
УТВЕРЖДЕНА

Решением Ученого совета
АНО ВО «Центральный университет»
«25» мая 2026 г.
Протокол № 2

**Рабочая программа дисциплины (модуля)
«Эксплуатация и внедрение моделей машинного обучения (Production ML)»**

Направление подготовки: 02.04.01 Математика и компьютерные науки

Направленность (профиль) подготовки: Прикладное машинное
обучение и интеллектуальные системы

Квалификация (степень) выпускника: магистр

Форма обучения: очная

Срок освоения программы: 2 года

Год набора: 2026

**Москва
2026**

Содержание

1. Краткая характеристика дисциплины (модуля)	3
2. Перечень планируемых результатов обучения.....	5
3. Тематический план.....	7
4. Содержание дисциплины (модуля).....	7
5. Учебно-методическое обеспечение	9
6. Материально-техническое обеспечение	9
7. Методические и оценочные материалы	11

1. Краткая характеристика дисциплины (модуля)

Рабочая программа дисциплины (модуля) «Эксплуатация и внедрение моделей машинного обучения (Production ML)» составлена в соответствии с федеральным государственным образовательным стандартом высшего образования – магистратура по направлению 02.04.01 Математика и компьютерные науки, профиль Прикладное машинное обучение и интеллектуальные системы, утвержденный приказом Министерства науки и высшего образования Российской Федерации № 810 от 23.08.2017 года.

Изучение дисциплины (модуля) «Эксплуатация и внедрение моделей машинного обучения (Production ML)» позволяет студентам получить практические навыки, необходимые для успешной интеграции машинного обучения в бизнес-процессы, что критически важно для достижения конкурентных преимуществ. Кроме того, оно способствует пониманию вызовов и решений, связанных с масштабированием и поддержкой ML-моделей в условиях реального времени.

Место дисциплины (модуля) в структуре образовательной программы

Настоящая дисциплина (модуль) включена в учебный план по программе подготовки магистратуры по направлению 02.04.01 Математика и компьютерные науки, профиль Прикладное машинное обучение и интеллектуальные системы и входит в обязательную часть Блока 1.

Дисциплина (модуль) изучается на 2 курсе в 3 семестре.

Цель изучения дисциплины (модуля): освоение студентами методов и практик внедрения, развертывания и поддержки моделей машинного обучения в реальных производственных системах.

Задачи изучения дисциплины (модуля):

- научиться обеспечивать повторяемость результатов экспериментов в машинном обучении через систематизацию процессов;
- освоить последовательность стадий создания и эксплуатации моделей машинного обучения в реальных условиях;
- разработать навыки организации хранения и контроля версий данных для эффективного обучения моделей;
- изучить различия в подходах к обучению и развертыванию моделей в оффлайн и онлайн режимах;
- приобрести умения по адаптации моделей машинного обучения для работы в масштабируемых системах.

В результате освоения дисциплины (модуля) обучающийся должен:

знать:

- как обеспечить воспроизводимость в машинном обучении;
- основные этапы жизненного цикла моделей;
- как устроен процесс управления данными для обучения, включая версионирование и обеспечение качества;
- отличия между оффлайн и онлайн моделями и как планировать задачи для их обучения и внедрения;
- как подготовить модели машинного обучения к масштабированию.

уметь:

- организовывать трекинг кода и результатов экспериментов;
- работать с пайплайнами обработки данных и обучения моделей;
- деплоить модели;
- автоматизировать процесс обучения и деплоя модели;
- разрабатывать тесты для проверки качества моделей и настраивать мониторинг для контроля их работы.

владеть:

- навыком выстраивания MLOps процесса (внедрение и поддержка работы моделей машинного обучения в продакшне);
- навыком работы с с MLOps и LLMOps инструментами.

2. Перечень планируемых результатов обучения

Компетенции, формируемые в результате освоения дисциплины (модуля) при проведении учебных занятий в форме контактной работы обучающихся с педагогическими работниками Университета и в форме самостоятельной работы обучающихся:

Компетенция	Содержание компетенции	Индикатор компетенции	Перечень планируемых результатов обучения по дисциплине (модулю)
ОПК-1.	Способен находить, формулировать и решать актуальные и значимые проблемы прикладной и компьютерной математики	ОПК-1.1.	Знает основные методы и подходы к решению задач прикладной и компьютерной математики, включая алгоритмы, математическое моделирование и теорию оптимизации, а также современные инструменты и технологии, используемые в этой области
		ОПК-1.2.	Умеет анализировать и формулировать математические задачи, применять соответствующие методы и алгоритмы для их решения, а также интерпретировать и представлять результаты в понятной и доступной форме
		ОПК-1.3.	Имеет практический опыт работы над проектами или исследованиями в области прикладной и компьютерной математики, включая участие в конкурсах, олимпиадах или научных публикациях, где были решены актуальные и значимые задачи
ПК-1.	Способен определять общие формы и закономерности области прикладного машинного обучения и интеллектуальных систем	ПК-1.1.	Знает основные теоретические концепции и принципы, относящиеся к области прикладного машинного обучения и интеллектуальных систем, а также ключевые закономерности и модели, которые помогают в анализе и интерпретации данных
		ПК-1.2.	Умеет проводить систематический анализ области разработки, выявлять и формулировать общие закономерности и тенденции, а также применять методы исследования для получения новых знаний и понимания
		ПК-1.3.	Имеет практический опыт работы в области прикладного машинного обучения и

			интеллектуальных систем, включая участие в научных проектах, исследованиях или практических заданиях, где были выявлены и описаны общие формы и закономерности
ПК-6.	Способен разрабатывать программное обеспечение для решения прикладных задач в сфере машинного обучения и интеллектуальных систем	ПК-6.1.	Знает основные языки программирования, методы разработки программного обеспечения, а также принципы проектирования и архитектуры программных систем, применяемых в машинном обучении и интеллектуальных системах
		ПК-6.2.	Умеет анализировать прикладные задачи, разрабатывать алгоритмы и реализовывать их в виде программного обеспечения, используя современные инструменты и технологии, а также проводить тестирование и отладку созданных решений
		ПК-6.3.	Имеет практический опыт разработки программного обеспечения в рамках реальных проектов, включая участие в командах, где были успешно реализованы решения для конкретных прикладных задач в сфере профессиональной деятельности

3. Тематический план

№ п/п	Наименование раздела дисциплины (модуля)	Трудоемкость, академические часы				ТКУ (текущий контроль успеваемости)
		Очная форма				
		Аудиторная работа		Контроль	Самостояте льная работа	
		Лекции	Семинары (практичес кие занятия)			
1	Машинное обучение в индустрии. Базовые инструменты инженера машинного обучения	3	3		16	Домашние задания Проект
2	Работа с данными в машинном обучении и трекинг экспериментов машинного обучения	4	4		17	Домашние задания Проект
3	Подготовка моделей машинного обучения к деплою	4	4		17	Домашние задания Проект
4	Мониторинг	4	4	4	17	Проект
5	Автоматизация обучения и деплоя моделей машинного обучения	5	5		17	Домашние задания Проект
6	Масштабирование и отказоустойчивость приложений машинного обучения	5	5		19	Домашние задания Проект
7	LLMOps	5	5		19	Домашние задания Проект
	Зачет с оценкой			4		Творческое задание
Итого:		30	30	8	122	
Объем дисциплины (модуля) (в ак. ч.)		190				
Объем дисциплины (модуля) (в зач. ед.)		5				

4. Содержание дисциплины (модуля)

№п/п	Наименование раздела дисциплины (модуля)	Содержание дисциплины (модуля) по темам
1	Машинное обучение в индустрии. Базовые инструменты инженера машинного обучения	Введение. Инструменты разработки.

2	Работа с данными в машинном обучении и трекинг экспериментов машинного обучения	Работа с данными в машинном обучении Трекинг экспериментов машинного обучения и воспроизводимость
3	Подготовка моделей машинного обучения к деплою	Деплой моделей машинного обучения. Часть 1. Деплой моделей машинного обучения. Часть 2. Индустриальный код/
4	Мониторинг	Мониторинг. Часть 1: стандартные инструменты Мониторинг. Часть 2: специфичные инструменты
5	Автоматизация обучения и деплоя моделей машинного обучения	Автоматизация пайплайна обучения моделей. Автоматизация сборки и развертывания для моделей машинного обучения.
6	Масштабирование и отказоустойчивость приложений машинного обучения	Масштабирование и отказоустойчивость приложений машинного обучения.
7	LLMOps	Индустриальное использование больших языковых моделей. ИИ Агенты в индустрии. Мониторинг и контроль использования больших языковых моделей.

5. Учебно-методическое обеспечение

Университет располагает полным набором лицензионного и свободно распространяемого программного обеспечения, включая продукты отечественного производства.

Каждый студент в течение всего периода обучения получает индивидуальный неограниченный доступ к электронно-библиотечной системе и электронной информационно-образовательной среде университета. Эти системы предоставляют возможность доступа к ресурсам из любой точки, где есть подключение к сети Интернет, как на территории университета, так и за его пределами.

Студентам обеспечен удаленный доступ к современным профессиональным базам данных и информационным справочным системам.

Основная литература:

1. Платонов, А. В. Машинное обучение : учебник для вузов / А. В. Платонов. — 2-е изд. — Москва : Издательство Юрайт, 2026. — 78 с. — (Высшее образование). — ISBN 978-5-534-22200-5. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/600893>.

2. Лакшманан, В. Машинное обучение. Паттерны проектирования : практическое пособие / В. Лакшманан, С. Робинсон, М. Мунн. - Санкт-Петербург : БХВ-Петербург, 2022. - 448 с. - ISBN 978-5-9775-6797-8. - Текст : электронный. - URL: <https://znanium.ru/catalog/product/2140204>.

Дополнительная литература:

1. Маркин, А. В. Программирование на SQL : учебник и практикум для вузов / А. В. Маркин. — 3-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2026. — 805 с. — (Высшее образование). — ISBN 978-5-534-18371-9. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/589589>.

6. Материально-техническое обеспечение

Университет располагает материально-технической базой, соответствующей действующим противопожарным правилам и нормам и обеспечивающей проведение всех видов дисциплинарной и междисциплинарной подготовки, практической и научно-исследовательской работ обучающихся, предусмотренных учебным планом.

Помещения, которые представляют собой учебные аудитории для проведения занятий лекционного типа, занятий семинарского (практического) типа, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации, а также помещения для самостоятельной работы и помещения для хранения и профилактического обслуживания учебного оборудования. Помещения укомплектованы специализированной мебелью и техническими средствами обучения, служащими для представления учебной информации большой аудитории.

Изучение дисциплины (модуля) обеспечивается в учебных аудиториях, оснащенных:

- столами и стульями;
- компьютерной техникой;
- механическими калькуляторами;

— специализированным оборудованием, включая демонстрационное оборудование.

Помещения для самостоятельной работы обучающихся, в том числе приспособленные для использования инвалидами и лицами с ограниченными возможностями здоровья, оснащены компьютерной техникой с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду Университета.

Обучающимся предоставляется доступ (в том числе удаленный) к ресурсам информационно-телекоммуникационной сети «Интернет», электронным ресурсам (в том числе электронным библиотечным системам, современным профессиональным базам данных и информационным справочным системам):

№	Наименование портала (издания, курса, документа)	Ссылка
1.	Научная электронная библиотека elibrary.ru библиотека	https://elibrary.ru/defaultx.asp
2.	База данных для IT-специалистов	https://habr.com
3.	База данных ScienceDirect	https://www.sciencedirect.com
4.	Официальный сайт Министерства науки и высшего образования Российской Федерации	https://minobrnauki.gov.ru/
5.	Федеральный портал «Российское образование»	https://www.edu.ru/
6.	Информационная система "Единое окно доступа к образовательным ресурсам"	http://window.edu.ru/
7.	Единая коллекция цифровых образовательных ресурсов	http://school-collection.edu.ru/
8.	Федеральный центр информационно - образовательных ресурсов	http://fcior.edu.ru/

Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), в том числе комплект лицензионного программного обеспечения, современные профессиональные базы данных и информационные справочные системы:

Наименование ПО	Производство	Лицензионное / свободно распространяемое
Операционные системы:		
Microsoft Imagine (Windows Client, Server)	зарубежное	лицензионное
Браузеры:		
Яндекс.Браузер	отечественное	свободно распространяемое
Google Chrome	зарубежное	свободно распространяемое
Офисные приложения:		
Microsoft Imagine (Visio, OneNote)	зарубежное	лицензионное
TeXstudio	зарубежное	свободно распространяемое
Adobe Acrobat Reader	зарубежное	свободно распространяемое
Программное обеспечение для планирования и учета времени:		
Toggle app	зарубежное	свободно распространяемое
Системы управления проектами:		
Microsoft Imagine (Project)	зарубежное	лицензионное

Системы управления базами данных:		
Microsoft Imagine (SQL Server)	зарубежное	лицензионное
Системы резервного копирования (backup):		
Acronis Backup Advanced for HyperV	зарубежное	лицензионное
Справочно-правовые системы:		
КонсультантПлюс: справочно-правовая система	отечественное	лицензионное
Средства антивирусной защиты:		
Kaspersky Endpoint Security для бизнеса Стандартный Russian Edition	отечественное	лицензионное
Среды разработки:		
Visual Studio Code	зарубежное	свободно распространяемое
Bash (Unix shell)	зарубежное	свободно распространяемое
Anaconda	зарубежное	свободно распространяемое
Robotic Operating System	зарубежное	свободно распространяемое
CopelliaSim	зарубежное	свободно распространяемое
Google Colaboratory	зарубежное	свободно распространяемое
Пакеты программных средств и библиотек:		
AutoPsy	зарубежное	свободно распространяемое
Interactive Disassembler (IDA)	зарубежное	свободно распространяемое
Системы управления библиографической информацией:		
Zotero	зарубежное	свободно распространяемое
Сервисы и службы:		
Bind	зарубежное	свободно распространяемое
Docker	зарубежное	свободно распространяемое

7. Методические и оценочные материалы

Методические указания для обучающихся по освоению дисциплины (модуля)

В процессе изучения дисциплины (модуля) «Эксплуатация и внедрение моделей машинного обучения (Production ML)» в рамках текущего контроля успеваемости используются такие виды учебной работы, как лекции, семинары, домашние задания, проекты, творческое задание, а также различные виды самостоятельной работы обучающихся по заданию преподавателя, направленные на развитие навыков профессиональной лексики, закрепление практических профессиональных компетенций, поощрение инициатив.

Лекция – систематическое, последовательное, монологическое изложение преподавателем учебного материала, как правило, теоретического характера.

В процессе лекций рекомендуется вести конспект лекций: кратко и схематично фиксировать основные идеи, выводы и обобщения лекции; выделять важные мысли, ключевые слова и термины. Необходимо отметить вопросы или материалы, которые вызывают затруднения, и попытаться найти ответы в рекомендованной литературе. Если разобраться в материале не удастся, следует сформулировать вопрос и задать его преподавателю на консультации или во время семинарского (практического) занятия.

Семинар – это форма учебной деятельности, проводимая в учебном заведении под руководством преподавателя, где студенты активно участвуют в обсуждениях, практических заданиях и других формах взаимодействия.

Для успешной подготовки к семинару рекомендуется заранее ознакомиться с темой занятия и основными материалами, чтобы иметь возможность активно участвовать в обсуждении. Также полезно подготовить вопросы и идеи для обсуждения, что поможет глубже понять материал и продемонстрировать заинтересованность.

Домашнее задание – набор задач по темам недели.

При работе над домашними заданиями важно внимательно ознакомиться с требованиями и сроками выполнения. Рекомендуется разбивать задания на этапы, чтобы избежать перегрузки и лучше усвоить материал. Использовать различные источники информации, включая учебники и онлайн-ресурсы, для более глубокого понимания темы.

Творческое задание — это форма учебной деятельности, где студенты самостоятельно разрабатывают оригинальные проекты, интегрируя математические методы и компьютерные технологии для решения нестандартных задач и демонстрации инновационного мышления.

Для успешного выполнения творческого задания рекомендуется глубоко исследовать тему, экспериментировать с различными подходами и документировать процесс, чтобы не только решить задачу, но и представить результаты в виде отчета или презентации, подчеркивающей креативность и практическую применимость.

Проект – исследовательская работа по дисциплине (модулю) и презентация результатов.

Для успешной подготовки к проекту: четко определите цели и задачи проекта, распределите роли и обязанности между участниками, а также установите сроки выполнения каждой части работы. Регулярно проводите встречи для обсуждения прогресса и решения возникающих вопросов.

Бонусные баллы — это оценки, которые студенты могут получить за выполнение дополнительных заданий.

Формат бонусных баллов позволяет студентам улучшить общую оценку по дисциплине (модулю) и стимулирует углубленное изучение материала.

Самостоятельная работа – работа студентов, направленная на углубленное изучение отдельных тем и вопросов учебной дисциплины (модуля).

В процессе самостоятельной работы студенты взаимодействуют с рекомендованными материалами при минимальном участии преподавателя. Задачи студента включают работу с конспектами лекций (обработка текста), повторное изучение учебных материалов планов и тезисов ответов, изучение дополнительных тем, выполнение учебно-исследовательских заданий и другое.

Система оценивания результатов обучения по дисциплине (модулю)

Критерии получения уровня и оценивания сформированности компетенций по дисциплине (модулю) «Эксплуатация и внедрение моделей машинного обучения (Production ML)»

Оценивание уровня учебных достижений обучающихся по дисциплине (модулю) осуществляется в виде текущего контроля успеваемости и промежуточной аттестации.

Промежуточная аттестация по дисциплине (модулю) осуществляется в форме *зачета с оценкой*, при этом проводится оценка компетенций, сформированных по дисциплине.

Для оценивания текущего контроля успеваемости и промежуточной аттестации используется десятибалльная шкала оценивания, которая соотносится с традиционной пятибалльной шкалой следующим образом, если иное не указано в syllabus дисциплины:

Десятибалльная оценка	Пятибалльная оценка	Общая характеристика результата обучения по дисциплине (модулю)
10	Отлично	Студент полностью владеет знаниями, изложенными в рабочей программе, и глубоко осмысляет дисциплину (модуль). Он самостоятельно и логически последовательно отвечает на все вопросы, акцентируя внимание на наиболее важном. Умеет анализировать, сравнивать, классифицировать, обобщать, конкретизировать и систематизировать изученный материал, выделяя ключевые моменты и устанавливая причинно-следственные связи. Четко формулирует ответы, уверенно интерпретирует результаты анализов и других исследований, а также решает сложные задачи. Студент хорошо знаком с методами исследования, необходимыми для практической деятельности, и умеет связывать теоретические аспекты дисциплины (модуля) с практическими задачами.
9	Отлично	
8	Отлично	
7	Хорошо	Студент обладает знаниями предмета почти в полном объеме рабочей программы и самостоятельно, логически последовательно и всесторонне отвечает на все вопросы, акцентируя внимание на наиболее значимых моментах. Он умеет анализировать, сравнивать, классифицировать, обобщать, конкретизировать и систематизировать изученный материал, выделяя его ключевые аспекты и устанавливая причинно-следственные связи. Формулирует свои ответы, уверенно интерпретирует результаты анализов и других исследований, а также решает сложные ситуационные задачи. Студент хорошо знаком с методами исследования, необходимыми для практической деятельности, и умеет связывать теоретические аспекты предмета с практическими задачами.
6	Хорошо	
5	Удовлетворительно	Студент обладает базовыми знаниями по дисциплине

Десятибалльная оценка	Пятибалльная оценка	Общая характеристика результата обучения по дисциплине (модулю)
4	Удовлетворительно	(модулю), но испытывает трудности при самостоятельных ответах и использует неточные формулировки. В ходе ответов он допускает ошибки, касающиеся сути вопросов. Студент способен решать только самые простые задачи и владеет лишь минимальным набором методов исследования.
3	Не сдан	Студент не овладел обязательным минимумом знаний по предмету и не может ответить на вопросы, даже если преподаватель задает дополнительные наводящие вопросы.
2	Не сдан	
1	Не сдан	

Дисциплина (модуль) «Эксплуатация и внедрение моделей машинного обучения (Production ML)» оценивается следующим образом:

Домашние задания	60%	13	Набор задач по темам недели
Проекты	30%	1	Исследовательская работа по дисциплине (модулю) и презентация результатов
Зачет с оценкой	10%	1	Творческое задание – форма учебной деятельности, где студентам предлагается разработать оригинальный проект или решить нестандартную задачу

В рамках изучения дисциплины (модуля) возможно получение бонусных баллов.

Формула расчёта итоговой оценки по дисциплине (модулю) «Эксплуатация и внедрение моделей машинного обучения (Production ML)»: $\langle 0,6 \times \text{среднее за домашние задания} + 0,3 \times \text{среднее за проекты} + 0,1 \times \text{зачет с оценкой} \rangle$.

Текущий контроль успеваемости обучающихся по дисциплине (модулю)

Примерные домашние задания

Домашнее задание 1.

Шаг 1. Поднять сервис Evidently AI (1 балл)

В файл `docker-compose.yml` добавить сервис `evidently` на порт 8502.

Шаг 2. Добавить создание репортов в `train.py` (6 баллов)

Написать функцию `log_evidently()`, выполняющую следующее:

- Проверяет наличие проекта в сервисе с названием `EV_PROJECT_NAME = "week-10"`
- Если проекта нет - создает его.
- Принимает `train_df`, `val_df`, создает из них `eval`'ы (`Dataset.from_pandas(...)`), описывает с помощью `DataDefinition`.

- Добавляет в данные признаки (с помощью дескрипторов - длина сообщений и наличие в них слов “bot” и “бот”)
- Создает Report с DataDriftPreset’ом
- Загружает 1 репорт в сервис (при каждом запуске train.py - новый репорт).

Шаг 3. Добавить дэшборд по репортам (3 балла)

Создать дэшборд типа “Line” и size=”full”, визуализирующий Drift Score колонки “text” из репортов (подумать, на каком шаге он должен быть создан).

Как выполнить и сдать задание

В этом задании вы не сдаёте ничего в ЛМС, а все проверки вашего домашнего задания осуществляются через ваш репозиторий.

Что необходимо для сдачи задания:

- Используйте для работы и запуска проверки репозиторий из self-service:

Структура репозитория:

- Обновите docker-compose.y(a)ml, добавив сервис evidently
- Обновите файл train.py
- Убедитесь, что все необходимые и **прошлые** переменные окружения (секреты) добавлены в Gitlab CI/CD Variables:

CI / CD Variables (добавить в Settings → CI/CD → Variables)

Переменная	Значение
EVIDENTLY_PORT	8502
EVIDENTLY_HOST	evidently

Домашнее задание 2.

Шаг 1 (0 балл). Подготовка инфраструктуры

1. На вашей виртуальной машине должны быть установлены Docker и Docker Compose
2. Выберите инструмент для хостинга LLM, доступный в форме Docker-контейнера (например, llama.cpp, text-generation-inference, vLLM и т. д.).
3. Убедитесь, что у вашей ВМ достаточно ресурсов (оперативной памяти и CPU) для выбранной модели, иначе подберите модель поменьше.
4. С помощью docker pull загрузите контейнер инструмента на ВМ.

Шаг 2 (5 баллов). Запуск модели

1. **Выбор модели:** выберите LLM (весом и объёмом, подходящим для вашей VM). Загрузите файл модели, поместив его либо на саму VM, либо в S3-minio хранилище.

2. **Соберите или возьмите готовый Docker-образ** с выбранным инструментом и запустите модель внутри контейнера, далее поднимите сервис через docker-compose:

- **Обеспечьте**, чтобы модель принимала запросы по OpenAI-like протоколу (см. документацию OpenAI). Обычно инструменты LLM (TGI, vLLM, llama.cpp и т. д.) предоставляют эндпоинты совместимые с POST /v1/chat/completions.

- **Ограничьте** использование CPU и RAM, чтобы исключить перегрузку вашей VM. В docker-compose.yaml это можно сделать, задав ресурсы контейнера, используя флаг --compatibility при запуске docker-compose.

(можете задавать ограничения на количество CPU и памяти — на ваше усмотрение, главное, чтобы VM не упала).

Шаг 3 (2 балла). API-key защита

1. Настройка

- Модель должна быть доступна по порту PORT=38300.
- Установите переменную окружения API_KEY=cu_prodml_hw_week13. При обращении к API с неправильным ключом доступ должен блокироваться: "Authorization: Bearer \$OPENAI_API_KEY"
- Убедитесь, что при запуске контейнера ваш сервис слушает **localhost:38300**.

1. Проверка

- Попробуйте curl-запрос вида ниже и проверьте, что получаете валидный JSON-ответ, содержащий сгенерированный текст.

```
curl http://localhost:38300/v1/chat/completions \
-H "Authorization: Bearer cu_prodml_hw_week13" \
-H "Content-Type: application/json" \
-d '{
  "model": "model_identifier",
  "messages": [{"role": "user", "content": "Hello!"}] }'
```

Если у вас сервис по умолчанию не слушает /v1, попробуйте без этого суффикса (всё зависит от конкретного фреймворка).

Важно: на шаге 3 мы будем подключать вашу локальную LLM к вашему же боту, так что формат и порт здесь должны оставаться неизменными, чтобы грейдер мог это протестировать.

Шаг 4 (3 балла). Апгрейд бота

1. В предыдущих заданиях вы работали с сервисом классификации сообщений. Теперь необходимо использовать сервис генерации сообщений из репозитория, далее именуемый как бот. Его нужно обновить так, чтобы вместо обращения к OpenAI ваш бот ходил к **локальной модели** на порту **38300**.

- Для этого в коде бота нужно понять, как изменить переменные конфигурации OPEN_AI_API_KEY, PROXY_URL, API_PORT, а также какой код добавить, чтобы использовался ваш сервис, например `http://gpt_service:38300` (для этого всё должно быть в одной docker-compose-сети).

- Используйте тот же ключ Bearer `cu_prodml_hw_week13` при обращении.

1. **Убедитесь**, что ваш бот успешно получает ответы из локальной LLM. Возможно, придётся подстроить JSON-схему ответа.

2. README.md:

- Опишите, какую модель выбрали, какой фреймворк для её развёртывания, почему, и как её задеплоили (docker-compose, ограничение CPU, переменные окружения).
- Укажите точные шаги запуска (если это что-то отличное от стандартного docker-compose up -d).
- Опишите, как ваш бот обращается к локальной LLM через API.

Как выполнить и сдать задание

В этом задании вы не сдаёте ничего в ЛМС, а все проверки вашего домашнего задания осуществляются через ваш репозиторий. Для корректной работы автоматической проверки используйте **тот же самый** репозиторий, который вы создавали в первом домашнем задании через self-service.

Что необходимо для сдачи задания:

- **Структура репозитория:**
 - В docker-compose.yml добавьте сервис, запускающий LLM
 - Убедитесь, что порт **38300** для LLM не конфликтует с другими сервисами (такими как MLflow, Minio и др.).
 - В коде вашего бота замените обращения к OpenAI на вызов `http://gpt_service:38300` с нужными заголовками.
 - Ответ сервиса будет проверяться на соответствие структуре ответов OpenAI — ваш сервис должен отвечать в нужном формате.
- **Переменные CI/CD:**
 - Оставьте все ранее добавленные переменные.
 - Добавьте переменную GPT_SERVICE_PORT — порт на котором находится ваш бот.
 - Если решите использовать другой ключ, отличный от `cu_prodml_hw_week13`, то обязательно добавьте переменную API_KEY.

- Задайте новый путь до проекта, если он изменился.

Домашнее задание 3.

Шаг 1. Практика по kubectl (10 баллов)

Подготовка:

1. Изучите пример манифеста manifest.yaml из материалов занятия
2. Поднимите локально или на виртуалке minikube и примените этот манифест

```
kubectl apply -f manifest.yaml
```

В своём репозитории подготовьте файл kuber_practice.py (в корне репозитория) и пропишите в нём **по одной команде kubectl** в соответствующую переменную (без |, &&, || и т. д.), **решая следующие пункты:**

1. Выведите все поды в неймспейсе my-namespace для деплоймента app1.
2. Выведите все поды **во всех** неймспейсах для деплоймента app1.
3. Выведите информацию о деплойменте app3 в неймспейсе my-namespace.
4. Выведите логи деплоймента app2 в неймспейсе my-namespace.
5. Увеличьте число реплик у деплоймента app1 в неймспейсе my-namespace до 4.
6. Верните обратно число реплик у деплоймента app1 в неймспейсе my-namespace до 1.
7. Удалите все поды у деплоймента app1 в неймспейсе my-namespace.
8. Сделайте деплоймент app1 доступным извне, чтобы сервис отвечал по порту **33888**.

Примерное описание задания и критерии оценивания к проектам

Проект: Внедрение модели классификации в API

Описание:

Разработка и деплой модели бинарной классификации (например, спам/не спам) с предоставлением доступа через REST API.

Задание:

1. Выбрать датасет (например, SMS Spam Collection).
2. Обучить модель (логистическая регрессия, Random Forest и т.п.).
3. Сохранить модель (pickle или joblib).
4. Создать API с FastAPI или Flask.
5. Упаковать приложение в Docker-контейнер.
6. Настроить логирование запросов и ответов.
7. Протестировать API с помощью curl или Postman.
8. Подготовить README с инструкцией по запуску.

Критерии оценивания:

- Реализовано обучение и сохранение модели
- API работает, принимает JSON, возвращает предсказание

- Контейнеризация через Docker
- Есть логирование и обработка ошибок
- Документация и воспроизводимость
- Тестирование и демонстрация

Проект: Автоматизация ML-пайплайна с MLflow и Airflow

Описание:

Создание автоматизированного пайплайна обучения модели с трекингом экспериментов и оркестрацией.

Задание:

1. Использовать MLflow для логирования: параметров, метрик, модели.
2. Написать скрипты: загрузка данных → предобработка → обучение → валидация.
3. Оркестрировать выполнение с помощью Apache Airflow (DAG).
4. Настроить запуск по расписанию (например, ежедневно).
5. Добавить проверку качества данных (на пропуски, типы).
6. Сохранить лучшую модель в реестре MLflow.
7. Подготовить отчёт о лучших экспериментах.

Критерии оценивания:

- Использование MLflow для трекинга и реестра
- Работающий DAG в Airflow
- Проверка данных и обработка ошибок
- Автоматизация и расписание
- Отчёт и документация

Проект: Мониторинг дрейфа данных в ML-системе

Описание:

Разработка системы обнаружения дрейфа данных после деплоя модели.

Задание:

1. Обучить модель на тренировочных данных.
2. Сэмулировать "продовые" данные с изменённым распределением.
3. Реализовать проверку дрейфа (с помощью Evidently, NannyML или вручную — сравнение статистик).
4. Настроить уведомление (логирование или print-сообщение при дрейфе).
5. Визуализировать распределения (до/после).
6. Интегрировать проверку в инференс-пайплайн.
7. Документировать пороги срабатывания.

Критерии оценивания:

- Корректная реализация детекции дрейфа
- Визуализация и интерпретация
- Интеграция в пайплайн
- Настройка уведомлений
- Документация и воспроизводимость

Примерное описание задания и критерии оценивания к творческому заданию

Проектирование системы на основе LLM для автоматизации клиентской поддержки

Описание задания:

Студент разрабатывает концепцию системы, использующей большую языковую модель (например, Llama 3, GPT) для обработки обращений клиентов.

Задание:

1. Опишите бизнес-задачу: автоматизация ответов на частые вопросы.
2. Выберите архитектуру: RAG (Retrieval-Augmented Generation) или fine-tuning.
3. Определите источник знаний (база FAQ, документы).
4. Спроектируйте схему: пользователь → чат-бот → RAG → LLM → ответ.
5. Укажите меры контроля: фильтрация токсичности, проверка на PII, ограничение длины.
6. Опишите мониторинг: логирование, оценка качества ответов, метрики задержки.
7. Предложите способ масштабирования (например, Kubernetes).
8. Оформите в виде схемы и текстового описания (1–2 стр.).

Критерии оценивания:

- Чёткость и актуальность задачи
- Обоснованность выбора архитектуры
- Учёт безопасности и контроля
- Продуманность мониторинга и масштабирования
- Качество схемы и оформления

Задания для промежуточной аттестации по дисциплине (модулю)

№ п/п	Задание	Ответ	Компетенция
1.	Как называется свойство эксперимента машинного обучения, при котором другой инженер может получить тот же результат на тех же данных и коде?	Воспроизводимость	ОПК-1
2.	Как называется ухудшение качества модели из-за изменения свойств входных данных после деплоя?	Дрифт данных/data drift/сдвиг данных	ОПК-1
3.	Как называется подход, при котором модель разворачивают в изолированной среде вместе со всеми зависимостями?	Контейнеризация/контейнер	ОПК-1
4.	Как называется характеристика сервиса, показывающая его способность продолжать работу при сбоях отдельных компонентов?	Отказоустойчивость	ОПК-1
5.	Как называется процесс автоматического запуска шагов от подготовки данных до обучения модели по заранее заданному сценарию?	Оркестрация пайплайна/оркестрация	ОПК-1
6.	Как называется показатель, который измеряет время ответа сервиса модели на запрос?	Latency/задержка	ОПК-1
7.	Как называется журналирование параметров, метрик и артефактов эксперимента в машинном обучении?	Трекинг экспериментов/отслеживание	ОПК-1

		экспериментов	
8.	Как называется процесс переноса обученной модели в рабочую среду для получения предсказаний?	Деплой/развертывание	ПК-1
9.	Как называется среда, в которой модель обслуживает реальные пользовательские запросы?	Продакшен/production	ПК-1
10.	Как называется проверка сервиса модели после изменений кода, которая подтверждает, что прежняя функциональность не нарушена?	Регрессионное тестирование	ПК-1
11.	Как называется практика автоматической сборки, тестирования и доставки изменений в сервис модели?	CI/CD/непрерывная интеграция и доставка	ПК-1
12.	Как называется версия данных, модели и кода, зафиксированная для повторного запуска эксперимента или восстановления результата?	Версионирование/версия артефактов	ПК-1
13.	Как называется способ развертывания, при котором новая версия модели получает только часть трафика для проверки?	Канареечный деплой/canary deployment/канареечное развертывание	ПК-1
14.	Как называется архитектурный подход, при котором функции приложения разделяются на независимые сервисы?	Микросервисная архитектура/микросервисы	ПК-1
15.	Как называется файл, в котором описывают шаги сборки контейнера для сервиса модели?	Dockerfile	ПК-6
16.	Как называется программный интерфейс, через который внешняя система отправляет данные в модель и получает предсказание?	API/api	ПК-6
17.	Как называется библиотека или инструмент, который часто используют для организации ML-экспериментов и хранения метрик, параметров и артефактов?	MLflow/mlflow	ПК-6
18.	Как называется автоматическая проверка работоспособности отдельных функций кода сервиса модели?	Модульное тестирование/unit testing	ПК-6
19.	Как называется инструмент управления контейнерами, который часто используют для масштабирования сервисов машинного обучения?	Kubernetes/k8s	ПК-6
20.	Как называется подход к работе с большими языковыми моделями в промышленной среде, включающий деплой, мониторинг и контроль их использования?	LLMOps	ПК-6