ПРИЛОЖЕНИЕ

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«РЯЗАНСКИЙ ГОСУДАРСТВЕННЫЙ РАДИОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ИМЕНИ
В.Ф. УТКИНА»

Кафедра «Систем автоматизированного проектирования вычислительных средств»

ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ПО ДИСЦИПЛИНЕ

Б1.В.09 «Проектирование и поддержка систем искусственного интеллекта»

Направление подготовки
02.03.02 Фундаментальная информатика и информационные технологии

Направленность (профиль) подготовки
««Искусственный интеллект и информационные технологии»

Квалификация выпускника – бакалавр

Форма обучения – очная

Рязань 2025

Фонд оценочных средств – это совокупность учебно-методических материалов (контрольных заданий, описаний форм и процедур), предназначенных для оценки качества освоения обучающимися данной дисциплины как части основной образовательной программы.

Цель – оценить соответствие знаний, умений и уровня приобретенных компетенций, обучающихся целям и требованиям основной образовательной программы в ходе проведения текущего контроля и промежуточной аттестации.

Основная задача – обеспечить оценку уровня сформированности общекультурных и профессиональных компетенций, приобретаемых обучающимся в соответствии с этими требованиями.

Контроль знаний обучающихся проводится в форме текущего контроля и промежуточной аттестации.

Текущий контроль успеваемости проводится с целью определения степени усвоения учебного материала, своевременного выявления и устранения недостатков в подготовке обучающихся и принятия необходимых мер по совершенствованию методики преподавания учебной дисциплины (модуля), организации работы обучающихся в ходе учебных занятий и оказания им индивидуальной помощи.

К контролю текущей успеваемости относятся проверка знаний, умений и навыков обучающихся: по результатам выполнения обучающимися индивидуальных заданий в ходе выполнения практических занятий.

При оценивании (определении) результатов освоения дисциплины применяется традиционная шкала оценки (отлично, хорошо, удовлетворительно, неудовлетворительно).

В качестве оценочных средств на протяжении семестра используются лабораторные работы.

Результат выполнения каждой лабораторной работы оценивается как "зачет" в случае выполнения обучающимся всех индивидуальных заданий.

Результат выполнения каждого индивидуального задания должен соответствовать всем критериям оценки в соответствии с компетенциями, установленными для заданного раздела дисциплины.

По итогам курса обучающиеся сдают экзамен. Форма проведения экзамена – устный ответ, по утвержденным экзаменационным билетам, сформулированным с учетом содержания учебной дисциплины. В экзаменационный билет включается два теоретических вопроса по темам курса.

Паспорт фонда оценочных средств по дисциплине

№ п/п	Контролируемые разделы (темы) дисциплины	Код контролируемой компетенции (или её части)	Вид, метод, форма оценочного мероприятия
1	Введение в ИИ-системы	ПК-23.1-3	текущий контроль
2	Анализ требований и формализация задачи для ИИ-системы	ПК-6.1	текущий контроль, выполнение ЛР
3	Сбор и предобработка данных	ПК-6.1	текущий контроль,

			выполнение ЛР
4	Выбор и сравнение архитектур нейронных сетей	ПК-6.1	текущий контроль, выполнение ЛР
5	Разработка и оптимизация моделей в фреймворках	ПК-6.2	текущий контроль, выполнение ЛР
6	Технологии обработки больших данных для ИИ	ПК-9.1	текущий контроль, выполнение ЛР
7	Проектирование архитектуры ИИ-системы	ПК-6.1 ПК-9.2 ПК-23.1	текущий контроль, выполнение ЛР
8	Обучение и валидация моделей	ПК-6.1-3 ПК-6.2-3	текущий контроль
9	Прикладные решения с элементами ИИ	ПК-9.2	текущий контроль, выполнение ЛР
10	Инфраструктура данных для ИИ-проектов	ПК-10.1-3 ПК-10.1-У	текущий контроль
11	Управление версиями и CI/CD для ML	ПК-25.1	текущий контроль, выполнение ЛР
12	Мониторинг и логирование в продакшене	ПК-24.2 ПК-25.1	текущий контроль, выполнение ЛР
13	Безопасность и приватность в ИИ-системах	ПК-25.1	текущий контроль, выполнение ЛР
14	Управление ИИ-проектами	ПК-24.1-3 ПК-24.2-3	текущий контроль
15	Инструменты промышленной разработки ИИ	ПК-24.1 ПК-25.1	текущий контроль, выполнение ЛР
16	Кейсы и антипаттерны	ПК-23.1-3 ПК-24.1-3 ПК-24.2-3	текущий контроль
17	Промежуточная аттестация	ПК-6.1-3 ПК-6.2-3 ПК-9.1-3 ПК-9.2-3 ПК-10.1-3 ПК-10.2-3 ПК-23.1-3 ПК-24.1-3 ПК-24.2-3 ПК-25.1-3	сдача экзамена

Показатели и критерии обобщенных результатов обучения

Результаты обучения по дисциплине	Показатели оценки результата	Критерии оценки результата
ПК-6.1. Осуществляет оценку и выбор моделей искусственных нейронных сетей и инструментальных средств для	•Знание архитектур нейронных сетей;	*Удовлетворительно*: знает основные архитектуры и фреймворки, но испытывает затруднения при сопоставлении задачи с моделью;

решения поставленной задачи **Знать** • основные архитектуры нейронных сетей (CNN, RNN, LSTM, Transformers); • критерии выбора моделей (точность, latency, ресурсоёмкость, интерпретируемость); • популярные фреймворки (TensorFlow, PyTorch, Keras). **Уметь** • сопоставлять задачи с подходящими архитектурами сетей; • оценивать адекватность модели под конкретные требования; • проводить бенчмаркинг моделей на тестовых данных. **Владеть** • методикой сравнительного анализа моделей; • инструментами оценки производительности (Latency, Throughput);	• Понимание критериев выбора моделей; • Владение фреймворками; • Умение проводить бенчмаркинг моделей; • Навыки сравнительного анализа и оценки производительности.	проводит базовый бенчмаркинг без глубокого анализа метрик. *Хорошо*: уверенно сопоставляет задачи с архитектурами, оценивает адекватность модели по ключевым критериям, проводит бенчмаркинг с интерпретацией результатов. *Отлично*: комплексно анализирует задачи, учитывает дополнительные факторы (масштабируемость, интерпретируемость), проводит детальный бенчмаркинг с обоснованным выбором модели.
ПК-6.2. Разрабатывает системы искусственного интеллекта на основе моделей искусственных нейронных сетей и инструментальных средств **Знать** • этапы разработки ИИ-системы от прототипа до продакшена; • принципы модульного проектирования ML-пайплайнов; • паттерны интеграции моделей в приложения. **Уметь** • реализовывать модели в выбранных фреймворках; • проектировать API для инференса; • отлаживать код с учётом специфики ML. **Владеть** • навыками кодирования ML-моделей; • техниками оптимизации инференса (квантование,	• Знание этапов разработки ИИ-системы (от прототипа до продакшена); • Понимание принципов модульного проектирования ML-пайплайнов; • Умение реализовывать модели в фреймворках и проектировать API для инференса; • Навыки оптимизации инференса (квантование, прунинг); • Владение	*Удовлетворительно*: реализует модели в фреймворках, создаёт базовый API, но не оптимизирует инференс и не учитывает масштабируемость. *Хорошо*: проектирует ML-пайплайны, оптимизирует инференс, контейнеризирует решения, учитывает требования к производительности. *Отлично*: разрабатывает масштабируемые модели с модульной архитектурой, применяет продвинутые техники оптимизации, обеспечивает готовность к продакшену.

прунинг); • инструментами контейнеризации (Docker) для развёртывания.	инструментами контейнеризации.	
ПК-9.1. Осуществляет выбор технологий обработки больших данных, приемлемых для создания прикладной системы ИИ с заданными требованиями **Знать** • экосистему инструментов для Big Data (Hadoop, Spark, Dask); • сценарии применения распределённых вычислений; • принципы масштабирования предобработки данных; **Уметь** • выбирать технологию под объём и тип данных; • проектировать ETL-пайплайны для ML; • интегрировать Big Data-инструменты с ML-фреймворками. **Владеть** • базовыми операциями в Spark/Dask; • методами оптимизации загрузки данных; • основами конфигурирования кластеров для обработки данных.	• Знание экосистемы Big Data (Hadoop, Spark, Dask); • Понимание сценариев распределённых вычислений; • Умение проектировать ETL-пайплайны для ML; • Навыки работы с Spark/Dask и конфигурирования кластеров.	*Удовлетворительно*: знает инструменты Big Data, но выбирает технологии без глубокого анализа требований; проектирует простые ETL-пайплайны. *Хорошо*: обоснованно выбирает технологии под объём и тип данных, проектирует ETL-пайплайны с учётом масштабируемости, выполняет базовые операции в Spark/Dask. *Отлично*: комплексно оценивает требования, проектирует оптимизированные ETL-пайплайны, конфигурирует кластеры для максимальной эффективности.
ПК-9.2. Разрабатывает и отлаживает прикладные решения с элементами ИИ с применением различных технологий обработки данных **Знать** • типовые архитектуры ИИ-приложений (микросервисы, serverless); • протоколы взаимодействия компонентов (REST, gRPC); • методы тестирования ML-компонентов. **Уметь** • интегрировать модели в бизнес-логику приложения; • разрабатывать API для инференса;	• Знание типовых архитектур ИИ-приложений (микросервисы, serverless); • Понимание протоколов взаимодействия (REST, gRPC); • Умение интегрировать модели в бизнес-логику; • Навыки развёртывания (FastAPI, Flask)	*Удовлетворительно*: интегрирует модели в простые приложения, создаёт базовый API, но не настраивает мониторинг и не учитывает отказоустойчивость. *Хорошо*: разрабатывает масштабируемые ИИ-приложения, настраивает мониторинг инференс-запросов, отлаживает распределённые системы. *Отлично*: проектирует высоконагруженные системы с микросервисной архитектурой, обеспечивает отказоустойчивость, применяет продвинутые методы отладки.

• настраивать мониторинг инференс-запросов. **Владеть** • инструментами развёртывания (FastAPI, Flask); • техниками контейнеризации ML-сервисов; • методами отладки распределённых систем.	и отладки распределённых систем.	
ПК-10.1. Осуществляет выбор направления вспомогательных технологических решений для формирования единого стека работы с большими данными для решения поставленной задачи **Знать** • типы СУБД для разных сценариев (SQL, NoSQL, векторные хранилища); • инструменты DataOps (Airflow, NiFi); • принципы гибридных архитектур данных. **Уметь** • подбирать стек технологий под требования проекта; • проектировать схемы хранения данных; • настраивать ETL-процессы. **Владеть** • навыками работы с PostgreSQL/MongoDB; • основами конфигурирования Data Pipeline; • методами оценки производительности хранилищ.	• Знание типов СУБД (SQL, NoSQL, векторные хранилища); • Понимание инструментов DataOps (Airflow, NiFi); • Умение подбирать стек технологий и проектировать схемы хранения; • Навыки работы с PostgreSQL/ MongoDB и конфигурирования Data Pipeline.	*Удовлетворительно*: знает типы СУБД, но выбирает стек технологий без учёта долгосрочных требований; проектирует базовые схемы хранения. *Хорошо*: обоснованно подбирает стек под требования проекта, проектирует схемы хранения с учётом масштабируемости, настраивает ETL-процессы. *Отлично*: проектирует гибридные архитектуры данных, оптимизирует производительность хранилищ, интегрирует DataOps-инструменты в пайплайны.
ПК-10.2. Разрабатывает и отлаживает прикладные решения с элементами ИИ с применением различных технологий организации инфраструктуры БД **Знать** • модели данных для ИИ-систем (реляционные, документоориентированные); • принципы индексации и шардинга; • способы интеграции БД с	• Знание моделей данных для ИИ-систем (реляционные, документоориентированные); • Понимание принципов индексации и шардинга; • Умение проектировать	*Удовлетворительно***:** проектирует простые схемы БД, выполняет базовые запросы, но не оптимизирует нагрузки и не обеспечивает консистентность. *Хорошо***:** проектирует схемы с учётом ML-нагрузок, оптимизирует запросы, настраивает кэширование и репликацию. *Отлично***:** проектирует высокопроизводительные хранилища с шардингом и индексацией, обеспечивает консистентность

ML-моделями. **Уметь** • проектировать схемы БД для хранения признаков и предсказаний; • оптимизировать запросы для ML-нагрузок; • обеспечивать консистентность данных. **Владеть** • SQL/NoSQL запросами для работы с данными моделей; • техниками кэширования результатов инференса; • навыками настройки репликации данных.	схемы БД для хранения признаков и предсказаний; • Навыки SQL/NoSQL запросов и настройки репликации.	данных в распределённых системах.
ПК-23.1. Создает и развивает архитектуры системы ИИ на всех этапах жизненного цикла **Знать** • фазы жизненного цикла ИИ-системы; • шаблоны проектирования (event-driven, микросервисы); • принципы эволюционного развития архитектуры. **Уметь** • разрабатывать архитектурные схемы с учётом масштабирования; • учитывать требования к отказоустойчивости; • документировать архитектурные решения. **Владеть** • инструментами визуализации архитектуры (UML, C4); • методиками рефакторинга ML-пайплайнов; • навыками оценки технического долга.	• Знание фаз жизненного цикла ИИ-системы и шаблонов проектирования • Умение разрабатывать архитектурные схемы с учётом масштабирования; • Навыки визуализации архитектуры (UML, C4) и оценки технического долга.	*Удовлетворительно*: разрабатывает простые архитектурные схемы без учёта масштабирования, использует базовые инструменты визуализации. *Хорошо*: проектирует масштабируемые архитектуры с учётом отказоустойчивости, документирует решения, оценивает технический долг. *Отлично*: создаёт эволюционные архитектуры с event-driven и микросервисными шаблонами, проводит рефакторинг пайплайнов, обеспечивает долгосрочную поддержку.
ПК-24.1. Осуществляет запуск и ведение проекта в области ИИ, в том числе планирование и контроль задач, оценку ресурсов **Знать** • методологии управления ML-проектами (Agile, Scrum); • этапы проектного цикла (планирование, исполнение,	• Знание методологий управления (Agile, Scrum) и этапов проектного цикла; • Умение составлять	*Удовлетворительно*: планирует задачи по шаблону, контролирует сроки без учёта рисков, использует инструменты менеджмента на базовом уровне. *Хорошо*: составляет детализированные дорожные карты, распределяет задачи с учётом компетенций, контролирует бюджет и

мониторинг); • методы оценки ресурсов (GPU/TPU, облачные затраты). **Уметь** • составлять дорожные карты ИИ-проектов; • распределять задачи в команде; • контролировать сроки и бюджет. **Владеть** • инструментами проектного менеджмента (СФЕРА.Задачи); • техниками оценки рисков; • навыками коммуникации с заказчиками.	дорожные карты и распределять задачи; • Навыки работы с инструментами проектного менеджмента (СФЕРА.Задачи).	сроки. *Отлично*: управляет проектами с адаптацией методологий, оценивает риски и ресурсы (GPU/TPU, облачные затраты), эффективно взаимодействует с командой.
ПК-24.2. Контролирует реализацию проекта в соответствии с разработанной архитектурой проекта **Знать** • стандарты документирования архитектурных решений; • метрики соответствия реализации проекту; • методы аудита кода и пайплайнов. **Уметь** • сверять реализацию с архитектурной схемой; • выявлять отклонения от проекта; • инициировать корректирующие действия. **Владеть** • инструментами статического анализа кода; • методиками code review для ML-проектов; • техниками контроля версий (Git, DVC).	• Знание стандартов документирования и метрик соответствия; • Умение сверять реализацию с архитектурной схемой; • Навыки статического анализа кода и code review для ML-проектов.	*Удовлетворительно*: проводит поверхностный аудит, выявляет очевидные отклонения, использует базовые инструменты контроля версий. *Хорошо*: систематически сверяет реализацию с архитектурой, инициирует корректирующие действия, проводит code review. *Отлично*: применяет комплексные методы аудита (статический анализ, мониторинг метрик), обеспечивает полное соответствие проекта документации, управляет техническим долгом.
ПК-25.1. Осуществляет выбор инструментов и инженерных практик промышленной разработки систем ИИ, развертывания и сопровождения моделей машинного обучения в продуктивной среде **Знать** • инструменты MLOps (MLflow, Kubeflow, Seldon Core);	• Знание инструментов MLOps (MLflow, Kubeflow, Seldon Core); • Понимание практик CI/CD для ML-пайплайнов	*Удовлетворительно*: использует базовые инструменты MLOps, настраивает трекинг экспериментов, но не автоматизирует пайплайны. *Хорошо*: внедряет CI/CD для ML, автоматизирует ретрейнинг, мониторит продуктовые модели, обеспечивает базовую отказоустойчивость. *Отлично*: создаёт полностью

• практики CI/CD для ML-пайплайнов; • стандарты безопасности при развёртывании моделей. **Уметь** • настраивать трекинг экспериментов; • автоматизировать ретрейнинг и деплой; • внедрять мониторинг продуктовых моделей. **Владеть** • навыками работы с MLflow/Weights & Biases; • техниками автоматизации ML-пайплайнов; • методами обеспечения отказоустойчивости инференса.	• Умение настраивать трекинг экспериментов и автоматизировать деплой; • Навыки работы с MLflow/Weights & Biases и обеспечения отказоустойчивости.	автоматизированные MLOps-пайплайны, применяет продвинутые практики безопасности, обеспечивает высокую доступность инференса.

Типовые контрольные задания или иные материалы

Вопросы для подготовки к защите лабораторных работ по дисциплине

1). Какие ключевые критерии учитываются при выборе архитектуры нейронной сети?

2). В чем заключается сходство и различие между архитектурами нейронных сетей CNN, RNN, LSTM, Transformer? Назовите область их применимости.

3). Как оцениваются latency и ресурсоёмкость модели? Какие инструменты используются для этого?

4). Опишите процесс бенчмаркинга моделей на ваших тестовых данных. Какие метрики считаются ключевыми и почему?

5). В чём преимущества и недостатки фреймворков TensorFlow, PyTorch, Keras? Для каких задач они используются?

6). Перечислите основные этапы разработки ИИ-системы от прототипа до продакшена.

7). Как проектируется API для инференса? Приведите пример эндпоинта и его параметров.

8). Какие техники оптимизации инференса (квантование, прунинг) применяются при проектировании? Каков ожидаемый эффект?

9). Как контейнеризируется прототип системы в Docker? Приведите ключевые строки из Dockerfile.

10). Как происходит отладка кода с учётом специфики ML? Назовите 2–3 типичные ошибки и способы их исправления.

11). В каких случаях выбирается Hadoop, Spark и Dask? Сравните функциональность этих инструментов.

12). Опишите архитектуру ETL-пайплайна для вашего ML-проекта. Какие этапы в него включаются?

13). Как масштабируется предобработка данных? Приведите пример оптимизации.

14). Какие операции выполняются в Spark/ Dask? Приведите фрагмент кода.

15). Как конфигурируется кластер для обработки данных? Какие параметры подлежат настройке?

16). Какая архитектура ИИ-приложения (микросервисы, serverless) выбирается и в каких случаях?

17). Как интегрируется ML-модель в бизнес-логику? Приведите схему взаимодействия.

18). Какие протоколы (REST, gRPC) используются для API? В чём их плюсы/минусы для вашей задачи?

19). Как настраивается мониторинг инференс-запросов? Какие метрики отслеживаются?

20). Какие инструменты (FastAPI, Flask) используются для развёртывания? В чем их отличие?

21). Какой тип СУБД (SQL, NoSQL, векторная) выбирается для хранения данных и в каких случаях этот тип является оптимальным?

22). Приведите ER-диаграмму схемы хранения данных.

23). Какие инструменты DataOps (Airflow, NiFi) применяются и для каких задач?

24). Как настраиваются ETL-процессы? Приведите пример DAG (или описания шагов).

25). Как оценивается производительность выбранного хранилища? Какие тесты при этом проводятся?

26). Какая модель данных (реляционная, документоориентированная) используется для проектирования вашей системы ИИ? В чем их преимущества и недостатки?

27). Как вы проектировали индексы для ускорения запросов? Приведите пример SQL/ NoSQL запроса.

28). Как обеспечивается консистентность данных при обновлении модели?

29). Какие техники кэширования результатов инференса применяются? Какой они могут дать выигрыш в производительности?

30). Как настраивается репликация данных? Опишите топологию вашей сети.

31). Какие шаблоны проектирования (event-driven, микросервисы) вы использовали в архитектуре? Приведите схему.

32). Как автоматизируется CI/CD для ML-пайплайна? Опишите шаги деплоя и мониторинга.

Список вопросов для подготовки к экзамену

1. Введение в ИИ-системы: понятия, жизненный цикл, роли в команде. Определение ИИ-системы. Классификация по задачам и архитектуре.
2. Фазы жизненного цикла: от анализа требований до сопровождения. Роли участников (ML-инженер, data scientist, DevOps) и зоны ответственности.
3. Анализ требований и формализация задачи для ИИ-системы. Выявление бизнес-цели и метрик успеха (KPI, ROI).
4. Типовые задачи ИИ (классификация, регрессия, генерация) и критерии выбора. Ограничения: данные, вычислительные ресурсы, этика, регуляторика.
5. Сбор и предобработка данных: инструменты и практики. Источники данных; разметка и аннотирование (инструменты: Label Studio, CVAT).
6. Очистка, нормализация, обработка пропусков и аномалий. Аугментация и балансировка данных для обучения моделей.
7. Выбор и сравнение архитектур нейронных сетей. Обзор моделей: CNN (компьютерное зрение), RNN/LSTM (временные ряды), Transformers (NLP). Критерии выбора: точность, latency, интерпретируемость, ресурсоёмкость.
8. Разработка и оптимизация моделей в фреймворках. Реализация в TensorFlow/Keras и PyTorch: от прототипа к продакшен-коду. Техники оптимизации: квантование, прунинг, дистилляция.
9. Технологии обработки больших данных для ИИ. Сравнение Hadoop, Spark, Dask: сценарии применения.
10. Интеграция с ML-пайплайнами (Spark MLlib). Масштабирование предобработки данных для обучения моделей.
11. Проектирование архитектуры ИИ-системы. Шаблоны: микросервисы, серверная/edge-инференция, event-driven.
12. Модули: ingestion, preprocessing, модель, API, мониторинг. Интеграция с legacy-системами и СУБД.
13. Обучение и валидация моделей: практики и метрики. Разбиение данных (train/val/test); кросс-валидация. Метрики: accuracy, F1, ROC-AUC, MAE, IoU (для сегментации).
14. Борьба с переобучением: регуляризация, dropout, early stopping.
15. Прикладные решения с элементами ИИ: интеграция и API. Разработка REST/gRPC API для инференса (FastAPI, Flask).
16. Контейнеризация моделей (Docker); оркестрация (Kubernetes). Тестирование: юнит-тесты для ML-компонентов.

17. Инфраструктура данных для ИИ-проектов. Выбор СУБД: PostgreSQL (структурированные данные), MongoDB (документы), Cassandra (time-series).
18. Инфраструктура данных для ИИ-проектов. ETL-инструменты (Apache NiFi, Airflow) и DataOps.
19. Инфраструктура данных для ИИ-проектов. Гибридные решения: SQL + NoSQL + векторные хранилища (Pinecone, Weaviate).
20. MLOps: управление версиями и CI/CD для ML. Инструменты: MLflow (трекинг экспериментов), DVC (версия данных), Kubeflow.
21. Автоматизация пайплайнов: GitHub Actions, GitLab CI. A/B-тестирование и многорукий бандит для развёртывания моделей.
22. Мониторинг и логирование в продакшене. Метрики: latency, throughput, error rate, data drift.
23. Дашборды (Grafana, Prometheus); трейсинг запросов (Jaeger). Аудит решений модели и логов безопасности.
24. Безопасность и приватность в ИИ-системах. Защита от adversarial attacks (FGSM, PGD). Приватность: дифференциальная приватность, федеративное обучение.
25. Резервирование и отказоустойчивость (multi-region деплоймент).
26. Управление ИИ-проектами: планирование и ресурсы. Методологии: Agile для ML-команд, Scrum с спринтами на эксперименты.
27. Бюджетирование: расчёт стоимости облачных ресурсов (GPU/TPU). Риски и метрики успеха (время до MVP, точность в продакшене).
28. Инструменты промышленной разработки ИИ. Использование Seldon Core и KServe для развёртывания моделей.
29. Инструменты промышленной разработки ИИ. Использование TensorBoard для визуализации обучения.
30. Инструменты промышленной разработки ИИ. Использование Weights & Biases для трекинга.
31. Инструменты промышленной разработки ИИ. CI/CD для ML: автоматизация ретрейнинга и деплоя.
32. Кейсы и антипаттерны: уроки реальных проектов. Успешные внедрения: рекомендательные системы, чат-боты, предиктивная аналитика. Типичные ошибки: некачественные данные, недооценка сопровождения, «чёрные ящики».

Оператор ЭДО ООО "Компания "Тензор"

ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ

ПОДПИСАНО ЗАВЕДУЮЩИМ КАФЕДРЫ	**ФГБОУ ВО "РГРТУ", РГРТУ,** Корячко Вячеслав Петрович, Заведующий кафедрой САПР	**15.12.25** 14:34 (MSK)	Простая подпись
ПОДПИСАНО ЗАВЕДУЮЩИМ ВЫПУСКАЮЩЕЙ КАФЕДРЫ	**ФГБОУ ВО "РГРТУ", РГРТУ,** Корячко Вячеслав Петрович, Заведующий кафедрой САПР	**15.12.25** 14:34 (MSK)	Простая подпись