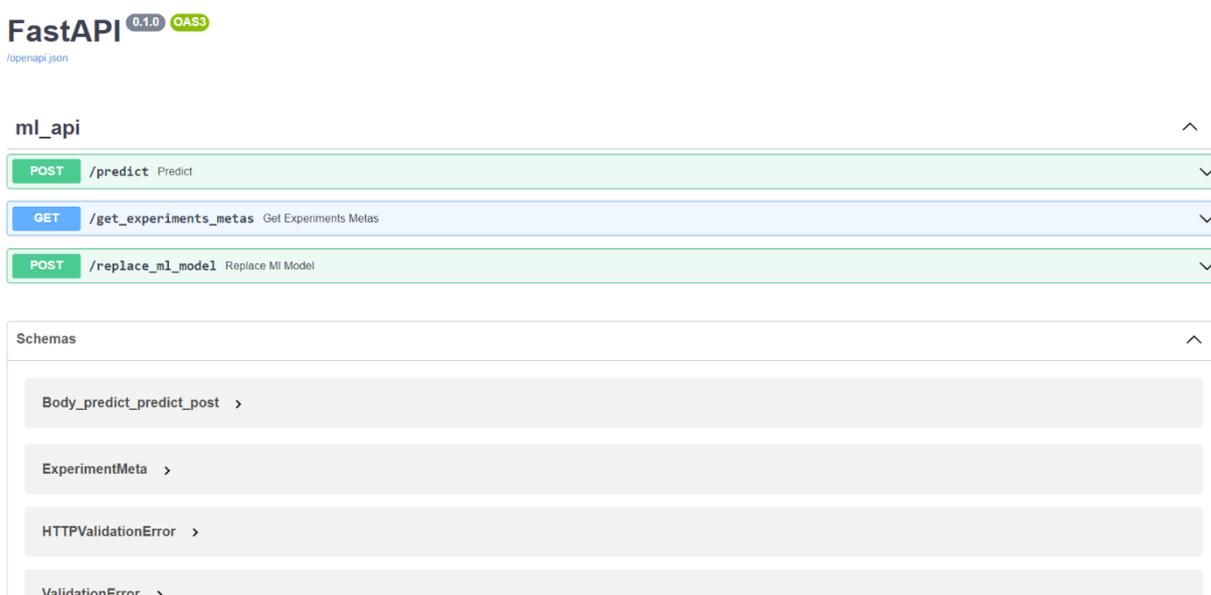


Рисунок 4. Страница интерфейса к API микросервиса

Показатели производительности модели выводятся на графики в Grafana (рис. 5 и 6).

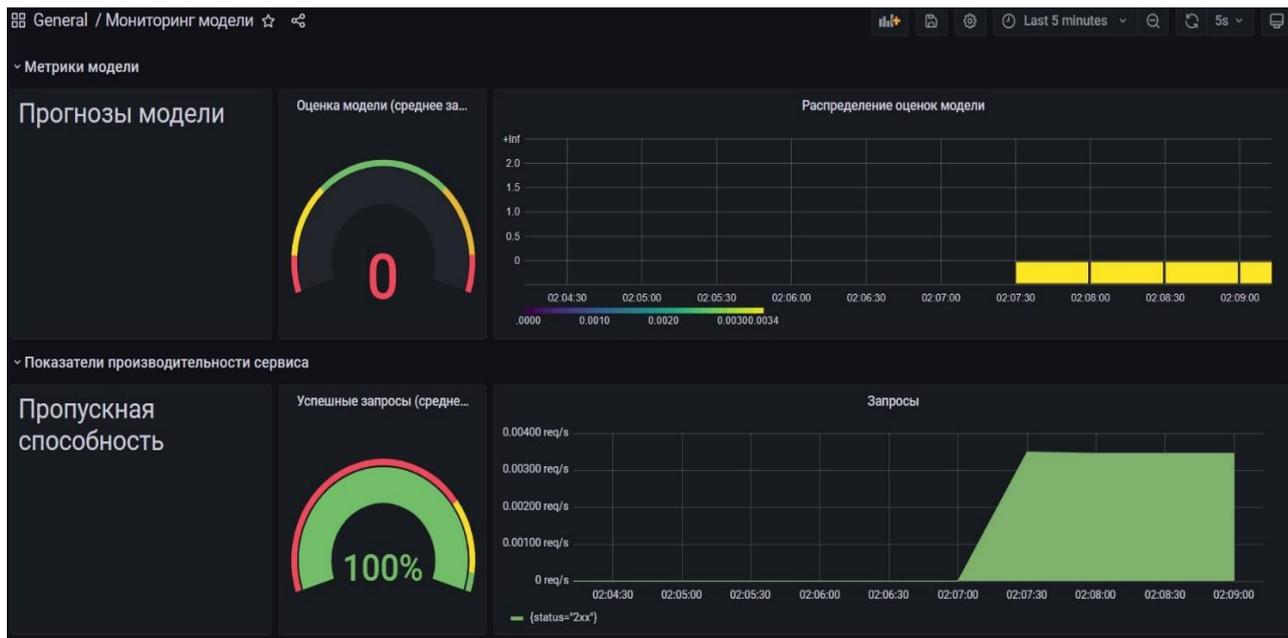


Рисунок 5. Графики метрик микросервиса с моделью



Рисунок 6. Показатели загрузки системы

ЗАКЛЮЧЕНИЕ

В ходе исследования был проведен анализ области MLOps, изучены уровни развития MLOps, его принципы и компоненты. Рассмотрены различные подходы и готовые платформы MLOps.

В результате исследования построен прототип конвейера системы MLOps для автоматизации задач разработки, развертывания и поддержки моделей ма-

шинного обучения на основе основных инструментов DevOps и библиотек с открытым исходным кодом.

В дальнейшем планируется более глубокое внедрение мониторинга и триггеров для дообучения модели.

СПИСОК ЛИТЕРАТУРЫ

1. *Makinen S., Skogstrom H., Laaksonen E., Mikkonen T.* Who Needs MLOps: What Data Scientists Seek to Accomplish and How Can MLOps Help? // Software Engineering for AI (WAIN) of 43rd International Conference on Software Engineering (ICSE). 2021.
2. *Van der Meulen R., McCall T.* Gartner Says Nearly Half of CIOs Are Planning to Deploy Artificial Intelligence // Gartner. 2018. URL: <https://www.gartner.com/en/newsroom/press-releases/2018-02-13-gartner-says-nearly-half-of-cios-are-planning-to-deploy-artificial-intelligence> (дата обращения: 01.06.2022).
3. *Posoldova A.* Machine Learning Pipelines: From Research to Production // IEEE Potentials. 2020. Vol. 39, No. 6. P. 38–42. <https://doi.org/10.1109/MPOT.2020.3016280>
4. *Alla S., Adari S.K.* What is mlops? // In: Beginning MLOps with MLFlow. Berkeley: Apress, 2021. P. 79–124.
5. *Gift N., Deza A.* Practical MLOps. O'Reilly Media, Inc., 2021.
6. *Symeonidis G., Nerantzis E., Kazakis A., Papakostas G.A.* MLOps – Definitions, Tools and Challenges // IEEE Annual Computing and Communication Workshop and Conference (CCWC). Las Vegas. 2022. Vol. 12. P. 453–460. <https://doi.org/10.48550/arXiv.2201.00162>
7. *Chen L.* Continuous Delivery: Huge Benefits, but Challenges Too // IEEE Software. 2015. Vol. 32. P. 50–54. <https://doi.org/10.1109/MS.2015.27>
8. *John M., Olsson H., Bosch J.* Towards MLOps: A Framework and Maturity Model // Euromicro Conference on Software Engineering and Advanced Applications (SEAA). Palermo. 2021. Vol. 47. P. 1–8. <https://doi.org/10.1109/SEAA53835.2021.00050>
9. MLOps: Continuous delivery and automation pipelines in machine learning // Google Cloud. 2021. URL: <https://cloud.google.com/architecture/mlops-continuous-delivery-and-automation-pipelines-in-machine-learning> (дата обращения:

03.07.2021).

10. Machine Learning operations maturity model // Microsoft. URL: <https://docs.microsoft.com/en-us/azure/architecture/example-scenario/mlops/mlops-maturity-model>, last accessed 2022/05/30.

11. Kreuzberger D., Kühl N., Hirschl S. Machine Learning Operations (MLOps): Overview, Definition, and Architecture // arXiv preprint arXiv:2205.02302. 2022. <https://doi.org/10.48550/arXiv.2205.02302>

12. MLOps Principles // MLOps. URL: <https://ml-ops.org/content/mlops-principles> (дата обращения: 03.07.2021).

13. Yandex DataSphere // Yandex Cloud. URL: <https://cloud.yandex.ru/services/datasphere>, (дата обращения: 05.03.2022).

14. Проект // Yandex datasphere документация. URL: <https://cloud.yandex.ru/docs/datasphere/concepts/project>, (дата обращения: 05.03.2022).

15. Развертывание эксплуатации моделей // Yandex datasphere документация. URL: <https://cloud.yandex.ru/docs/datasphere/concepts/deploy>, (дата обращения: 05.03.2022).

16. MLflow. URL: <https://mlflow.org> (дата обращения: 28.12.2021).

17. MLflow Tracking // MLflow. URL: <https://mlflow.org/docs/latest/tracking.html> (дата обращения: 28.12.2021).

18. MLflow Projects // MLflow. URL: <https://mlflow.org/docs/latest/projects.html>, (дата обращения: 28.12.2021).

19. MLflow Models // MLflow. URL: <https://mlflow.org/docs/latest/models.html> (дата обращения: 28.12.2021).

20. MLflow Model Registry // MLflow. URL: <https://mlflow.org/docs/latest/model-registry.html> (дата обращения: 28.12.2021).

21. Khandelwal N. MLflow Alternatives for Data Version Control: DVC vs. MLflow // Censious. URL: <https://censious.ai/blogs/dvc-vs-mlflow> (дата обращения: 30.05.2022).

22. Hewage N., Meedeniya D. Machine Learning Operations: A Survey on MLOps Tool Support // arXiv preprint arXiv:2202.10169. 2022. <https://doi.org/10.48550/arXiv.2202.10169>

23. Introduction // Kubeflow documentation. URL: <https://www.kubeflow.org/docs/started/introduction> (дата обращения: 11.03.2022).

24. What is Kubeflow? // Kubeflow. URL: <https://www.kubeflow.org> (дата обращения: 11.03.2022).

25. Architecture // Kubeflow documentation. URL: <https://www.kubeflow.org/docs/started/architecture> (дата обращения: 11.03.2022).

26. *Kaewsanmua K.* Best 8 Machine Learning Model Deployment Tools That You Need to Know // Neptune. 2021. URL: <https://neptune.ai/blog/best-8-machine-learning-model-deployment-tools> (дата обращения: 01.06.2022).

27. DVC. URL: <https://dvc.org> (дата обращения: 27.12.2021).

28. *Zhao Y.* MLOps: Data versioning with DVC — Part I // Medium. 2020. URL: <https://yizhenzhao.medium.com/mlops-data-versioning-with-dvc-part-i-8b3221df8592> (дата обращения: 27.12.2021).

29. *Mesquita D.* The ultimate guide to building maintainable Machine Learning pipelines using DVC // Towards data science. 2020. URL: <https://towardsdatascience.com/the-ultimate-guide-to-building-maintainable-machine-learning-pipelines-using-dvc-a976907b2a1b> (дата обращения: 27.12.2021).

30. CML Documentation // CML. URL: <https://cml.dev/doc> (дата обращения: 27.12.2021).

31. Continuous Integration and Deployment for Machine Learning // DVC. URL: <https://dvc.org/doc/use-cases/ci-cd-for-machine-learning> (дата обращения: 27.12.2021).

32. Continuous Integration with CML and Github Actions // MLOps Guide. URL: https://mlops-guide.github.io/CICD/cml_testing (дата обращения: 27.12.2021).

33. Kubernetes Documentation // Kubernetes. URL: <https://kubernetes.io/docs/hom> (дата обращения: 22.05.2022).

34. What is Prometheus? // Prometheus. URL: <https://prometheus.io/docs/introduction/overview>

35. Grafana // Grafana Labs. URL: <https://grafana.com/grafana> (дата обращения: 22.05.2022).

ANALYSIS AND DEVELOPMENT OF THE MLOPS PIPELINE FOR ML MODEL DEPLOYMENT

Rustem Yamikov¹[0000-0001-9240-5168], Karen Grigoryan²[0000-0001-6470-1832]

^{1, 2} Kazan (Volga Region) Federal University, 35 Kremlevskaya str., Kazan, 420008

¹jamrustem@yandex.ru, ²karigri@yandex.ru

Abstract

The growth in the number of IT products with machine-learning features is increasing the relevance of automating machine-learning processes. The use of MLOps techniques is aimed at providing training and efficient deployment of applications in a production environment by automating side infrastructure issues that are not directly related to model development.

In this paper, we review the components, principles, and approaches of MLOps and analyze existing platforms and solutions for building machine learning pipelines. In addition, we propose an approach to build a machine learning pipeline based on basic DevOps tools and open-source libraries.

Keywords: MLOps, DevOps, CI/CD, CT, ML, machine learning pipeline.

REFERENCES

1. Makinen S., Skogstrom H., Laaksonen E., Mikkonen T. Who Needs MLOps: What Data Scientists Seek to Accomplish and How Can MLOps Help? // Software Engineering for AI (WAIN) of 43rd International Conference on Software Engineering (ICSE). 2021.
2. Van der Meulen R., McCall T. Gartner Says Nearly Half of CIOs Are Planning to Deploy Artificial Intelligence // Gartner. 2018. URL: <https://www.gartner.com/en/newsroom/press-releases/2018-02-13-gartner-says-nearly-half-of-cios-are-planning-to-deploy-artificial-intelligence>, last accessed 2022/06/01.
3. Posoldova A. Machine Learning Pipelines: From Research to Production // IEEE Potentials. 2020. Vol. 39, No. 6. P. 38–42. <https://doi.org/10.1109/MPOT.2020.3016280>
4. Alla S., Adari S.K. What is mlops? // In: Beginning MLOps with MLFlow.

Berkeley: Apress, 2021. P. 79–124.

5. *Gift N., Deza A.* Practical MLOps. O'Reilly Media, Inc., 2021.
6. *Symeonidis G., Nerantzis E., Kazakis A., Papakostas G.A.* MLOps - Definitions, Tools and Challenges // IEEE Annual Computing and Communication Workshop and Conference (CCWC). Las Vegas. 2022. Vol. 12. P. 453–460.
<https://doi.org/10.48550/arXiv.2201.00162>
7. *Chen L.* Continuous Delivery: Huge Benefits, but Challenges Too // IEEE Software. 2015. Vol. 32. P. 50–54. <https://doi.org/10.1109/MS.2015.27>
8. *John M., Olsson H., Bosch J.* Towards MLOps: A Framework and Maturity Model // Euromicro Conference on Software Engineering and Advanced Applications (SEAA). Palermo. 2021. Vol. 47. P. 1–8.
<https://doi.org/10.1109/SEAA53835.2021.00050>
9. MLOps: Continuous delivery and automation pipelines in machine learning // Google Cloud. 2021. URL: <https://cloud.google.com/architecture/mlops-continuous-delivery-and-automation-pipelines-in-machine-learning>, last accessed 2021/07/03.
10. Machine Learning operations maturity model // Microsoft. URL: <https://docs.microsoft.com/en-us/azure/architecture/example-scenario/mlops/mlops-maturity-model>, last accessed 2022/05/30.
11. *Kreuzberger D., Kühl N., Hirschl S.* Machine Learning Operations (MLOps): Overview, Definition, and Architecture // arXiv preprint arXiv:2205.02302. 2022.
<https://doi.org/10.48550/arXiv.2205.02302>
12. MLOps Principles // MLOps. URL: <https://ml-ops.org/content/mlops-principles>, last accessed 2021/07/03.
13. Yandex DataSphere // Yandex Cloud. URL: <https://cloud.yandex.ru/services/datasphere>, last accessed 2022/03/05.
14. Proekt // Yandex datasphere dokumentaciya. URL: <https://cloud.yandex.ru/docs/datasphere/concepts/project>, last accessed 2022/03/05.
15. Razvertyvanie ekspluatatsii modelej // Yandex datasphere dokumentaciya. URL: <https://cloud.yandex.ru/docs/datasphere/concepts/deploy>, last accessed 2022/03/05.
16. MLFlow. URL: <https://mlflow.org>, last accessed 2021/12/28.

17. MLflow Tracking // MLflow. URL: <https://mlflow.org/docs/latest/tracking.html>, last accessed 2021/12/28.

18. MLflow Projects // MLflow. URL: <https://mlflow.org/docs/latest/projects.html>, last accessed 2021/12/28.

19. MLflow Models // MLflow. URL: <https://mlflow.org/docs/latest/models.html>, last accessed 2021/12/28.

20. MLflow Model Registry // MLflow. URL: <https://mlflow.org/docs/latest/model-registry.html>, last accessed 2021/12/28.

21. *Khandelwal N.* MLflow Alternatives for Data Version Control: DVC vs. MLflow // Censious. URL: <https://censious.ai/blogs/dvc-vs-mlflow>, last accessed 2022/05/30.

22. *Hewage N., Meedeniya D.* Machine Learning Operations: A Survey on MLOps Tool Support // arXiv preprint arXiv:2202.10169. 2022. <https://doi.org/10.48550/arXiv.2202.10169>

23. Introduction // Kubeflow documentation. URL: <https://www.kubeflow.org/docs/started/introduction>, last accessed 2022/03/11.

24. What is Kubeflow? // Kubeflow. URL: <https://www.kubeflow.org>, last accessed 2022/03/11.

25. Architecture // Kubeflow documentation. URL: <https://www.kubeflow.org/docs/started/architecture>, last accessed 2022/03/11.

26. *Kaewsanmua K.* Best 8 Machine Learning Model Deployment Tools That You Need to Know // Neptune. 2021. URL: <https://neptune.ai/blog/best-8-machine-learning-model-deployment-tools>, last accessed 2022/06/01.

27. DVC. URL: <https://dvc.org>, last accessed 2021/12/27.

28. *Zhao Y.* MLOps: Data versioning with DVC — Part I // Medium. 2020. URL: <https://yizhenzhao.medium.com/mlops-data-versioning-with-dvc-part-i-8b3221df8592>, last accessed 2021/12/27.

29. *Mesquita D.* The ultimate guide to building maintainable Machine Learning pipelines using DVC // Towards data science. 2020. URL: <https://towardsdatascience.com/the-ultimate-guide-to-building-maintainable-machine-learning-pipelines-using-dvc-a976907b2a1b>, last accessed 2021/12/27.

30. CML Documentation // CML. URL: <https://cml.dev/doc>, last accessed

2021/12/27.

31. Continuous Integration and Deployment for Machine Learning // DVC. URL: <https://dvc.org/doc/use-cases/ci-cd-for-machine-learning>, last accessed 2021/12/27.

32. Continuous Integration with CML and Github Actions // MLOps Guide. URL: https://mlops-guide.github.io/CICD/cml_testing, last accessed 2021/12/27.

33. Kubernetes Documentation // Kubernetes. URL: <https://kubernetes.io/docs/home>, last accessed 2022/05/22.

34. What is Prometheus? // Prometheus. URL: <https://prometheus.io/docs/introduction/overview>, last accessed 2022/05/22.

35. Grafana // Grafana Labs. URL: <https://grafana.com/grafana>, last accessed 2022/05/22.

СВЕДЕНИЯ ОБ АВТОРАХ



ЯМИКОВ Рустем Рафикович – магистрант, Казанский (Приволжский) федеральный университет, г. Казань.

Rustem Raficovich YAMIKOV – Master’s student, Kazan (Volga region) Federal University, Kazan.

Email: jamrustem@yandex.ru

ORCID: 0000-0001-9240-5168



ГРИГОРЯН Карен Альбертович – кандидат экономических наук, доцент, Казанский (Приволжский) федеральный университет, г. Казань.

Karen Albertovich GRIGORIAN – Candidate of Economics, Associate Professor, Kazan (Volga region) Federal University, Kazan.

Email: karigri@yandex.ru

ORCID: 0000-0001-6470-1832

Материал поступил в редакцию 25 мая 2022 года
