Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«Уральский федеральный университет
имени первого Президента России Б.Н. Ельцина»

Институт радиоэлектроники и информационных технологий — РТФ

Школа профессионального и академического образования

ДОПУСТИТЬ К ЗАЩИТЕ ПЕРЕД ГЭК

РОП И.Н. Обабков

(подпись)

«29» мая 2023 г.

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

РАЗРАБОТКА УЧЕБНО–ПРАКТИЧЕСКОГО КОМПЛЕКСА ДЛЯ ДИСЦИПЛИНЫ «АВТОМАТИЗАЦИЯ МАШИННОГО ОБУЧЕНИЯ».

Научный руководитель: Корелин И. А. подпись
к. т. н., доцент
Нормоконтролер: Огуренко Е. В. подпись
Студент группы РИМ-210990 Токарев А. В. подпись

Екатеринбург

2023

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение высшего образования
«Уральский федеральный университет имени первого Президента России Б.Н. Ельцина»

Институт радиоэлектроники и информационных технологий-РТФ
Школа профессионального и академического образования
Направление подготовки 09.04.04 Программная инженерия
Образовательная программа «Разработка и управление в программных проектах»

УТВЕРЖДАЮ
РОП _____
«29» мая 2023 г.

ЗАДАНИЕ

на выполнение выпускной квалификационной работы

студента Токарева Александра Владимировича **группы** РИМ-210990
(фамилия, имя, отчество)

1. Тема выпускной квалификационной работы *Разработка учебно–практического комплекса для дисциплины «Автоматизация машинного обучения».*
Утверждена распоряжением по институту от «____» _____ 2023 г. № ___
2. Руководитель Корелин Иван Андреевич, доцент, к.т.н.
(Ф.И.О., должность, ученое звание, ученая степень)
3. Исходные данные к работе *методические материалы курса «Автоматизация администрирования MLOps»*
4. Перечень демонстрационных материалов презентация
5. Календарный план

№ п/п	Наименование этапов выполнения работы	Срок выполнения этапов работы	Отметка о выполнении
1.	*Подготовка технического обеспечения учебного комплекса «Автоматизация машинного обучения»*	до 26.03.2023 г.	
2.	*Реализация кейсов учебного комплекса «Автоматизация машинного обучения»*	до 20.04.2023 г.	
3.	*Разработка программного обеспечения автоматизации проверок практических заданий по дисциплине «Автоматизация машинного обучения»*	до 23.05.2023 г.	
4.	*ВКР в целом*	до 05.06.2023 г.	

Руководитель _____ Корелин И. А.
(подпись) Ф.И.О.
Задание принял к исполнению 13.02.2023 _____
дата (подпись)

6. Выпускная квалификационная работа закончена «05» июня 2023г. Считаю возможным допустить Токарева Александра Владимировича к защите выпускной квалификационной работы в Государственной экзаменационной комиссии.
Руководитель _____ Корелин И. А.
(подпись) Ф.И.О.

7. Допустить Токарева Александра Владимировича к защите магистерской диссертации в Государственной экзаменационной комиссии.
РОП _____ Обабков И. Н.
(подпись) Ф.И.О.

РЕФЕРАТ

Пояснительная записка по ВКР содержит 95 с., 27 рис., 6 табл., 30 источника, 5 прил.

РАЗРАБОТКА УЧЕБНО–ПРАКТИЧЕСКОГО КОМПЛЕКСА ДЛЯ ДИСЦИПЛИНЫ «АВТОМАТИЗАЦИЯ МАШИННОГО ОБУЧЕНИЯ».

Объект исследования: является процесс обучения MLOps.

Цель работы: состоит в создании учебно-практического комплекса для дисциплины «Автоматизация машинного обучения» для обучения студентов основным инструментам и технологиям автоматизации машинного обучения с возможностью дальнейшего применения полученных знаний в профессиональной деятельности. Для достижения поставленной цели необходимо решить следующие задачи.

Методы исследования включают в себя:

1) Анализ методических документов, сравнение используемых технологий, систематизацию и обобщение данных о существующих онлайн-курсах по дисциплине «Автоматизация машинного обучения»;

2) Анализ современных программных инструментов, позволяющих выполнить и ускорить процесс автоматизации машинного обучения;

СОДЕРЖАНИЕ

РЕФЕРАТ 2
ОПРЕДЕЛЕНИЯ, ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ 5
ВВЕДЕНИЕ 6
1 ГЛАВА «ПОДГОТОВКА ТЕХНИЧЕСКОГО ОБЕСПЕЧЕНИЯ УЧЕБНОГО КОМПЛЕКСА “АВТОМАТИЗАЦИЯ МАШИННОГО ОБУЧЕНИЯ”» 9
1.1 Сравнительный анализ систем LMS 9
1.2 Сравнительный анализ облачных платформ 23
2 ГЛАВА «РЕАЛИЗАЦИЯ КЕЙСОВ УЧЕБНОГО КОМПЛЕКСА “АВТОМАТИЗАЦИЯ МАШИННОГО ОБУЧЕНИЯ”» 31
2.1 Теоретическое обоснование разработки учебно-практического комплекса 31
2.2 Определение состава разрабатываемого комплекса 36
3 ГЛАВА «РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ АВТОМАТИЗАЦИИ ПРОВЕРОК ПРАКТИЧЕСКИХ ЗАДАНИЙ ПО ДИСЦИПЛИНЕ “АВТОМАТИЗАЦИЯ МАШИННОГО ОБУЧЕНИЯ”» 70
3.1 Проектирование ПО 70
3.2 Разработка основного функционала 74
3.3 Разработка плагина проверки первого практического задания 77
3.4 Разработка плагина проверки второго практического задания 78
3.5 Разработка плагина проверки третьего практического задания 79
3.6 Разработка плагина проверки четвертого практического задания 81
3.7 Разработка плагина проверки пятого практического задания 82
ЗАКЛЮЧЕНИЕ 83
БИБЛИОГРАФИЧЕСКИЙ СПИСОК 84
ПРИЛОЖЕНИЕ А 90
ПРИЛОЖЕНИЕ Б 91

ПРИЛОЖЕНИЕ В 92
ПРИЛОЖЕНИЕ Г 93
ПРИЛОЖЕНИЕ Д 95

ОПРЕДЕЛЕНИЯ, ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

В пояснительной записке используются следующие определения, обозначения и сокращения

ПО	Программное обеспечение
CI	Непрерывная интеграция
CD	Непрерывная доставка
CT	Непрерывное тестирование
LMS	Система управления обучением
S3	Облачное объектное хранилище
ВМ	Виртуальная Машина
MLOps	Операции машинного обучения

ВВЕДЕНИЕ

Актуальность темы исследования. Современный рынок труда растет и появляются новые профессии и новые специальности, образовательным учреждениям приходится реагировать на вызов. Возникает необходимость в обучении студентов новым специальностям. Одна из них — автоматизация машинного обучения (анг. MLOps — Machine Learning Operations — Машин Лёрнинг Оперейшенс) Согласно отчету Deloitte, к 2025 году объем мирового рынка MLOps вырастет почти до $4 млрд, а также согласно исследованию it-world на 17.11.2022 дефицит кадров ML-инженеров составляет 50% от рыночной потребности [1]. Чтобы обеспечить новую дисциплину образовательным контентом было принято решение о создании учебно-практического комплекса для дисциплины «Автоматизация машинного обучения». Одной из задач комплекса является формирование теоретических знаний и практических навыков по автоматизации машинного обучения у студентов, а также способствование обучению использованию студентами программного инструментария для выполнения автоматизации в машинном обучении.

В условиях большого количества студентов для учебного заведения со статусом университета логично разработать дистанционный курс. В использовании преимуществ дистанционного обучения заинтересованы не только учебные заведения, но и многие компании по всему миру.

Для дистанционного обучения удобнее использовать онлайн-курс.

Таким образом, была сформирована формулировка темы ВКР: «Разработка учебно-практического комплекса для дисциплины “Автоматизация машинного обучения”».

Гипотеза Применение современных облачных платформ и облачных сервисов в разработке интерактивного практического комплекса для дисциплины «Автоматизация машинного обучения» позволит реализовать

качественный учебный курс, повысит удобство проверки практических заданий, позволит сократить время на проверку работ студентов по дисциплине «Автоматизация машинного обучения».

Цель исследования состоит в создании учебно-практического комплекса для дисциплины «Автоматизация машинного обучения» для обучения студентов основным инструментам и технологиям автоматизации машинного обучения с возможностью дальнейшего применения полученных знаний в профессиональной деятельности. Для достижения поставленной цели необходимо решить следующие задачи.

Задачи:

– Провести сравнительный анализ и обосновать выбор LMS и облачных платформ для учебно-практического комплекса по дисциплине «Автоматизация машинного обучения»;
– Реализовать тестовые, практические и контрольные задания для интерактивного практического комплекса по дисциплине «Автоматизация машинного обучения»;
– Разработать программное обеспечение для автоматизации проверок практических заданий по дисциплине «Автоматизация машинного обучения»

Объектом исследования является процесс обучения MLOps.

Предметом исследования является обучающий комплекс для студентов по техническим дисциплинам.

Методы исследования включают в себя:

1) Анализ методических документов, сравнение используемых технологий, систематизацию и обобщение данных о существующих онлайн-курсах по дисциплине «Автоматизация машинного обучения»;

2) Анализ современных программных инструментов, позволяющих выполнить и ускорить процесс автоматизации машинного обучения;

Теоретической основой исследования стали:

1) Отечественные и зарубежные исследования в области разработки онлайн-курса MLOps;

2) Техническая документация облачных платформ.

Теоретическая и практическая значимость.

Теоретическая значимость работы обусловлена недостаточностью исследований образовательных технологий по дисциплине «Автоматизация машинного обучения».

Практической значимостью работы стала возможность реализации и использования учебно-практического комплекса для дисциплины «Автоматизация машинного обучения» на практике для обучения студентов технических направлений.

1 Глава «подготовка технического обеспечения учебного комплекса “автоматизация машинного обучения”»

1.1 Сравнительный анализ систем LMS

Для того, чтобы решить задачу по созданию учебного практического комплекса, необходимо выбрать LMS (Learning Management System), на которой будет создан и реализован данный комплекс. Для выполнения этой задачи был проведен сравнительный анализ систем LMS по некоторым характеристикам, которые имеют наибольшую значимость для удовлетворения требований по созданию учебного практического комплекса.

Введем определение системы LMS. Learning Management System (LMS) — это сервис для организации и управления обучением, с помощью которого можно [2]:

– создавать обучающий контент разных типов: онлайн—уроки, курсы, лекции, вебинары, презентации, тренинги, мастер—классы, тесты;

– предоставлять доступ к учебным материалам конкретным группам пользователей (организовывать «виртуальные классы»);

– устанавливать обратную связь между преподавателями и обучающимися (пользователи общаются в подходящем для них формате — в чате, на форуме, в режиме онлайн-конференции);

– отслеживать результаты обучения для этого в системе предусмотрена функция автоматического составления отчетов об успеваемости и посещаемости.

По сути, это универсальный инструмент планирования и контроля учебного процесса. С его помощью можно организовать как групповое, так и индивидуальное обучение.

Можно выделить несколько основных функций, которые выполняет LMS:

1) Управление контентом. Оно включает следующие возможности:

– собирать весь контент в одном месте и создавать корпоративную базу знаний,

– загружать на платформу уже готовые курсы,

– создавать собственные интерактивные видео во встроенном редакторе обучающей платформы,

– добавлять интерактивные вставки и тесты, чтобы курс нельзя было смотреть на перемотке,

– предоставлять доступ к контенту с любых устройств, включая старые модели телефонов,

– обеспечивать разным сотрудникам разные уровни доступа к материалам,

– приглашать сотрудников на обучающие курсы, напоминать о ближайших занятиях [3].

2) Отчётность и отслеживание результатов. Может включать следующие объекты и события:

– кто из студентов прошёл курс, с какой попытки и с каким результатом выполнил задания,

– какой прогресс в обучении у выбранного студента или группы,

– выгрузка результатов обучения

Стоит отметить, что удобно, когда гибкий и детальный отчёт по обучению можно выгрузить в Excel и CSV форматах.

Стоит определиться и с форматом обучения, который позволяют реализовать LMS. Чаще всего системы управления обучением используют для реализации приведенных ниже типов обучения.

3) Дистанционное обучение

Дистанционное обучение — это форма обучения, при которой учителя и учащиеся взаимодействуют между собой на расстоянии, используя различные технологии. Дистанционное обучение может быть частью дистанционного образования, которое включает в себя различные виды образовательных

программ. Дистанционное обучение может быть организовано как в рамках образовательных учреждений, так и в рамках дополнительного профессионального образования [4].

4) Гибридное обучение

Что мы имеем в виду, когда говорим про смешанное или гибридное обучение? Технология смешанного обучения — это объединение классических форм обучения в аудитории с электронным обучением. Есть несколько вариантов таких курсов. Например, преподаватель может проводить обучение в аудитории, взаимодействуя с аудиторией через мобильное приложение. В нём можно проводить опросы, давать тестовые задания или демонстрировать презентации.

Другой вариант — обучение через LMS с возможностью задавать вопросы преподавателю или отправлять задания на проверку.

Такой тип обучения включает как аудиторную работу преподавателя со студентами, так и активное использование методов дистанционного обучения, например, использование вебинаров и электронных учебных материалов [5].

5) Индивидуальные траектории обучения

Одно из важных преимуществ онлайн-обучения — индивидуальный подход. Студенты отличаются друг от друга опытом, уровнем знаний, темпераментом, скоростью усвоения информации, мотивацией, и так далее

Так, стоит отметить, что обучение с помощью LMS-платформы позволяет легко персонализировать программу. Такие платформы предполагают возможности:

– составления отдельных курсов для разных студентов,

– предоставления доступа к разным материалам для новичков и слушателей продвинутого уровня,

– выдачи дополнительных заданий тем, кому требуется больше времени на освоение материала.

Индивидуальная траектория обучения означает, что студенты могут получать только нужный образовательный контент, основанный на истории и роли каждого учащегося [6].

LMS бывают двух типов — облачные платформы и «коробочные» решения. Рассмотрим их подробнее.

Облачные платформы наиболее популярный вид LMS, который обычно представляет веб сервис со всем необходимым функционалом. Можно отметить, что данный вид систем крайне прост, что позволяет пользователям легко ориентироваться на платформе и запускать курс без помощи специалистов. Бывает достаточно просто зарегистрировать и создать курс по предложенному шаблону или конструктору.

Коробочные решения. Такой вид обычно используют организации или учебные заведения. Данный вид систем обычно представляет из себя ПО (программное обеспечение), которое требует выделение ресурсов и установки на внутренний сервер. Отсюда вытекает, что для обслуживания таких систем нужен квалифицированный специалист с необходимыми навыками [7].

Так же стоит отметить, что помимо описанных двух вариантов есть системы с открытым исходным кодом. Такие системы требуют навыков программирования, чтобы можно было ими пользоваться и ввести их в эксплуатацию, но в большинстве случаев такие системы бесплатные.

Рассмотрим подробнее процесс обучения с использованием LMS.

Обучение на платформе можно сравнить с традиционными методами обучения. Большинство вузов пользуются популярными LMS или имеют свою собственную систему. Построение уроков может быть разным и зависеть от преподавателя.

Если рассматривается облачный сервис, то первым шагом выполняется регистрация пользователей в системе. В коробочных же решениях функционал доступен сразу.

Изначально преподавателю или автору курса необходимо подготовить материал, который будет размещен на платформе. Здесь также есть несколько

вариантов: либо загрузка готового курса, либо генерация учебного материала самостоятельно при помощи системы и ее инструментов.

Следующим шагом идет добавление пользователей к курсу, но в большинстве LMS это процесс происходит автоматически. В большинстве систем права доступа разграничены для разных ролей пользователей: учеников, администраторов, преподавателей, авторов курса (часто бывает так, что последние два совпадают).

Далее следует выполнение слушателями самого курса. Оно включает изучение теоретического материала — это могут быть как текстовые материалы, видеоматериалы, так и непосредственная работа с преподавателем; и выполнение практических заданий. Чаще всего обучение строится таким образом, что темы и задания открываются последовательно, по порядку, и чтобы приступить к выполнению следующего раздела, предыдущий должен быть выполнен. К примеру, чтобы выполнить пятое задание, нужно, чтобы были выполнены все четыре предыдущих задания [8].

Практические задания могу иметь различные формы, например:

– Выполнение лабораторной работы с возможностью загрузки файла,
– Тесты с единственным вариантом ответа,
– Тесты с множественными вариантами ответа,
– Тесты со свободным ответом,
– Задания на соответствие,
– Задания на заполнение пропущенных элементов,
– Задания на расположение элементов в нужном порядке,
– Интерактивные задания с графическими элементами,
– И т.д.

После выполнения задания поступают на проверку к преподавателям, либо другим квалифицированным специалистам. Чаще всего выдается обратная связь при проверке задания — если задание выполнено неверно, оно

будет отправлено студенту на повторное решение или на доработку. После того, как задание будет выполнено, откроется следующий раздел.

На последнем этапе происходит оценка результатов и подведение итогов обучения. Для этого могут применяться проверочные задания и инструменты, которые предоставляет система. Чаще всего в LMS есть возможность генерации отчета, демонстрирующего, сколько времени учащиеся потратили на прохождения курса, какие задания были выполнены и сколько баллов было получено.

Далее перейдем непосредственно к выбору и анализу различных систем LMS. Для сравнения были выбраны 5 вариантов LMS платформ, которые можно было бы использовать для реализации учебного практического комплекса:

- Moodle
- Openedu.urfu
- Microsoft teams
- Canvas
- Google Classroom

Рассмотрим каждую из них подробнее.

Moodle — это система управления обучением с открытым исходным кодом. Эта (коробочная) платформа электронного обучения включает в себя современную настраиваемую панель и удобный для пользователя макет [9].

Openedu — современная образовательная платформа, предлагающая онлайн-курсы по базовым дисциплинам, изучаемым в российских университетах. Платформа создана Ассоциацией "Национальная платформа открытого образования", учрежденной ведущими университетами: МГУ им. М.В. Ломоносова, СПбПУ, СПбГУ, НИТУ «МИСиС», НИУ ВШЭ, МФТИ, УрФУ и Университет ИТМО.

Microsoft Teams — корпоративная платформа с функциональными элементами системы LMS, которая позволяет создавать команды, выкладывать задания, назначать встречи [10].

Canvas — это платформа для онлайн-обучения с открытым исходным кодом. Это совершенно бесплатная система управления обучением (LMS) для школ, университетов и учебных центров. Достоинства Canvas — это гибкость, надежность, настраиваемость. Данная онлайн-платформа позволяет учащимся проходить курсы, смотреть семинары, лекции и выполнять домашние задания онлайн. Преподаватели и студенты могут общаться через видеоконференции и популярные мессенджеры.

Google Classroom или «Гугл Класс» — интернет-сервис для онлайн-обучения. Позволяет создавать курсы, проводить вебинары и реализовывать практические и проверочные задания.

Так, для проведения сравнительного анализа выбранных систем LMS, на основании приведенного выше раскрытия функциональных возможностей систем управления обучением, были определены критерии их сравнения, наиболее значимые в контексте обучения дисциплине «Автоматизация машинного обучения». Эти характеристики:

– наличие мобильного приложения,
– кроссбраузерность,
– техподдержка для выкладываемых материалов,
– поддержка вебинаров,
– возможность создания тестов, опросов и различных практических заданий,
– возможность выдачи домашних заданий,
– возможность выставления оценки курса учениками,
– возможность выставления оценки работ преподавателем,
– возможность ограничения доступа к материалам с определенными условиями,

– возможность оставления комментариев,
– возможность самостоятельного добавления учеников к курсу,
– выгрузка статистики по успеваемости пользователей,
– выгрузка аналитики,
– возможность создания e-mail рассылки с платформы,
– стоимость использования LMS.

Рассмотрим подробнее и обоснуем выбор каждой из характеристик.

Наличие мобильного приложения.

На сегодняшний день наличие мобильного приложение — достаточно важный плюс для любого приложения. Мобильное приложение позволит студентам получать доступ к контенту LMS даже там, где нет возможности использовать компьютер или ноутбук. Наличие мобильного приложения призвано повысить активность работы студентов с курсом.

Выбор такой характеристики как кроссбраузерность связан с тем, что на данный момент насчитывается большое количество различных браузеров, с различными версиями и совместимостью. У обучающихся по онлайн-курсам могут встретиться любые браузеры и их версии, и лучший вариант для реализации учебного комплекса — когда LMS совместима с максимальным числом веб-браузеров, что даст возможность максимальному количеству студентов без проблем работать с онлайн-средой, используя привычные для себя инструменты.

Техподдержка для выкладываемых материалов. Характеристика подразумевает, что размещением учебных материалов на платформе, а также дальнейшей технической поддержкой работоспособности этих материалов, занимаются специалисты от самой платформы. Эта характеристика позволяет облегчить работу с платформой преподавателям и авторам курсов, и в случае возникновения технических ошибок, у пользователей всегда есть возможность обратиться к службе поддержки.

Поддержка вебинаров. Подразумевает возможность создания онлайн-встречи или вебинара средствами самой платформы LMS, без использования стороннего ПО. Занятия в формате вебинаров расширяют возможный спектр представления информации студентам преподавателем. Особенно важна данная характеристика в формате дистанционного обучения, когда у студентов и преподавателей нет возможности находиться очно в аудитории. Автоматизация машинного обучения — не самая простая для освоения дисциплина, в рамках которой студентам зачастую может понадобиться помощь преподавателя в режиме реального времени.

Возможность создания тестов, опросов и различных практических заданий — одна из основных характеристик для учебного практического комплекса. Неотъемлемая часть любого обучающего курса, в рамках которого формируются практические навыки и профессиональные компетенции — это выполнение студентами практических и тестовых заданий. Автоматизации машинного обучения — не исключение. В рамках дисциплины необходимо регулярно проверять степень освоения теоретического материала студентами, а также формировать практические навыки по дисциплине.

Возможность выдачи домашних заданий. Данная характеристика предполагает предоставляемую преподавателю возможность задавать задания студентам и получать от них результат средствами системы LMS. Такие задания обычно подразумевают назначение длительного срока выполнения, а также автоматизированное средство сбора результатов, иногда автоматизированную проверку результатов. Данная возможность — возможность выдачи заданий для самостоятельной домашней работы — важна для любого учебного процесса, помогает студентам оценить степень готовности самостоятельно справляться с заданиями по изучаемой дисциплине.

Возможность выставления оценки курса учениками. Характеристика позволяет получать обратную связь от студентов по пройденному курсу, что при условии регулярного сбора статистики, дает возможность выявлять

слабые места, ошибки и неточности в учебном материале. Это позволяет корректировать и улучшать учебный курс, а также контролировать актуальность предоставляемого материала, степень довольства обучающихся и удобства построения учебного процесса. В контексте обучения дисциплине «Автоматизация машинного обучения» особенно острым может стать вопрос об устаревании учебного материала в связи с быстротой развития современных используемых технологий. Так, сбор обратной связи о курсе у студентов поможет лучше контролировать актуальность материалов учебного практического комплекса.

Возможность выставления оценки работ преподавателем — еще одна из важнейших для учебного процесса характеристик, которая позволяет преподавателю производить оценивание выполнения заданий студентов. Данная характеристика подразумевает, что у преподавателя на платформе LMS есть возможность просмотра результатов выполнения тех или иных заданий студентами и выставления оценок «вручную», т.е. по собственным критериям без использования автоматизированных средств. А также присутствует возможность дальнейшей корректировки выставленных оценок. В рамках обучения автоматизации машинного обучения такая характеристика важна по причине возможного выполнения комплексных и сложных практических заданий без использования средств LMS, но с предоставлением результатов на платформу для удобства единой системы оценивания.

Возможность ограничения доступа к материалам с определенными условиями. Данная характеристика LMS позволяет настраивать ограничения доступа к материалам курса по определенным условиям. Наиболее удобный вариант — предоставление платформой максимального количества различных настраиваемых условий и фильтров для данной цели. Например, это могут быть временные ограничения — выполнение тем курса происходит в четко определенные временные рамки. Или это может быть условие последовательности выполнения заданий — задание 3 невозможно выполнить, не выполнив перед этим 1 и 2 задания. Также бывают условия

предоставления доступа к определенному материалу для определенных групп или подгрупп студентов — бывает актуально для деления студентов по подгруппам, например, в зависимости от степени владения материалом. Это позволяет преподавателю более эффективно планировать подачу материала в рамках дисциплины.

Возможность оставления комментариев. Характеристика представляет своего рода функционал форума на платформе LMS. Бывают комментарии закрытого и открытого типов. Первые имеют ограничения на права чтения для пользователей, вторые доступны всем. Комментарии могут быть оставлены как студентами, так и преподавателями. Они обязательно прикрепляются к какому-либо объекту: лекционному материалу, практическому или домашнему заданию и т.д. К комментарию в системе можно оставлять ответ. Это способствует более активному взаимодействию между студентами и преподавателями, позволяет получать своего рода обратную связь от студентов о курсе в порядке его прохождения.

Возможность самостоятельного добавления учеников к курсу предполагает возможность для преподавателя самостоятельно (без помощи и разрешения специалистов поддержки LMS) добавлять учеников к учебному курсу. Также актуально подключение студентов по учебным группам и деление их на подгруппы в процессе обучения. В связи с тем, что во время учебного процесса состав обучающихся, хоть и незначительно, но может изменяться, преподавателю важно иметь возможность быстро и без лишних обращений и проблем изменять состав обучаемой группы в LMS.

Выгрузка статистики по успеваемости пользователей. Характеристика позволяет преподавателю (а иногда и самим студентам) следить за успеваемостью по изучаемому курсу. Оценка результатов усвоения материала и определение статистики — неотъемлемая часть любого учебного процесса. Средства систем управления обучением предоставляют удобные инструменты для автоматизации подсчетов и наглядной демонстрации результатов успеваемости обучающихся. Иногда есть возможность гибко настраиваемой

фильтрации статистических данных, что делает работу преподавателя с LMS еще более удобной.

Выгрузка аналитики. Данная характеристика подразумевает возможность получения преподавателем подробных результатов прохождения курса обучающимися, адаптированных для выставления оценок на сторонних ресурсах. Зачастую высшие учебные заведения работают с проприетарными едиными системами оценивания успеваемости студентов. Большим плюсом для LMS является возможность генерации результатов студентов в наглядном графическом виде для удобства дальнейшей обработки результатов и внесения их на сторонние ресурсы.

Возможность создания e-mail рассылки с платформы. Данная характеристика удобна в рамках любого образовательного процесса. Позволяет делать широковещательные e-mail рассылки на группу обучаемых студентов (или даже использовать фильтрацию получателей по определенным критериям), при этом избавляя пользователя с ролью преподавателя от ручного ввода большого количества получателей электронной почты. Служит для поддержания более оперативной и удобной связи со студентами.

Стоимость использования LMS — критически важная характеристика для обучения учебных групп студентов в высших учебных заведениях. Большим плюсом для LMS является возможность бесплатного использования предоставляемых сервисов, в противном случае учебным организациям необходимо планировать бюджет в соответствии с количеством онлайн-курсов и используемыми системами [11 — 13].

По представленным критериям был выполнен сравнительный анализ приведенных выше систем управления обучением. Результаты сравнения в таблице 1.

Таблица 1 — сравнительный анализ систем LMS

Параметр	Moodle	Openedu.urfu	Microsoft teams	Canvas	Google Classroom

Наличие мобильного приложения	+	-	+	+	+
Кроссбраузерность	+	+	+	+	+
Техподдержка для выкладываемых материалов	-	+	-	-	+
Поддержка вебинаров	-	-	+	+	+
Возможность создания тестов, опросов и различных практических заданий	+	+	+	+	+
Возможность выдачи домашних заданий	+	+	+	+	+
Возможность выставления оценки курса учениками	-	-	-	-	-
Возможность выставления оценки работ преподавателем	+	-	+	+	+
Возможность ограничения доступа к материалам с определенными условиями	+	+	+	+	+
Возможность оставления комментариев	+	+	-	+	+
Возможность самостоятельного добавления учеников к курсу	+	-	+	+	+

Выгрузка статистики по успеваемости пользователей	+	-	-	+	+
Выгрузка аналитики	+	+	-	+	+
Возможность создания e-mail рассылки с платформы	+	+	-	+	+
Стоимость использования LMS	бесплатно	частично платный функционал	бесплатно	бесплатно	частично платный функционал

Все рассматриваемые LMS имеют свои плюсы и минусы. Опираясь на результаты сравнения, было выявлено, что для учебного практического комплекса наиболее подходящей системой является Moodle. Такой вывод обусловлен максимальным количеством выполненных требований (12), в соответствии с составленной таблицей критериев, а также наличием необходимого и удобного инструментария и функционала для создания учебного онлайн-курса.

Ряд причин имеет также отклонение других приведенных LMS.

Так, например, Google Classroom соответствует многим выдвинутым характеристикам и имеет обширный функционал, но значительным минусом является требование наличия платной подписки для работы некоторого функционала платформы. Это существенный критерий для реализации в высших учебных заведениях.

Canvas — неплохая по функционалу LMS и могла бы стать хорошей альтернативой для реализации учебного практического комплекса, но Canvas имеет существенные требования к выделяемым ресурсам. Так, данная платформа разворачивается на внутренних сервисах учебных заведений, а значит требует планирования и выделения технических ресурсов, что

становится неоспоримым минусом при сравнении функционала различных LMS.

Microsoft Teams и Openedu.urfu имеют ряд минусов, существенных для реализации учебного практического комплекса.

1.2 Сравнительный анализ облачных платформ

Для эффективной работы студентов на практической части курса и автоматизации проверки их работы возникает потребность в выборе оптимальной платформы для реализации интерактивного практического комплекса. В контексте обучения автоматизации машинного обучения такая платформа может использоваться для реализации непосредственно практических заданий, формирующих практические навыки реализации конвейеров машинного обучения различными методами и сопровождения проектов машинного обучения. Важно учитывать, какие решения востребованы на мировом уровне, и какие варианты могут быть использованы, исходя из условий технологического суверенитета РФ. Для этого были изучены несколько облачных платформ с целью выбрать наиболее подходящие под требования реализации интерактивного практического комплекса.

Для сравнительного анализа было решено выбрать 4 платформы отечественных компаний и 4 платформы зарубежных компаний.

Рассмотрим все платформы подробнее. Зарубежные:

Alibaba Cloud — это компания, занимающаяся облачными вычислениями. Она предоставляет набор облачных продуктов и услуг для предприятий по всему миру. Она предлагает ряд услуг, таких как: хранение данных, обработку больших данных, защиту от DDoS и т.д. Компания была основана в 2009 году и предоставляет услуги облачных вычислений и управления данными для крупных и малых предприятий.

Google Cloud (GC) — это набор сервисов облачных вычислений, предлагаемых Google. Она включает в себя более 100 продуктов, таких как

решения по управлению данными, гибридные и мультиоблачные решения, и т.д. GC широко используется благодаря своим удобным и простым в использовании инструментам, и услугам.

Microsoft Azure — это служба облачных вычислений, которая предлагает множество функций, включая расширенные модели кодирования и языка, управляемые облачные базы данных Azure SQL и анализ контента с помощью ИИ. Она является частью платформы Microsoft Cloud, в которую также входят Microsoft Security, Microsoft 365, Dynamics 365 и другие отраслевые решения.

Amazon Web Services (AWS) — это платформа облачных вычислений, предоставляемая Amazon, которая предлагает широкий спектр услуг, включая инфраструктуру как услугу (IaaS), платформу как услугу (PaaS) и программное обеспечение как услугу, т.е. ряд предложений a-service (SaaS). AWS предоставляет инструменты для вычислительной мощности, хранения баз данных, доставки контента и т. д. Была запущен в 2006 году и с тех пор стала одной из самых комплексных облачных платформ в мире с центрами обработки данных по всему миру. AWS широко используется стартапами, крупными предприятиями и правительствами. Платформа предлагает модель облачных вычислений с оплатой по мере использования и предоставляет инструменты, которые автоматически оценивают потенциальные риски безопасности [14].

Также были рассмотрены следующие отечественные платформы:

Yandex Cloud — общедоступная облачная платформа, разработанная российской интернет-компанией Яндекс. Предоставляет инфраструктуру и вычислительные ресурсы в формате «как услуга», включая услуги по управлению инфраструктурой и данными, инструменты для разработки облачных приложений и моделей машинного обучения, а также проприетарные услуги на основе машинного обучения. Yandex Cloud использует ту же инфраструктуру, что и основные сервисы Яндекса, и располагается в тех же дата-центрах.

VK Cloud — корпоративный облачный сервис, созданный VK, ведущей российской интернет-компанией. Он предлагает масштабируемые и безопасные ИТ-услуги для инфраструктуры и разработки. VK Cloud также имеет сервис облачных игр под названием VK Play Cloud, который позволяет пользователям играть в современные компьютерные игры на своих смартфонах или планшетах.

SberCloud — облачный провайдер, предлагающий услуги IaaS/PaaS и машинного обучения. Ранее он принадлежал российскому банку Сбербанк, но в мае 2022 года банк продал свою долю в SberCloud. SberCloud предлагает такие услуги, как эластичный облачный сервер и механизм облачных контейнеров. В последние годы компания продемонстрировала значительный рост выручки.

CloudMTS — ИТ-сервисная и консалтинговая компания, предоставляющая облачные сервисы. Она является одним из крупнейших облачных провайдеров в России со своими дата-центрами и каналами связи по всей стране. Облачные сервисы CloudMTS спроектированы так, чтобы быть надежными, масштабируемыми и простыми в управлении. Они предлагают облачную модель подписки на безопасное и современное офисное программное обеспечение, что снижает затраты на инфраструктуру для клиентов и позволяет быстро организовать рабочий процесс с использованием облачных решений [15].

Для того, чтобы выбрать, какие характеристики существующих платформ наиболее востребованы, были изучены платформы компаний-мировых лидеров, которые можно бы было использовать для решения потребностей в реализации интерактивного практического комплекса по дисциплине. В качестве критериев сравнения был выбран ряд характеристик предоставляемых платформами сервисов, а именно:

Предоставление первоначального гранта либо бесплатного доступа для работы с платформой. Выбор данной характеристики обусловлен необходимостью учебных заведений планировать и выделять бюджет для

реализации практических работ студентов. Отсутствие такой необходимости — т.е. бесплатный доступ к платформе, становится существенным преимуществом при выборе облачной платформы для реализации практикума.

Наличие технологии контейнеризации. Контейнеризация — это процесс развертывания программного обеспечения, при котором для разворачиваемого ПО сформировано изолированное виртуальное пространство. Выбор данной характеристики связан непосредственно с содержанием практических работ по автоматизации машинного обучения. В рамках практики студентам предстоит освоить работу по разворачиванию моделей машинного обучения в контейнерах. В частности, например, студенты столкнутся с технологиями Docker и Docker Compose. Docker — это проект с открытым исходным кодом для автоматизации развертывания приложений в виде переносимых автономных контейнеров, выполняемых в облаке или локальной среде.

Наличие интеграции с Github. Под интеграцией в данном случае понимаются средства и службы, производящие подключения к GitHub Enterprise Cloud для дополнения и расширения рабочего процесса. Применение такой технологии позволяет достичь сразу нескольких целей. В первую очередь — создание условий для автоматизированной сборки, упаковки и проверки готового продукта на предмет ошибок. Во вторую — сокращение количества ошибок в создании программного обеспечения, повышение скорости развёртывания сервиса или приложения в разных средах. Данная характеристика облачной платформы поможет студентам лучше освоить и научиться применять современные технологии CI/CD.

Наличие объектного хранилища. Объектное хранилище здесь — это технология хранения и управления данными в неструктурированном формате, называемом объектами. В современных организациях производятся и анализируются большие объемы неструктурированных данных, таких как фотографии, видео, электронные письма, веб-страницы, сенсорные данные и

аудиофайлы. Управление такими данными входит в обучающий процесс дисциплины «Автоматизация машинного обучения».

Возможность создания прерываемой виртуальной машины. Прерываемые виртуальные машины — это виртуальные машины, которые могут быть принудительно остановлены в любой момент. Данная характеристика призвана обеспечить сокращение затрат на создание виртуальной машины, что также определяет планирование учебного процесса и бюджета для него.

Доступ на территории РФ. Данная характеристика определяет возможность беспрепятственного доступа и использования сервисов облачной платформы пользователей с территории Российской Федерации. Разрабатываемый учебный практический комплекс предназначен для реализации в рамках учебного процесса именно на территории РФ, что делает приведенный критерий одним из наиболее значимых при выборе облачной платформы.

Данные характеристики являются наиболее значимыми в процессе организации практических занятий в учебно-практическом комплексе. Наличие описанных характеристик у облачных платформ дает возможность в полной мере реализовать практические задания, предусмотренные учебно-практических комплексом, и обеспечить беспрепятственное их выполнение студентами.

Так, в соответствии с выделенными критериями, проведем сравнительный анализ выбранных зарубежных облачных платформ. Анализируемые платформы: Alibaba Cloud, Google Cloud, Azure Microsoft Cloud, Amazon Web Services.

Результаты соответствия критериям приведены в таблице 2.

Таблица 2 — Сравнение зарубежных облачных платформ

	Характеристики и сервисы

Название облачной платформы	предоставление гранта	наличие контейнеризации	интеграция с git	наличие объектного хранилища	прерываемость ВМ	доступ в России
Alibaba Cloud	Бесплатная пробная версия	Есть	Есть	Object Storage Service (OSS)	Нет	Есть
Google Cloud	Грант на 300$	Есть	Есть	Google Cloud Storage	Нет	Нет
Azure Microsoft Cloud	Грант на 12 бесплатных месяцев	Есть	Есть	Azure Blob Storage	Нет	Нет
Amazon Web Services	Грант на 12 бесплатных месяцев	Есть	Есть	Simple Storage Service (S3)	Есть	Нет

Как видно из таблицы 2, большинство зарубежных платформ предоставляют гранты или бесплатные пробные версии для того, чтобы пользователь смог протестировать возможности данной облачной платформы; имеют интеграцию с системами контроля версий (СКВ) и контейнеризацию. Стоит отметить, что прерываемые машины есть только у Amazon Web Services, что дает некоторое преимущество данной платформе. Отдельно стоит обратить внимание на Alibaba Cloud. Большой плюс данной платформы заключается в том, что доступ к платформе возможен из России. Одна из особенностей этой облачной платформы — ее объектное хранилище несколько отличается по своей структуре от других платформ, которые были рассмотрены.

Далее был проведен сравнительный анализ четырех облачных платформы отечественных компаний, опирающийся также на выделенные ранее критерии. Анализируемые платформы: Yandex Cloud, VK Cloud, SberCloud, CloudMTS.

Результаты сравнения представлены в таблице 3.

Таблица 3 — Сравнение отечественных облачных платформ

Название облачной платформы	Характеристики и сервисы					
	предоставление гранта	наличие контейнеризации	интеграция с git	наличие объектного хранилища	прерываемость ВМ	доступ в России
Yandex Cloud	Грант на 4000 р	Есть	Нет	Yandex Object Storage	Есть	Есть
VK Cloud	Грант на 3000р	Есть	Нет	VK Cloud Storage (S3)	Нет	Есть
SberCloud	Нет	Есть	Нет	Object Storage (S3)	Нет	Есть
CloudMTS	Грант на 5000р	Есть	Нет	Object Storage CloudMTS (S3)	Нет	Есть

Стоит отметить, что прерываемость ВМ есть только у Yandex Cloud, что дает преимущество данной облачной платформе в сравнении со своими конкурентами.

Сравнение зарубежных и отечественных облачных платформ — сложная тема, требующая детального анализа различных факторов, таких как функциональность, ценообразование, безопасность и пользовательский опыт. В нашем случае анализ был проведен по выбранных характеристикам, которые более всего актуальны для реализации учебно-практического комплекса.

После выполнения сравнительного анализа облачных платформ по заданным характеристикам, можно отметить, что почти все зарубежные облачные платформы, которые участвовали в сравнительном анализе, либо ушли из России, либо ограничили доступ к сервисам. По данной причине выбор облачной платформы для реализации комплекса происходил среди отечественных платформ. Yandex Cloud — единственный облачный провайдер в России, предлагающий полноценную масштабируемую внутреннюю инфраструктуру. Именно Yandex Cloud удовлетворяет

большинству выдвинутых характеристик (5) для реализации учебного практического комплекса. Поэтому было принято решение использовать облачную платформу Yandex Cloud для реализации части практических заданий в учебно-практическом комплексе.

2 Глава «реализация кейсов учебного комплекса “автоматизация машинного обучения”»

2.1 Теоретическое обоснование разработки учебно-практического комплекса

Во вводной части работы была доказана значимость и актуальность подготовки кадров в сфере автоматизации машинного обучения. Так, появилась необходимость создания учебно-практического комплекса по дисциплине «Автоматизация машинного обучения», отвечающего требованиям высших учебных заведений для возможности последующего внедрения разработанных материалов в учебный процесс высших учебных заведений.

В качестве целевой аудитории для освоения учебно-практического комплекса были определены студенты магистратуры специальностей IT-направлений. Выбор такой категории обучающихся обусловлен тем, что для освоения предлагаемой в рамках работы дисциплины слушателям необходимы теоретические знания и практические навыки в сфере информационных технологий и программной инженерии. Достаточные начальные знания слушателями могут быть получены в рамках освоения специальностей по IT-направлениям подготовки бакалавриата, либо приобретены в следствие опыта работы в сфере программной инженерии.

Для основы разработки учебно-практического комплекса было взято среднестатистическое содержание учебной дисциплины «Автоматизация машинного обучения». Оно включает следующие темы: вводная часть; непрерывная интеграция, доставка и обучение (CI/CD/CT) в проектах ML; виртуализация и контейнеризация; управление данными; конвейеры операций; автоматизация тестирования; запуск проекта в промышленном окружении; подведение итогов и оценка полученных навыков.

Приведенные темы позволят слушателям курса «Автоматизация машинного обучения» приобрести теоретические знания, а также практические компетенции, востребованные для специалистов в данной области.

Так, вводная часть позволит слушателям изучить историю появления методологии MLOps, понять причину ее востребованности. Также слушатели в данном разделе ознакомятся с терминологией MLOps, базовыми понятиями и методами автоматизации машинного обучения, что характерно для всех вводных модулей по различным учебным дисциплинам. В этом же разделе они изучат важные аспекты управления проектами машинного обучения: жизненный цикл проекта, состав команды и роли участников проекта и познакомятся с наиболее популярными инструментами для решения задач MLOps. Именно вводная часть курса позволит слушателям установить важную взаимосвязь между техническими требованиями к проекту машинного обучения и содержанием задач MLOps. С практической стороны вводная часть познакомит слушателей с основами автоматизации конвейера машинного обучения. Еще не ознакомившись с типичными для данных задач технологиями, слушатели смогут понять причины необходимости самой автоматизации процессов, а также постараются самостоятельно реализовать процесс автоматизации с использованием простейших инструментов, что позволит им глубже разобраться в методах автоматизации.

Тема непрерывной интеграции, доставки и обучения (CI/CD/CT) в проектах ML дает понимание слушателям, что именно необходимо автоматизировать в процессе машинного обучения. В данном разделе слушатели узнают основные задачи и инструменты, используемые для CI/CD/CT для проектов машинного обучения. Так, слушатели ознакомятся с понятиями автоматизации, непрерывной интеграции, развертывания и тестирования проектов. Изучат задачи процесса автоматизации и ознакомятся с используемыми в данной сфере актуальными инструментами, в том числе опробуют их использование на практике. Более подробное знакомство

происходит с инструментом Jenkins. Практическая часть модуля построена на разработке собственного конвейера автоматизации машинного обучения, но уже с использованием специализированного инструментария, а именно, Jenkins. Так, слушатели приобретут практический навык создания конвейера автоматизации машинного обучения с использованием специальных инструментов, что, безусловно, является одним из основополагающих умений инженера по автоматизации машинного обучения.

Виртуализация и контейнеризация — пожалуй, одни из основных технологий на сегодняшний день, используемые при разработке и внедрении ПО, тем более в контексте проектов машинного обучения. Этот раздел позволит слушателям ознакомиться с основными технологиями и инструментами создания изолированных сред выполнения программ, а также научит применять эти знания для решения задач MLOps. В данном модуле слушатели поймут, зачем необходимы изолированные среды выполнения программ и какие есть подходы для их реализации. Познакомятся с такими понятиями, как виртуальная машина, виртуальная среда, контейнер, а также с практическим применением этих технологий в контексте MLOps. В практической части данного модуля слушатели освоят работу технологиями с docker и docker-compose, смогут на их основе развернуть собственный микросервис. В условиях современных тенденций деления масштабных сервисов приложений на приложения с микросервисной архитектурой практическая работа данного модуля является инструментом приобретения крайне актуального и востребованного навыка слушателями — навыка создания и развертывания микросервисов. Также это позволит слушателям даже в рамках учебной дисциплины автоматизации машинного обучения создавать проекты с более лаконичной и продуманной архитектурой.

Модуль по управлению данными призван научить слушателей формулировать задачи для каждого этапа обработки данных в проекте машинного обучения, выбирать инструменты для решения этих задач, а также применять существующие инструменты для решения практических задач для

управления данными. Слушатели изучат важнейшие основы работы с данными, необходимые для реализации процессов машинного обучения: освоят методологию постановки задач инженерии данных, разберутся в этапах обработки данных, смогут понять процесс управления данными с использованием специализированного ПО. Данный модуль подразумевает знакомство с понятиями сбора данных, препроцессинга (предварительной обработки данных), передачи, хранения данных, разведочного анализа данных и выбора признаков для обучения моделей как с теоретической, так и с практической точки зрения. В практической части модуля происходит знакомство с инструментом по управлению данными — dvc. Слушателям предложено будет создать и настроить различные версии собственного датасета с помощью инструмента dvc. Практическая работа подразумевает прохождение всех шагов: от установки dvc в систему до переключения между созданными датасетами. Это знакомит слушателей с современными методами обработки и управления данными, что является неотъемлемой частью сферы машинного обучения.

В теме «конвейеры операций» слушатели изучат понятие конвейера операций и инструменты для его создания. Они научатся декомпозировать процесс работы с моделью машинного обучения на отдельные операции, создавать конвейеры операций с использованием специальных инструментов. Также слушатели изучат и опробуют применение конвейеров операций в эксплуатации моделей машинного обучения, осуществление запуска и мониторинга выполнения данных операций. В модуле будет подробно описана теория конвейеров операций, а в практической части слушателям будет предложена работа с инструментом Apache Airflow. В рамках практики слушатели смогут создать конвейер операций с помощью Apache Airflow, а также решить несколько практических задач по конвейеризации. Данная практическая часть способствует приобретению и закреплению слушателями навыков создания конвейеров операций с помощью специального ПО, т.е. ключевых навыков самой автоматизации в процессе машинного обучения.

Раздел по автоматизации тестирования позволит слушателям узнать о методах проверки качества работы разработанного программного обеспечения. Изучение данного модуля сформирует у слушателей понятие роли процесса тестирования в проекте разработки программного обеспечения как в целом, так и в контексте проектов машинного обучения. Слушатели познакомятся с видами тестирования ПО. Также данный раздел включает обзор средств автоматизации тестирования. Рассматриваются и отличительные особенности тестирования для проектов машинного обучения. На практической части модуля слушатели познакомятся с такими инструментами как unittest и pytest, смогут реализовать собственный процесс тестирования проекта. Тяжело переоценить значимость тестирования разрабатываемого ПО, тем более в процессе непрерывной разработки и внедрения. Данный модуль призван познакомить слушателей с наиболее актуальными методами и инструментами тестирования проектов машинного обучения, а также сформировать практические навыки по автоматизации тестирования с использованием специализированных инструментов и устранения, и проработки выявленных недостатков ПО.

В разделе, посвященном запуску проекта в промышленном окружении, слушатели смогут закрепить, обобщить и подытожить полученные в предыдущих модулях знания и навыки. Реализация и запуск проекта — основная цель команд разработчиков ПО. Поэтому умение грамотно подготовить среду и запустить разработанный проект — это основной навык и критерий квалификации инженера по автоматизации разработки ПО, в данном случае — в контексте машинного обучения. Данный модуль позволит слушателям научиться планировать архитектуру и создавать инфраструктуру для всех этапов и задач проекта машинного обучения, начиная от сбора данных и заканчивая выводом решения в производственное окружение и его эксплуатацией. Все полученные в предыдущих разделах теоретические знания будут закреплены, а в практической части модуля слушатели создадут собственную полноценную программную систему, включающую модель

машинного обучения, и с использованием изученных средств автоматизации развернут данную модель в производственном окружении.

2.2 Определение состава разрабатываемого комплекса

Любой учебно-практический комплекс включает задания различных типов. Разрабатываемый комплекс — не исключение. В нем предполагается использование тестовых заданий с единственным выбором ответа, с множественных выбором, задания на соответствие. Такого типа задания предназначены для оценки усвоения теоретического материала студентами. Для формирования практических навыков в комплексе предназначены практические и домашние задания. Они включают работу студентов непосредственно с инструментами и технологиями автоматизации машинного обучения.

Так, первый модуль посвящен знакомству с дисциплиной MLOps, техническим обеспечением и с основами автоматизации конвейера машинного обучения. Данный модуль разделен на восемь юнитов, каждый юнит завершается проверочными тестовыми заданиями. В конце каждого модуля идет практическое задание.

Для контроля усвоения теоретического материала были предложены тестовые задания, включаемые в курс по мере раскрытия тем. Ниже будут приведены примеры тестовых заданий для некоторых юнитов. Так, для юнита знакомства со сферой MLOps предложено несколько тестовых вопросов:

– на выбор единственного ответа на тему расшифровки аббревиатуры «MLOps»;

– на множественный выбор по тематике рисков разработки, связанных с машинным обучением;

– на расположение элементов по порядку из материала по этапам машинного обучения;

– на сопоставление элементов, касающихся навыков инженера MLOps. [16]

Примеры тестовых заданий:

1) Что содержит аббревиатура MLOps

a) Machine Learning + Operations,

b) Machine Learning + Opinions,

c) Maximum Learning + Operations,

d) Machine Lean + Operations,

2) Отметьте причины, по которым в проектах разработки программного обеспечения добавились дополнительные риски, касающиеся машинного обучения

a) чувствительность модели машинного обучения к любому шуму в данных,

b) малое количество статей и готовых решений,

c) невозможность долговременного сохранения окружения, в котором эксплуатируется модель машинного обучения,

d) использование сложных математических алгоритмов,

3) Расположите этапы проекта машинного обучения в правильном хронологическом порядке

a) сбор данных,

b) преобразование данных,

c) тренировка модели машинного обучения,

d) вывод в эксплуатацию,

e) предсказание на промышленных данных,

f) оценка качества работы,

4) Поставьте в соответствие навыки, которыми должен обладать MLOps, и причины важности обладания этими навыками. Пример таблица 4.

Таблица 4 — Пример теста на соответствие

Навык	Причина
Программирование на C++	Основной код разрабатываемой системы написан на этом языке
Администрирование linux	Большинство инструментов, которыми пользуется команда, реализованы этими средствами
SQL запросы	Для организации сбора данных
bash, python	Для разработки скриптов автоматизации процедур
Машинное обучение, метрики	Для своевременного отслеживания ухудшения качества работы системы

Данные задания реализуются на выбранной LMS как задания тестового типа. Предлагаются в качестве тестов для самоконтроля после освоения студентами текстового материала. Оцениваются по шкале от 0 до 1 — за каждый вопрос, весь тест имеет максимальный балл 4. Примеры реализации представлены на рисунке 1 и рисунке 2.

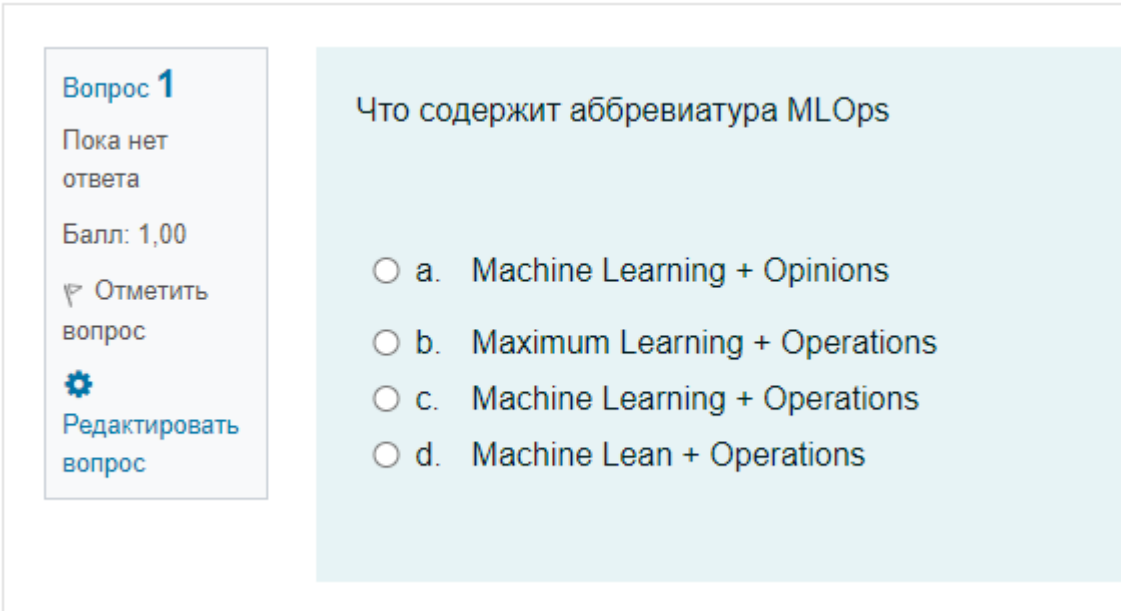


Рисунок 1 — Пример тестового задания с одним вариантом ответа

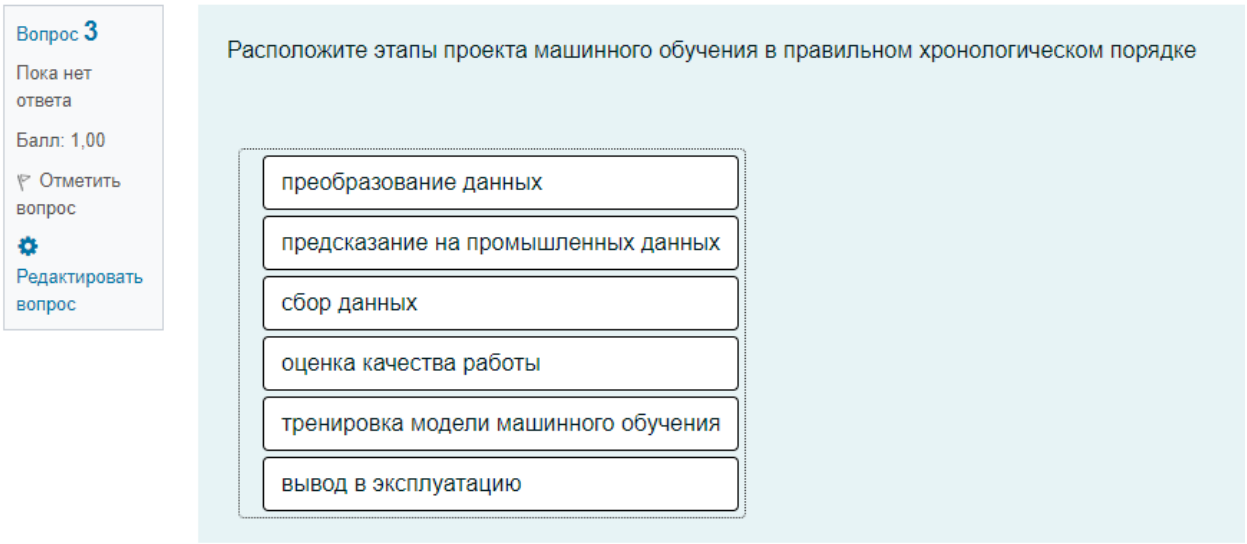


Рисунок 2 — Пример тестового задания на расположение этапов в правильном порядке

Далее следует юнит «Составляющие элементы MLOps». Он посвящен идеологии MLOps и составным этапам автоматизации машинного обучения. Предложены тестовые задания по теоретическому материалу:

– с единственным вариантом ответа на понимание структуры MLOps;

– с единственным вариантом ответа на запоминание терминов в сфере MLOps;

– на множественный выбор на тему предметов процесса автоматизации машинного обучения;

– на множественный выбор на понимание проблем MLOps.

Примеры тестовых заданий:

1) Что символизирует символ «бесконечность» на рисунке, описывающем процесс DevOps

a) то, что проблем в DevOps/MLOps бесконечно много,

b) то, что итерации в процессе DevOps/MLOps должны быстро и много повторяться,

c) то, что в DevOps/MLOps используется бесконечно много разных инструментов,

d) то, что расходы на DevOps/MLOps неограничены,

2) Что такое артефакт этапа?

a) документ с перечнем ошибок,

b) список задач этапа,

c) результат выполнения этапа, который передается на следующий этап,

d) недокументированные возможности работы системы,

3) Что можно автоматизировать в проекте машинного обучения?

a) подготовку технического задания,

b) сбор данных,

c) обучение модели машинного обучения,

d) вывод системы в продакшн,

4) Какие проблемы может решить MLOps?

a) человеческие ошибки при повторении большого количества одинаковых рутинных операций,

b) высокая стоимость вычислительных ресурсов,

c) ошибки, связанные с несовместимостью различных версий библиотек, использованных в решении,

d) некорректное техническое задание,

Данные задания реализуются на выбранной LMS как задания тестового типа. Предлагаются в качестве тестов для самоконтроля после освоения студентами текстового материала. Оцениваются по шкале от 0 до 1 — за каждый вопрос, весь тест имеет максимальный балл 4. Примеры реализации представлены на рисунке 3 и рисунке 4.

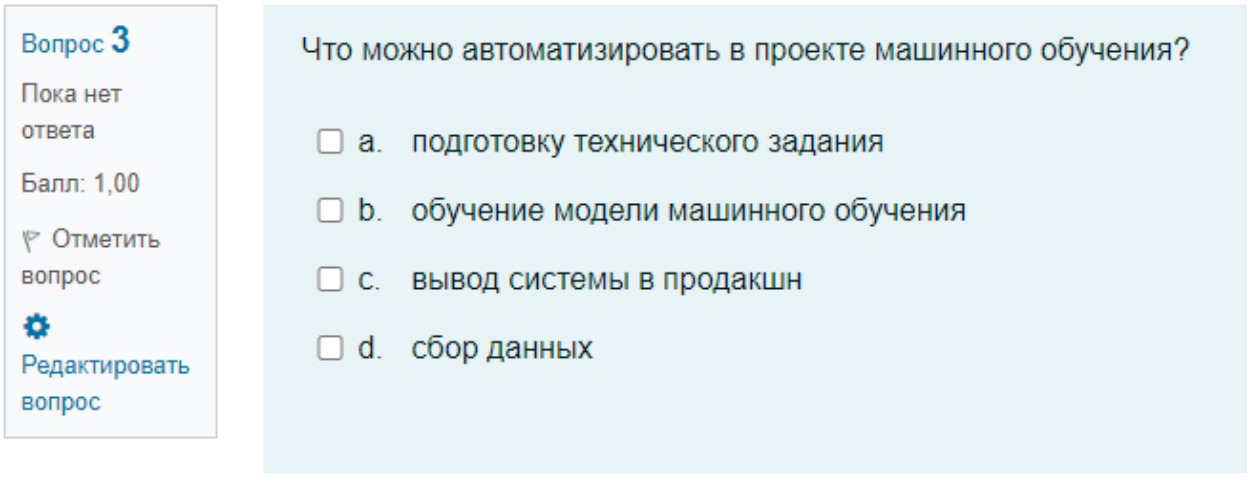


Рисунок 3 — Пример тестового задания с множественным выбором варианта ответа

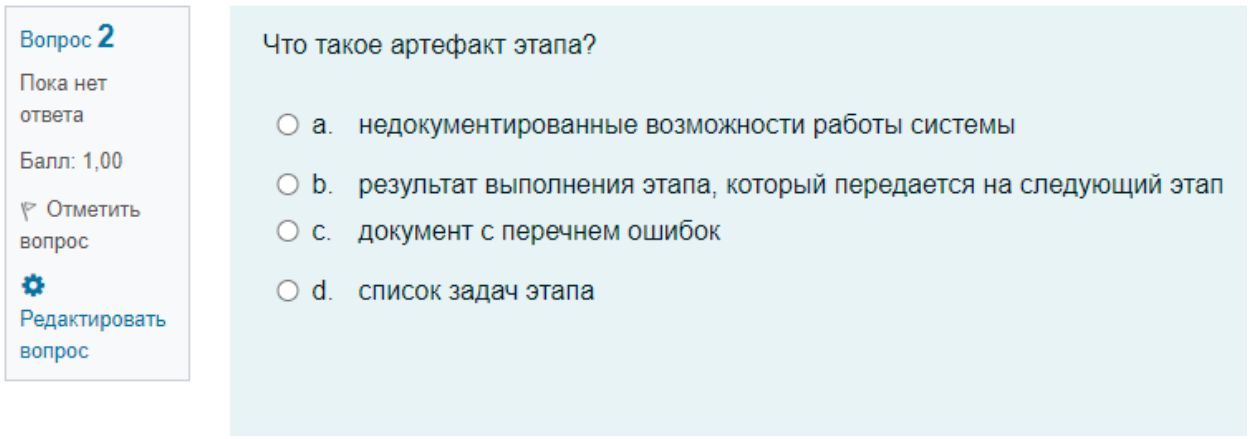


Рисунок 4 — Пример тестового задания с единственным вариантом ответа

Далее в курсе раскрывается юнит «Управление проектами MLOps». В нем речь идет о методологиях управления проектами, способах повышения эффективности автоматизации и обеспечения непрерывности процесса. Для оценки понимания теоретического материала предложены тестовые задания:

– с единственным вариантом ответа по теме технического долга;

– с единственным вариантом ответа по теме «версионирования»;

– с множественным выбором по определению задач MLOps в процессе разработки;

– на соответствие на понимание задач, решаемых с помощью визуализации.

Примеры тестовых заданий:

1) что такое «технический долг»?

a) задолженность исполнителя перед заказчиком,

b) задолженность заказчика перед исполнителем,

c) перечень функций, которые необходимо реализовать, чтобы получить приемлемое для заказчика качество продукта,

d) список ошибок, найденных при тестировании,

2) что такое «версионирование»?

a) контроль версий,

b) правила нумерации датасетов,

c) правила нумерации всех объектов,

d) формат номера,

3) каких задач MLOps нет в обычном проекте разработки

a) версионирование датасетов,

b) версионирование пайплайнов,

c) управление изменениями в коде,

d) настройка виртуального окружения,

4) поставьте в соответствие этапам задачи, которые решаются с помощью визуализации. Пример в таблице 5.

Таблица 5 — Пример теста на соответствие

Анализ данных	обнаружение ошибок в данных, выявление характерных статистических признаков
Построение моделей	контроль переобучения
Эксплуатация	контроль качества работы решения, обнаружение отклонений в работе
Анализ результатов проекта	создание наглядных интерпретируемых графиков для презентации результатов заказчику

Данные задания реализуются на выбранной LMS как задания тестового типа. Предлагаются в качестве тестов для самоконтроля после освоения студентами текстового материала. Оцениваются по шкале от 0 до 1 — за каждый вопрос, весь тест имеет максимальный балл 4. Примеры реализации представлены на рисунке 5 и рисунке 6.

Рисунок 5 — Пример тестового задания с множественным выбором варианта ответа

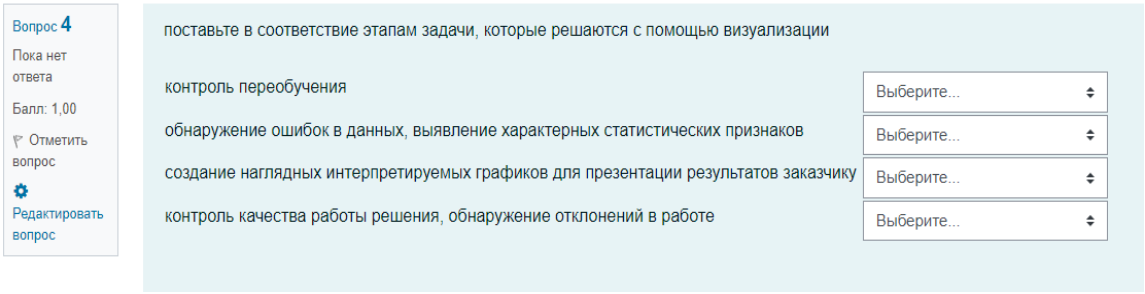
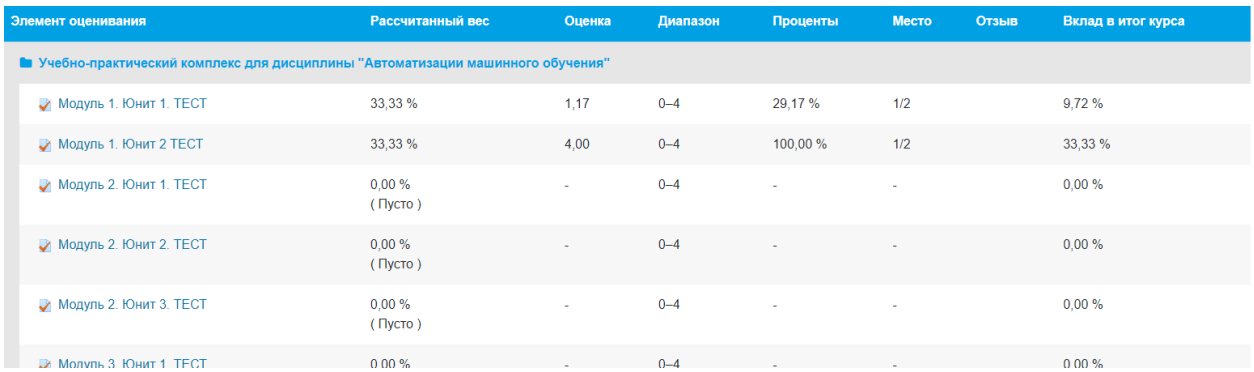


Рисунок 6 — Пример тестового задания на соответствие

Все юниты, которые были описаны выше, относятся к одному модулю.

Проверка тестовых заданий для каждого юнита осуществляется с помощью средств LMS платформы. Пример представлен ниже на рисунке 7.



Элемент оценивания	Рассчитанный вес	Оценка	Диапазон	Проценты	Место	Отзыв	Вклад в итог курса
Учебно-практический комплекс для дисциплины "Автоматизации машинного обучения"							
Модуль 1. Юнит 1. ТЕСТ	33,33 %	1,17	0–4	29,17 %	1/2		9,72 %
Модуль 1. Юнит 2 ТЕСТ	33,33 %	4,00	0–4	100,00 %	1/2		33,33 %
Модуль 2. Юнит 1. ТЕСТ	0,00 % (Пусто)	-	0–4	-	-		0,00 %
Модуль 2. Юнит 2. ТЕСТ	0,00 % (Пусто)	-	0–4	-	-		0,00 %
Модуль 2. Юнит 3. ТЕСТ	0,00 % (Пусто)	-	0–4	-	-		0,00 %
Модуль 3. Юнит 1. ТЕСТ	0,00 %	-	0–4	-	-		0,00 %

Рисунок 7 — Пример оценки тестовых заданий на LMS платформе

После прохождения всех юнитов следует практическое задание на тему общего модуля.

Пример практического задание представлен ниже:

Цель задания:

В этом задании вы решите простую и во многом учебную задачу по созданию автоматического конвейера проекта машинного обучения. Подобный подход с применением простых скриптов автоматизации для «склейки» отдельных частей конвейера используется в небольших проектах. Чаще же для автоматизации используется специализированное программное обеспечение, например, Jenkins.

Содержание задания:

Необходимо из «подручных средств» создать простейший конвейер для автоматизации работы с моделью машинного обучения. Отдельные этапы конвейера машинного обучения описываются в разных python–скриптах, которые потом соединяются (иногда используют термин «склеиваются») с помощью bash-скрипта.

Этапы:

– Создайте python-скрипт (data_creation.py), который создает различные наборы данных, описывающие некий процесс (например, изменение дневной температуры). Таких наборов должно быть несколько, в некоторые данные можно включить аномалии или шумы. Часть наборов данных должны быть сохранены в папке «train», другая часть в папке «test».

– создайте python-скрипт (model_preprocessing.py), который выполняет предобработку данных, например, с помощью sklearn.preprocessing.StandardScaler.

– создайте python-скрипт (model_preparation.py), который создает и обучает модель машинного обучения на построенных данных из папки «train».

– создайте python-скрипт (model_testing.py), проверяющий модель машинного обучения на построенных данных из папки «test».

– Напишите bash-скрипт (pipeline.sh), последовательно запускающий все python-скрипты.

Подготовленные скрипты необходимо опубликовать в git репозитории, ссылку на который необходимо предоставить как результат выполнения задания.

После прохождения практического задания по первому модулю, идет следующий модуль

Модуль два имеет название «Непрерывная интеграция, доставка и обучение (CI/CD/CT) в проектах ML». Данный модуль разделен на пять юнитов в которых раскрываются важные темы непрерывной интеграции.

Для контроля усвоение теоретического материала данного модуля были предложены тестовые задания, включаемые в курс по мере раскрытия тем. Тестовые задание расположены в конце каждого юните, как и в предыдущем модуле. Ниже будут рассмотрены некоторые тестовые задания из нескольких юнитов.

Для юнита Автоматизация. Непрерывная интеграция, развертывание и тестирование. В котором говориться о задачах CI/CD/CT. Отдельно более подробно разобрана наиболее важная задача: автоматизация. Для данного юнита были предложены тестовые вопросы: [18]

– На выбор единственного ответа на расшифровку аббревиатуры CT

– На выбор единственного ответа, что такое непрерывная интеграция

– Установите соответствие между CI/CD/CT и задачами

– на множественный выбор по, что может быть триггером для запуска процесса сборки программного пакета в CI

Примеры тестов представлены ниже:

1) Что такое непрерывная интеграция

a) Непрерывно работающий сервис по сбору данных от микросервисов,

b) Автоматизированный процесс сборки общего программного решения из отдельных элементов,

c) Постоянный мониторинг всех процессов,

d) Ручная сборка программного пакета,

2) Как расшифровывается CT

a) Common tools,

b) Continuous training,

c) Continuous testing,

d) Cost transfer,

3) Установите соответствие между CI/CD/CT и задачами. Пример в таблице 6.

Таблица 6 — Пример теста на соответствие

Автоматический запуск тестов	CI, CT
Настройка окружения для запуска программы	CD
Проверка изменений в программном коде	CI
Создание ролей пользователя	CD

4) Что может быть триггером для запуска процесса сборки программного пакета в CI

a) Публикация изменений кода в репозитории проекта,

b) Расписание,

c) Ручной запуск,

d) Письмо от заказчика с негативным отзывом о работе системы,

Данные задания реализуются на выбранной LMS как задания тестового типа. Предлагаются в качестве тестов для самоконтроля после освоения студентами текстового материала. Оцениваются по шкале от 0 до 1 — за каждый вопрос, весь тест имеет максимальный балл 4. Примеры реализации представлены на рисунке 7 и рисунке 8.

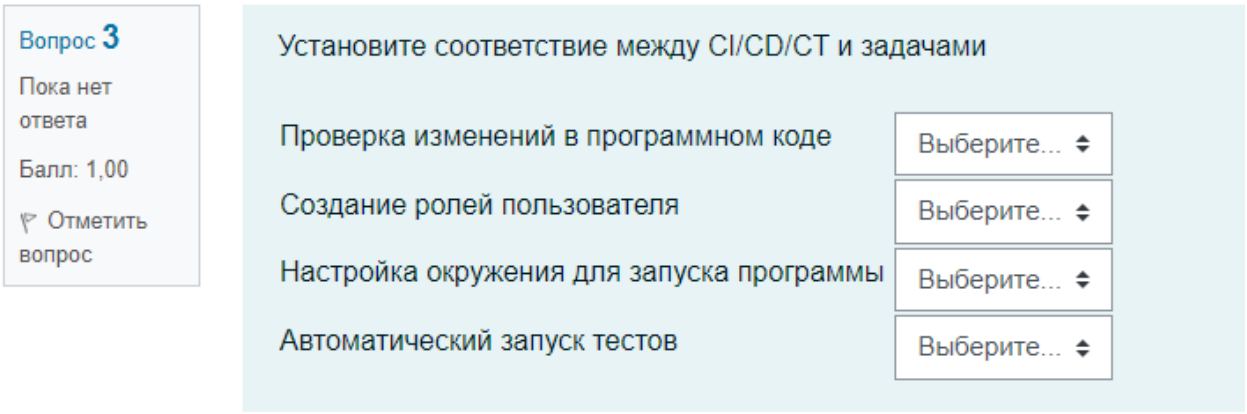


Рисунок 8 — Пример тестового задания на соответствие

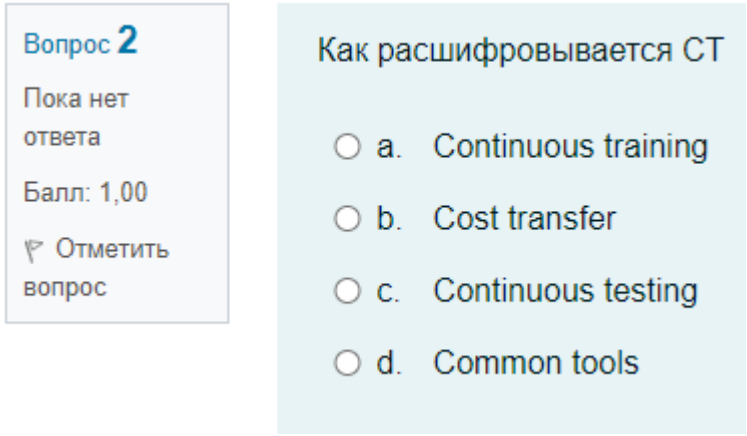


Рисунок 9 — Пример тестового задания с единственным вариантом ответа

Следующим был рассмотрен юнит Задачи и инструменты CI/CD/CT в проектах машинного обучения, в котором говориться о рассмотрим особенности CI/CD/CT в проектах машинного обучения, отличающие их от обычных проектов разработки программного обеспечения. По данному юниту была предложена реализация тестов:

– на множественный выбор на тему, что отличает MLOps CI от обычного проекта

– на множественный выбор на тему, что отличает MLOps CD от обычного проекта

– с единственным вариантом ответа на понимание термина CT

– на множественный выбор на тему какие параметры важно поставить под мониторинг при развертывании и эксплуатации модели машинного обучения

Примеры тестов представлены ниже:

1) Что отличает MLOps CI от обычного проекта

a) Наличие датасетов,

b) Наличие моделей машинного обучения,

c) Наличие комментариев в коде,

d) Повышенные требования к производительности системы,

2) Что отличает MLOps CD от обычного проекта

a) Необходимость более тщательно настраивать среду выполнения программы для более качественной работы модели (инференса),

b) Необходимость проверять качество данных, поступающих при эксплуатации системы, на которых работает модель,

c) Повышенные требования к работе аппаратного обеспечения,

d) Большее количество ролей участников в проекте,

3) Какой еще смысл возникает у термина CT, который означает Continuous Testing, применительно к проектам машинного обучения,

a) Common Thinking,

b) Collaborative Testing,

c) Continuous Training,

d) Нет никаких изменений у термина CT,

4) Какие параметры важно поставить под мониторинг при развертывании и эксплуатации модели машинного обучения

a) Оперативную память системы,

b) Скорость работы модели,

c) Качество поступающих данных,

d) Температуру окружающей среды,

Данные задания реализуются на выбранной LMS как задания тестового типа. Предлагаются в качестве тестов для самоконтроля после освоения студентами текстового материала. Оцениваются по шкале от 0 до 1 — за каждый вопрос, весь тест имеет максимальный балл 4. Примеры реализации представлены на рисунке 10.

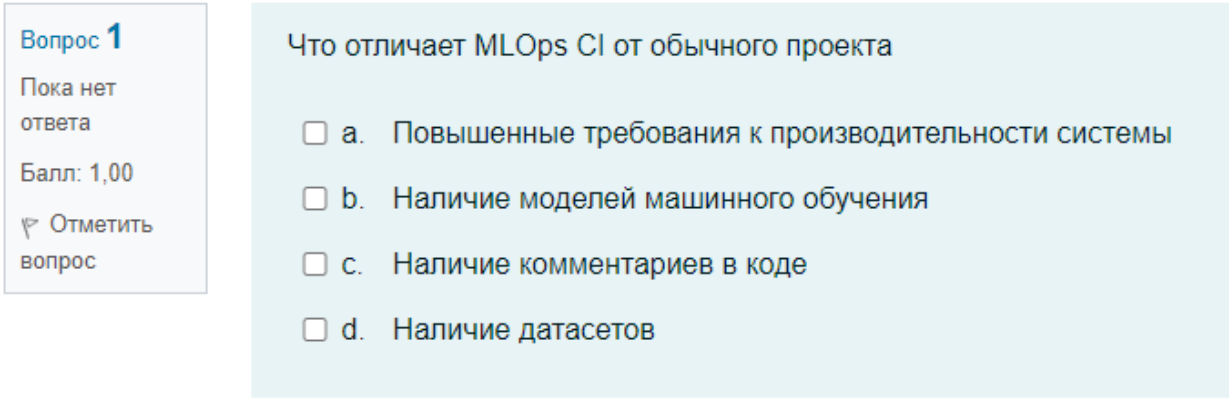


Рисунок 10 — Пример тестового задания с множественным выбором варианта ответа

Следующим идет юнит знакомство с Jenkins. Установка. В котором было рассмотрено приложение Jenkins и подробное описание установки данного приложения.

Для данного юнита были предложены следующие тесты:

– с единственным вариантом ответа на понимание работы приложения Jenkins

– на множественный выбор на понимание работы приложения Jenkins

Ниже представлены примеры тестов [19]:

1) Для автоматизации каких задач предназначен Jenkins?

a) публикация изменений кода в репозиторий проекта,

b) разработка программного обеспечения,

c) тестирование изменений,

d) доставка системы в среду выполнения,

2) Какая библиотека необходима для работы Jenkins?

a) Python,

b) Java 1,

c) Java 8,

d) jenkins-lib,

3) Какой командой в linux можно проверить статус работы сервиса Jeknins?

a) get status jenkins,

b) sudo service jenkins status,

c) select jenkins status,

d) which status for jenkins,

4) На каком номере порта запускается web-интерфейс Jenkins?

a) 1,

b) 80,

c) 8080,

Данные задания реализуются на выбранной LMS как задания тестового типа. Предлагаются в качестве тестов для самоконтроля после освоения студентами текстового материала. Оцениваются по шкале от 0 до 1 — за каждый вопрос, весь тест имеет максимальный балл 4. Примеры реализации представлены на рисунке 11 и рисунке 12.

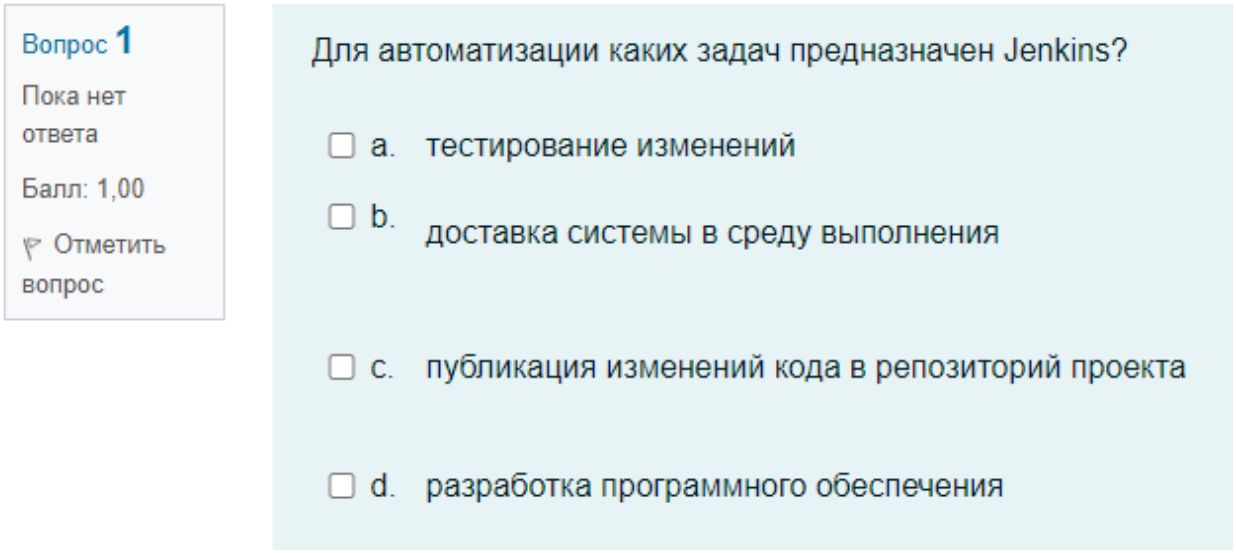


Рисунок 11 — Пример тестового задания с множественным выбором варианта ответа

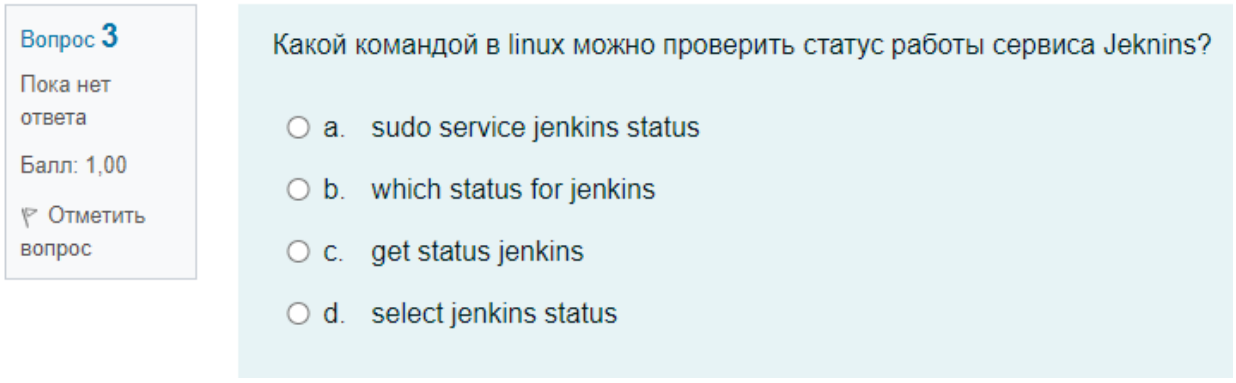


Рисунок 12 — Пример тестового задания с единственным вариантом ответа

Проверка тестовых заданий для каждого юнита осуществляется с помощью средств LMS платформы. Пример представлен ниже на рисунке 13.

Рисунок 13 — Пример проверки оценки пройдённых тестов

По завершению всех юнитов в данном модуле следует практическое задание.

Данное практическое задание представлено ниже

В практическом задании по этому модулю вам нужно разработать собственный конвейер автоматизации для проекта машинного обучения, аналогичный тому, который мы рассмотрели в последнем юните этого модуля. Для этого вам понадобится виртуальная машина с установленным Jenkins, python и необходимыми библиотеками. В ходе выполнения практического задания вам необходимо автоматизировать сбор данных, подготовку датасета, обучение модели и работу модели.

Этапы задания

1) Развернуть сервер с Jenkins, установить необходимое программное обеспечение для работы над созданием модели машинного обучения,

2) Выбрать способ получения данных (скачать из github, из Интернета, wget, SQL запрос, …),

3) Провести обработку данных, выделить важные признаки, сформировать датасеты для тренировки и тестирования модели, сохранить,

4) Создать и обучить на тренировочном датасете модель машинного обучения, сохранить в pickle или аналогичном формате,

5) Загрузить сохраненную моде на о проанализировать ее качество на тестовых данных,

Критерии выполнения задания:

1) установлен и настроен Jenkins,

2) написан корректно работающий job в Jenkins, решающий задачу,

3) получены данные о качестве работы модели машинного обучения,

После проверки практики будет доступен следующий модуль, который имеет название Виртуализация и контейнеризация. Модуль состоит из шести юнитов, в каждом юните рассмотрена отдельная тема, в конце каждого юнита сделан тест на проверку лекционного материала. В конце всего модуля предоставлено практическое задание.

Ниже представлены примеры тестов для юнитов и пример практического задания.

Первым будет рассмотрен юнит Различные подходы к созданию изолированных программных сред. В данном юните рассматриваются различные подходы к созданию виртуальных изолированных сред для выполнения программ. Для данного юнита были предложены следующие варианты тестов на множественный выбор на тему виртуализация и контейнеризация [20].

Пример тестов представлен ниже:

1) Отметьте гипервизоры

a) Hyper-V,

b) Docker,

c) KVM,

d) VMWare,

2) Отметьте инструменты для контейнеризации

a) Docker,

b) Kubernetes,

c) Containers,

d) VMWare,

3) Отметьте задачи, которые решает виртуализация в проектах машинного обучения

a) дает возможность делать качественные визуализации,

b) ускоряет развертывание проекта с заданными характеристиками,

c) обеспечивает повторяемость результатов работы модели,

d) позволяет обрабатывать большие объемы информации,

4) Отметьте положительные стороны использования виртуальных машин

a) возможность создания и использования унифицированных типовых конфигураций и политик для виртуальных машин,

b) существенное ускорение производительности оборудования,

c) оптимизация расходов на аппаратное обеспечение,

d) гарантия качества работы виртуальной машины компанией-разработчиком,

Данные задания реализуются на выбранной LMS как задания тестового типа. Предлагаются в качестве тестов для самоконтроля после освоения студентами текстового материала. Оцениваются по шкале от 0 до 1 — за каждый вопрос, весь тест имеет максимальный балл 4. Примеры реализации представлены на рисунке 14 и рисунке 15.

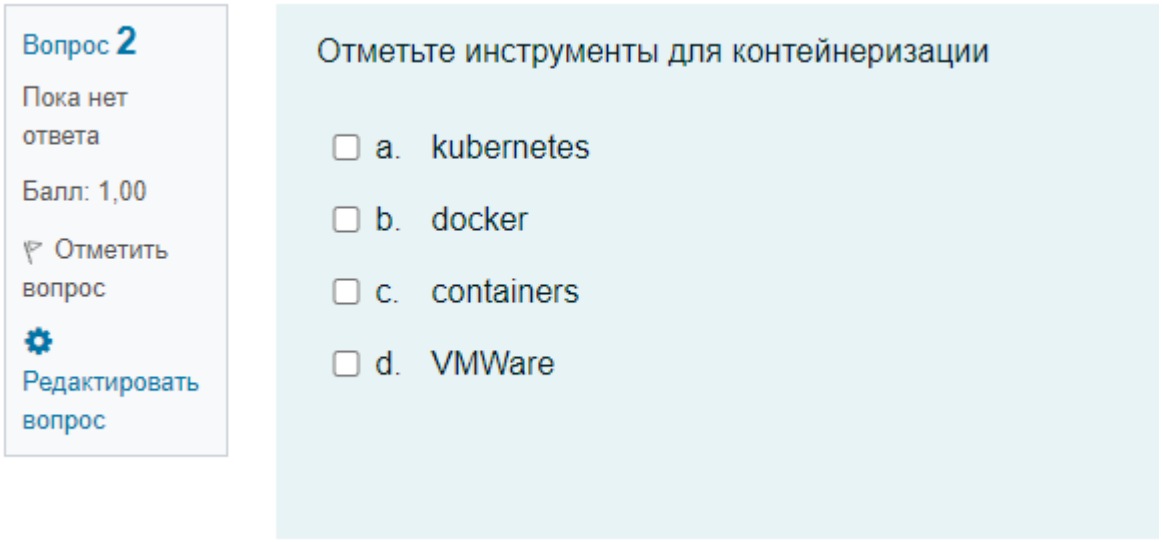


Рисунок 14 — Пример тестового задания с множественным выбором варианта ответа

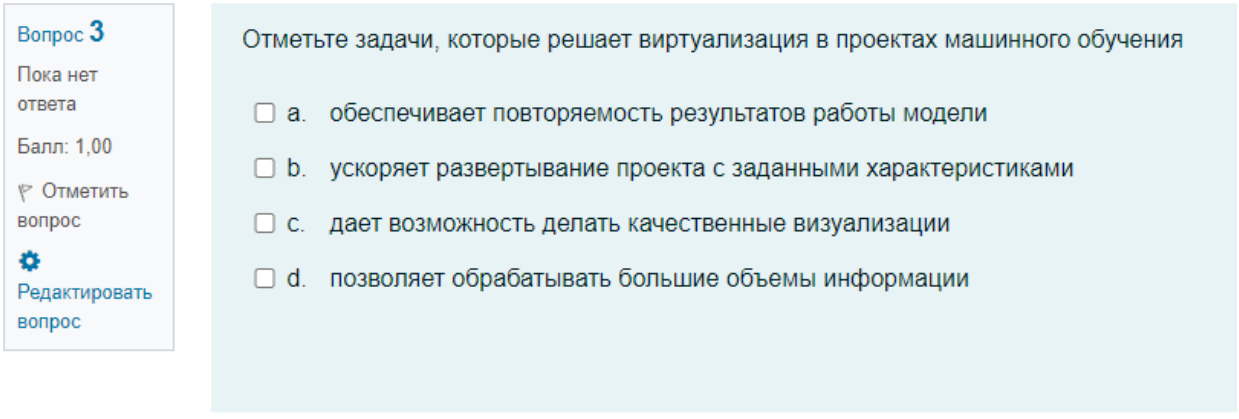


Рисунок 15 — Пример тестового задания с множественным выбором варианта ответа

Следующий юнит, который будет рассмотрен Виртуализация с использованием VirtualBox. В данном юните рассматривается задача виртуализации с использованием популярного бесплатного инструмента VirtualBox компании Oracle. С использованием простых шагов, подробно описанных в этом юните, вы сможете самостоятельно создавать виртуальные машины. В качестве проверки лекционного материала к данному юниту были предложены следующие тесты с единственным вариантом ответа на тему работа с виртуальной машиной.

Примеры тестов предоставлены ниже:

1) Как можно получить официальный дистрибутив VirtualBox?

a) скачать установщик с сайта производителя,

b) использовать утилиту pip install,

c) использовать консольную команду virtualbox install,

d) эта утилита входит по умолчанию во все стандартные операционные системы, ее не надо скачивать и устанавливать,

2) сколько виртуальных сетевых адаптеров может быть связано с виртуальной машиной в VirtualBox?

a) 1,

b) 2,

c) 3,

d) 4,

3) сколько операционных систем может работать в виртуальной машине?

a) 1,

b) 2,

c) 3,

d) 4,

4) как вернуть перехваченное управление клавиатурой и мышью из гостевой операционной системы обратно в операционную систему хоста?

a) написать письмо в техническую поддержку производителя,

b) нажать правую клавишу ctrl,

c) перезагрузить компьютер,

d) нажать ctrl+alt+del,

Данные задания реализуются на выбранной LMS как задания тестового типа. Предлагаются в качестве тестов для самоконтроля после освоения студентами текстового материала. Оцениваются по шкале от 0 до 1 — за каждый вопрос, весь тест имеет максимальный балл 4. Примеры реализации представлены на рисунке 16 и рисунке 17.

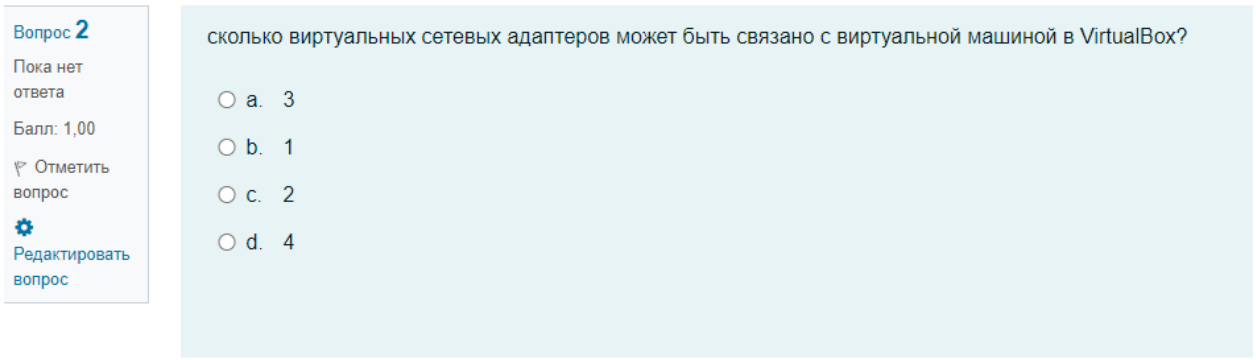


Рисунок 16 — Примет тестового задания с единственным вариантом ответа

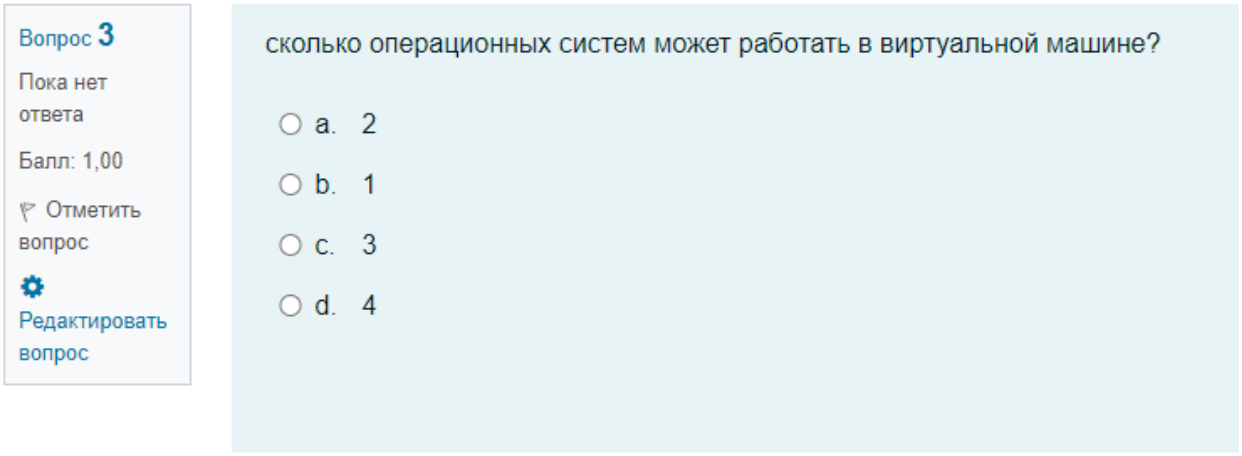


Рисунок 17 — Пример тестового задания с единственным вариантом ответа

Далее следует юнит Виртуальные окружения. В данном юните рассматривается какие задачи решают виртуализация и контейнеризация. Для

данного юнита в качестве проверки лекционного материала были предложены следующий варианты тестовых задний [21]:

– с единственным вариантом ответа на тему виртуальное окружение

– на множественный выбор вариантов ответа на тему виртуальное окружение

Примеры тестовых заданий предоставлены ниже:

1) Отметьте утилиты для создания виртуальных окружений
   a. virtual-environment,
   b. venv,
   c. windows,
   d. poetry,
2) Какой командой запускается виртуальное окружение?
   a. start env,
   b. locate /bin/activate,
   c. run env,
   d. activate env,
3) Какой программой остановить виртуальное окружение?
   a. Stop,
   b. Release,
   c. Deactivate,
   d. Exit,
4) Какая команда poetry добавить библиотеку numpy в проект?
   a. poetry add numpy,
   b. poetry install numpy,
   c. poetry load numpy,
   d. poetry get numpy,

Данные задания реализуются на выбранной LMS как задания тестового типа. Предлагаются в качестве тестов для самоконтроля после освоения студентами текстового материала. Оцениваются по шкале от 0 до 1 — за

каждый вопрос, весь тест имеет максимальный балл 4. Примеры реализации представлены на рисунке 18 и рисунке 19.

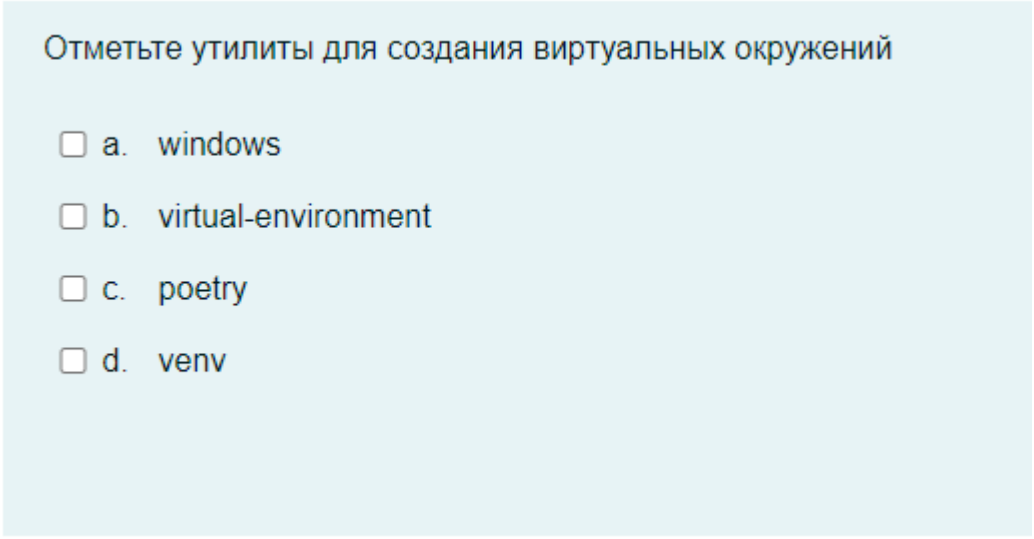


Рисунок 18 — Пример тестового задания с множественным выбором варианта ответа

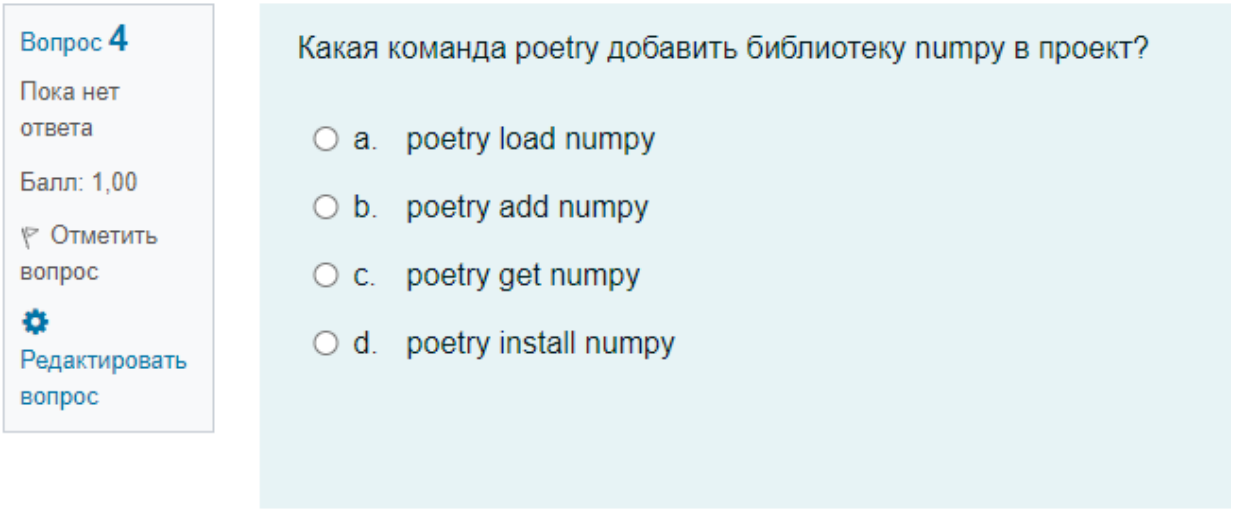


Рисунок 19 — Пример тестового задания с единственным вариантом ответа

Проверка тестовых заданий для каждого юнита осуществляется с помощью средств LMS платформы. Пример представлен ниже на рисунке 20.

Модуль 3. Юнит 1. ТЕСТ	11,11 %	4,00	0–4	100,00 %	1/2	11,11 %
Модуль 3. Юнит 2. ТЕСТ	11,11 %	4,00	0–4	100,00 %	1/2	11,11 %
Модуль 3. Юнит 3. ТЕСТ	11,11 %	4,00	0–4	100,00 %	1/2	11,11 %

Рисунок 20 — Пример проверки оценки пройдённых тестов

После прохождения всех юнитов данного модуля, предлагается выполнить практическое задание, пример практического задание предоставлен ниже

В качестве образца создайте docker образ для модели машинного обучения, в котором обучается модель логистической регрессии для предсказания типа цветка ириса из датасета sklearn.datasets.iris и возвращает предсказание класса на каком-то тестовом примере, например, [1, 1, 1, 1]. Следует выполнить следующие шаги:

1) Подготовить python код для модели,
2) Создать Docker file,
3) Создать Docker образ,
4) Запустить Docker контейнер,

Далее практическое задание проверяется.

После выполнения практического задание открывается доступ к следующему модулю.

Следующий модуль, который рассматривается в курсе это модуль Управление данными. В данном модуле три юнита в конце каждого юните предусмотрено тестовое задание на проверку лекционного материла, по прохождению всех юнитов предоставляет практическое задание. Ниже представлены примеры юнитов с тестовыми задания и пример практического задания.

В качестве примера был рассмотрен юнит Задачи инженерии данных. В данном юните слушатель знакомится с задачами, которые решает инженер данных (Data Engineer, DataOps), наиболее актуальными проблемами при работе с данными, методами предупреждения или решения этих проблем,

базовым инструментарием. В качестве проверки лекционного материала были предложены некоторые варианты тестов:

– На множественный выбор вариантов ответа на понимание работы машинного обучения;

– с единственным вариантом ответа понимание работы машинного обучения

Примеры тестовых материалов представлены ниже

1. Отметьте причины, по которым в процессе эксплуатации модели машинного обучения происходит незначительное ухудшение качества работы модели.

a. изменяется структура данных, например меняется набор параметров для наблюдения,

b. выход из строя источников данных (датчики, счетчики),

c. кратковременный сбой в работе каналов передачи данных,

d. шум в данных, обусловленный ошибкой датчика

2. Какие изменения в данных приводят к необходимости изменения модели машинного обучения?

a. изменение бизнес-логики процесса,

b. шум в данных,

c. сбой в каналах передачи данных,

d. изменение структуры данных

3. Какие задачи относятся к инженерии данных?

a. сбор данных,

b. анализ и преобразование данных,

c. подбор гиперпараметров модели машинного обучения,

d. организация хранения данных

4. Какие инструменты используются для версионирования.

a. dvc,

b. git,

c. jupyterhub,

d. python,

Данные задания реализуются на выбранной LMS как задания тестового типа. Предлагаются в качестве тестов для самоконтроля после освоения студентами текстового материала. Оцениваются по шкале от 0 до 1 — за каждый вопрос, весь тест имеет максимальный балл 4. Примеры реализации представлен на рисунке 21.

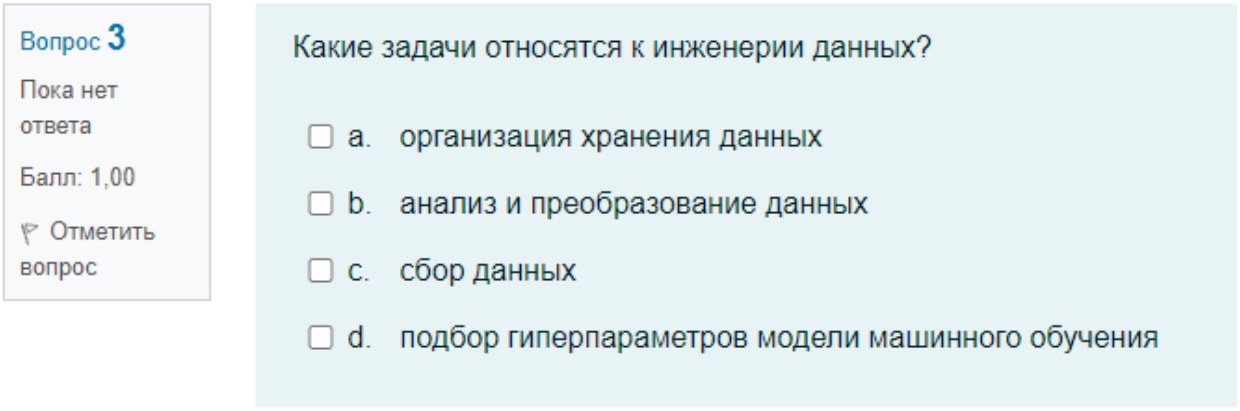


Рисунок 21 — Пример тестового задания на множественный выбор варианта ответа

Проверка тестовых заданий для каждого юнита осуществляется с помощью средств LMS платформы. Пример представлен ниже на рисунке 22.

Рисунок 22 — Пример проверки оценки пройдённых тестов

После прохождения всех юнитов и выполнения всех тестовых заданий, открывается доступ к практическому заданию. Пример данного практического задания предоставлен ниже:

В практическом задании данного модуля вам необходимо продемонстрировать навыки практического использования утилиты dvc для работы с данными. В результате выполнения этих заданий вы выполните все

основные операции с dvc и закрепите полученные теоретические знания практическими действиями.

Этапы задания:

1) Установите git и dvc,

2) Создайте папку проекта,

3) Настройте папку проекта для работы с git и dvc,

4) Настройте git репозиторий,

5) Настройте удаленное хранилище файлов, например на Google Disk,

6) Создайте датасет о пассажирах «Титаника», например, catboost.titanic(),

7) Создайте датасет, в котором содержится информация о классе («Pclass»), поле («Sex») и возрасте («Age») пассажира. Сделайте коммит в git и push в dvc,

8) Создайте новую версию датасета, в котором пропущенные (nan) значения в поле «Age» будут заполнены средним значением. Сделайте коммит в git и push в dvc,

9) Создайте новый признак с использованием one-hot-encoding для строкового признака «Пол» («Sex»). Сделайте коммит в git и push в dvc,

10) Выполните переключение между всеми созданными версиями датасета,

При правильном выполнении задания и вас появится git репозиторий с опубликованной метаинформацией и папка на Google Disk, в которой хранятся различные версии датасетов.

В постановке задачи используется датасет из конкурса «Titanic Disaster», однако вы можете использовать свои наборы данных, в этом случае в п.п.6-9 необходимо использовать информацию и признаки из вашего датасета.

Рекомендация: вы можете пользоваться описанием соответствующих процедур, которые были в юните 3 данного модуля.

В качестве результата необходимо прислать ссылку на репозиторий git.

После выполнения практического задание открывается доступ к следующему модулю данного курса.

Следующий модуль, который рассматривается автоматизация тестирования. состоит из трех юнитов, в качестве примера будут рассмотрены тестовые задания для одного юнита и пример практического задания.

Был рассмотрен юнит виды тестирования. В этом юните рассматривается процесс тестирования, данный процесс являющимся важным этапом любого проекта разработки программного обеспечения. В качестве проверки лекционного материла были предусмотрены тестовые задания данного вида [22]:

– На множественный выбор вариантов ответа на понимание инструментов конвейерной операции;

– с единственным вариантом ответа на понимание работы конвейерной операции

Пример тестовых заданий предоставлен ниже:

1) Укажите виды тестирования

a) Модульное,

b) Корпусное,

c) Интеграционное,

d) Дифференциальное,

2) Что проверяет регрессионное тестирование?

a) Возможность решения задачи с помощью линейной регрессии,

b) Корреляцию между величинами,

c) Отсутствие ранее известных ошибок после внесения изменений,

d) Отсутствие новых ошибок после внесения изменений,

3) Отметьте фреймворки для автоматизации тестов

a) Pytest,

b) Windows,

c) Unittest,

d) Assembler,

4) Что необходимо тестировать в проекте машинного обучения?

a) Модели,

b) Данные,

c) Признаки,

d) Инфраструктуру,

Данные задания реализуются на выбранной LMS как задания тестового типа. Предлагаются в качестве тестов для самоконтроля после освоения студентами текстового материала. Оцениваются по шкале от 0 до 1 — за каждый вопрос, весь тест имеет максимальный балл 4. Примеры реализации представлены на рисунке 23 и рисунке 24.

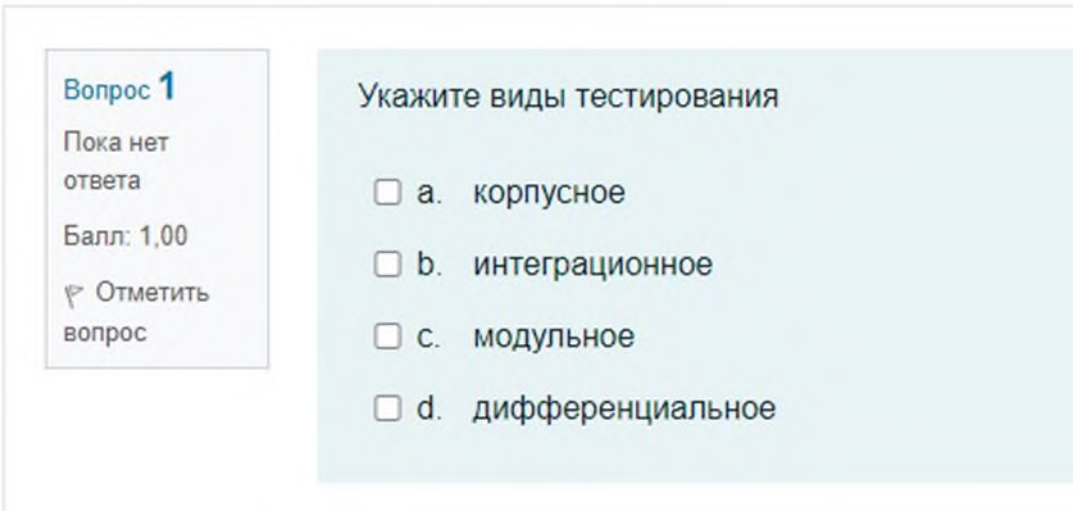


Рисунок 23 — Пример тестового задания на множественный выбор варианта ответа

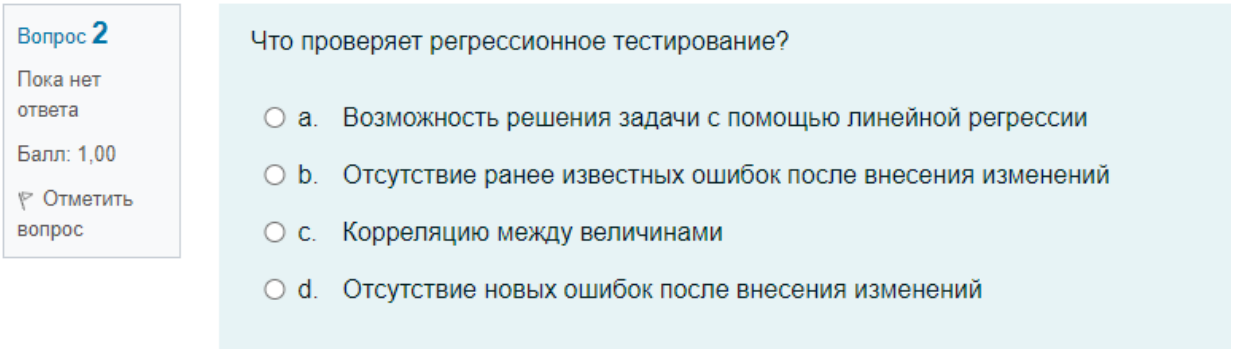


Рисунок 24 — Пример тестового задания на выбор единственного варианта ответа

Проверка тестовых заданий для каждого юнита осуществляется с помощью средств LMS платформы. Пример представлен ниже.

Рисунок 25 — Пример проверки оценки пройдённых тестов

В конце данного модуля предусмотрено практическое задание, пример данного задания представлен ниже.

– Цель задания: применить средства автоматизации тестирования python для автоматического тестирования качества работы модели машинного обучения на различных датасетах.

– Содержание задания:

– Создать три датасета с «качественными» данными, на которых можно обучить простую модель линейной регрессии, например, как на рисунке 26

```
import matplotlib.pyplot as plt
import numpy as np
```

```
xs = np.linspace(0, 10, 100)
ys = xs + np.random.random(100)*2-1

plt.scatter(xs, ys)
plt.show()
```

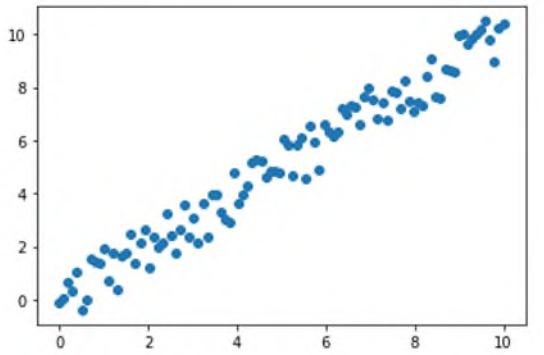


Рисунок 26 — Модель линейной регрессии

На одном из этих датасетов обучить модель линейной регрессии

Создать датасет с шумом в данных, например, как на рисунке 27

```
xs = np.linspace(0, 10, 100)
ys = xs + np.random.random(100)*2-1
ys[25:45] *= 2

plt.scatter(xs, ys)
plt.show()
```

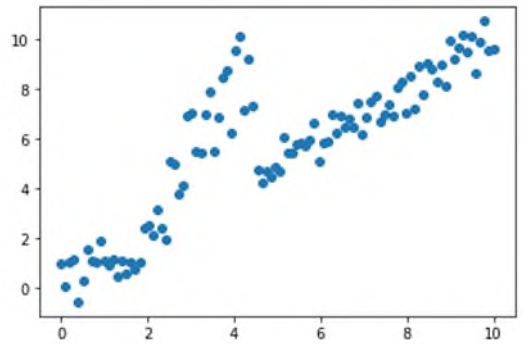


Рисунок 27 — Датасет с шумом

– Провести тестирование работы модели на разных датасетах с использованием pytest, анализируя качество предсказания, обнаружить проблему на датасете с шумами.

– Критерии: данное задание необходимо полностью выполнить в виде jupyter ноутбука и предоставить его на проверку.

– Подсказка: вы можете записать содержимое ячейки jupyter ноутбука в отдельный файл с помощью команды

%%writefile«имя файла»

А также можете выполнить любую linux команду прямо из ячейки jupyter ноутбука, с помощью синтаксиса

! «имя команды»

2.3 Рекомендуемая методика проведения занятий

Рекомендуемой методикой проведения занятий в нашем случае является дистанционное обучение

Под дистанционным обучением понимается применение в педагогическом процессе телекоммуникационных средств, позволяющих педагогу обучать учеников, находясь от них на большом расстоянии. Положительными характеристиками метода являются возможность вовлечения большого числа учащихся, возможность обучения на дому, возможность выбора учащимися наиболее подходящего времени для занятий и возможность переносить результаты процесса обучения на различные электронные носители. Недостатками здесь можно назвать высокие требования к технической оснащённости педагогического процесса, отсутствие визуального контакта педагога и учащегося и, как следствие, пониженную мотивацию со стороны последнего.

Для успешного освоения, разработанного учебно-практический комплекс, рекомендуется проходить его следующим образом. Для более полной картины мы будем рассматривать наш учебно-практический комплекс с условием, что студенты получают информацию в виде визуальных лекции по автоматизации машинного обучения (Визуальные лекции — предусматривают использование специальных виртуальных средств подачи материала. В рамках такого занятия проводится демонстрация видеороликов, фотографий, презентаций, слайдов, работа с онлайн-информацией, комментирование и обсуждение просмотренного материала.) Этот вариант

популярен среди преподавателей и студентов, используется в практике большинства учебных учреждений.

1) Визуальная лекция
2) Тест на усвоение материала
3) Выполнение практического задания.

Сначала идет освоение текстового материала лекции или просмотр если это видео-лекция. Мы будем рассматривать текстовый формат лекций. Лекция разделена на несколько юнитов, в каждом из таких юнитов идет разбор подтемы основной темы лекции. Обозначает цель, что студент должен изучить, в конце юнита идет небольшой вывод о продлённой работе, какие знания были получены, что нового изучил слушатель.

В конце каждого юнита идет тест на усвоение материала. Вопросы составлены из материалов, которые предоставлялись слушателю в текстовой лекции.

Тесты могут иметь вид:

– С выбором одного правильного ответа
– С выбором нескольких правильных ответов
– С формой для ввода ответа с клавиатуры
– На сопоставление слова и его определения.
– На перетаскивание правильных ответов в нужный столбец

Прохождения теста на усвоение слушателем в большинстве случаев позволяет определить на сколько хорошо был освоен материал, который он получил при прочтении информации в юните.

Для перехода к практическому заданию у студентов должно быть выполнено тестовое задание, без этого доступ к прохождению практического задания учебного практического комплекса невозможно.

Практическое задание выделено как отельный юнит при переходе к практическому заданию, вы увидите:

– Цель задания

– Содержание задание

– Этапы выполнения

Рассмотрим каждый из пунктов подробнее [23].

В цели задания, кратко написано, что нужно решить и по какой теме. Так же указано какое программное обеспечение можно использовать.

В содержание задания описано, как сделать выполнить поставленную цель

Этапы выполнения содержат в себе информацию как выполнить задание, что нужно сделать на каждом из этапов, дается краткая инструкция по выполнению задания и что нужно получить в результате выполнения каждого из этапов задания.

Метод практической работы способствует углублению знаний и оттачиванию навыков, стимулированию познавательной деятельности и решению задач контролирующего и коррекционного характера.

Переход к следующей лекции возможно только в том случае если у слушателя будет выполнено практическое задание.

Такой способ обучения наиболее эффективный потому, что, он позволяет давать информацию студентам

3 ГЛАВА «РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ АВТОМАТИЗАЦИИ ПРОВЕРОК ПРАКТИЧЕСКИХ ЗАДАНИЙ ПО ДИСЦИПЛИНЕ “АВТОМАТИЗАЦИЯ МАШИННОГО ОБУЧЕНИЯ”»

3.1 Проектирование ПО

Для проверки и оценки тестовых заданий предложено использование средств автоматизации проверки самой платформы LMS. Более подробно предложенная система оценивания описана в предыдущих разделах.

Для автоматизации проверки практических работ студентов на платформе LMS нет подходящих инструментов. В связи с этим возникла необходимость разработки собственного ПО для автоматизации проверки практических работ. Для разрабатываемого ПО был выдвинут ряд требований, позволяющих выполнять проверку наиболее эффективно. Такими требованиями стали:

– Масштабируемость — что означает возможность изменения количества и состава выполняемых проверок;

– Гибкость настроек проверок — предполагает возможность независимого редактирования проводимых проверок для каждой практической работы без ущерба для работы всего ПО;

– Портабельность — означает возможность легкого переноса ПО на другие системы без требования установки;

– Интерактивность — возможность принимать от пользователя желаемые параметры проверки в процессе работы.

Так, для реализации ПО был выбран язык программирования bash. Данный выбор обусловлен наличием интерпретатора на подавляющем количестве современных дистрибутивов Linux. Предполагается, что проверка заданий преподавателем производится именно в ОС Linux, для тестирования был взят дистрибутив Ubuntu 22.04.

В качестве структуры ПО была принята структура фреймворка. ПО предоставляется в виде общей папки, переносимой на систему, и содержащей несколько файлов-скриптов на языке bash. Среди указанных файлов есть основная функция запуска ПО, содержащаяся в отдельном файле. Именно путем запуска этого файла с передачей определенных параметров ПО начинает и осуществляет дальнейшую работу. Также присутствуют отдельные файлы, содержащие функции-плагины проверок конкретных практических работ. Основной файл содержит импорты всех файлов-плагинов и таким образом может вызывать из них функции для своей работы. Общий вид архитектуры разрабатываемого ПО представлен на рисунке 28.

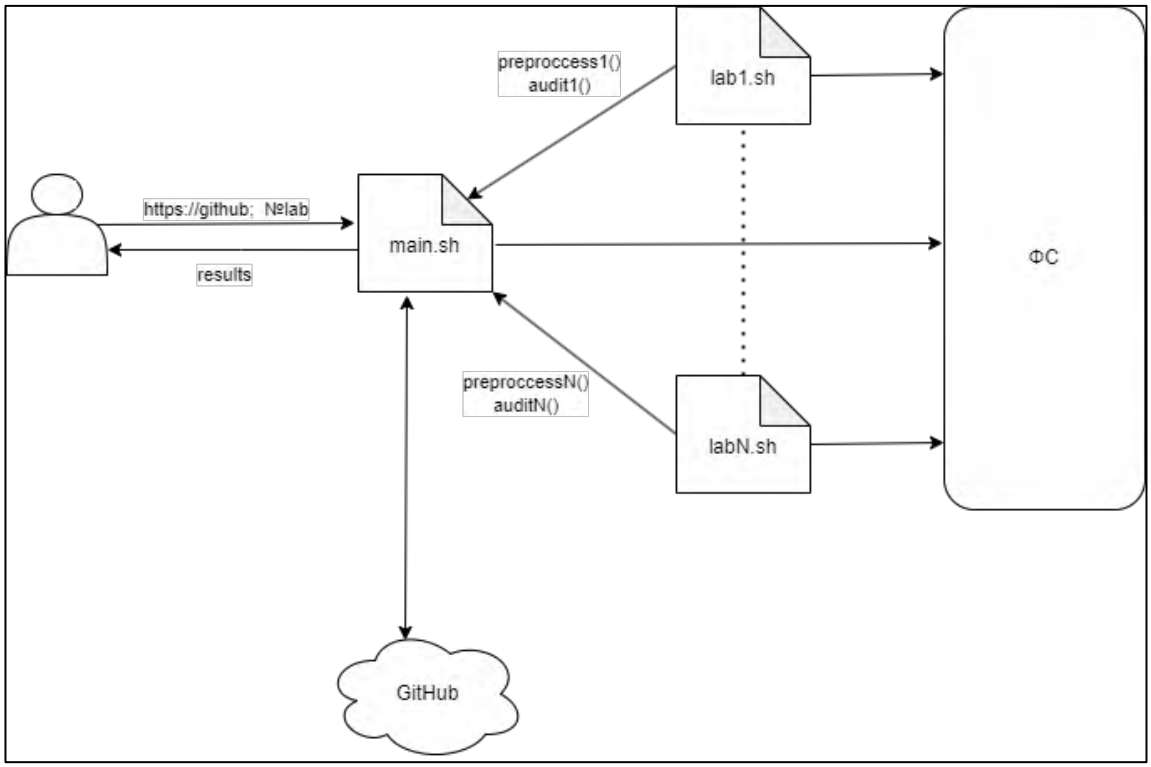


Рисунок 28 — Общая схема работы ПО

Такая архитектура позволяет соблюдать принципы масштабируемости и гибкости — при добавлении новых практических работ, либо при удалении устаревших, работа всего ПО никак не нарушается, корректируются только зависимости в основном файле и добавляются или удаляются файлы-плагины. При изменении или корректировке проверок для конкретной лабораторной

работы также происходит работа только с конкретным файлом-плагином, содержащим проверки для нужного практического задания, что никак не влияет на проверку других работ.

При написании кода ПО не было использовано дополнительных библиотек или языков программирования — все ПО написано на чистом bash. Это позволяет утверждать, что ПО является работоспособным на всех Debian-подобных дистрибутивах Linux, с которых и предполагается выполнять проверку. Таким образом, требования к системе, на которой предполагается запуск и работа разработанного ПО, минимальны, а именно [24]:

– Наличие системного интерпретатора bash (есть по умолчанию в современных Debian-подобных дистрибутивах Linux),

– Наличие менеджера процессов systemd (есть по умолчанию в современных Debian-подобных дистрибутивах Linux),

– Наличие интерпретатора python3 (необходимо для проверки работоспособности создаваемых студентами моделей ML),

– Наличие среды тестирования pytest (необходимо для проверки корректности работы разработанных студентами тестов),

– Наличие установленного пакета git (необходимо для клонирования репозитория git, содержащего работы студентов),

– Наличие установленного пакета docker (необходимо для осуществления проверок работоспособности создаваемых студентами контейнеров),

– Наличие установленного пакета curl (необходимо для проверки корректности работы разворачиваемых сервисов),

– Наличие установленного пакета Jenkins (необходимо для проверки работоспособности конвейеров в среде Jenkins),

– Наличие установленного пакета dvc (необходимо для тестирования корректности работы инициализации системы контроля версий).

Отметим, что основной файл программы называется main.sh. Файлы-плагины имеют префикс «lab» и далее номер практического задания, которое подлежит проверке. Так, например, плагин для проверки первого практического задания называется lab1.sh, второго — lab2.sh, и т.д. Так, схема работы проверяющего преподавателя будет выстроена таким образом:

– На локальную систему переносится папка с ПО для автоматизированной проверки;

– Выполняется чтение рекомендаций по работе с ПО из файла readme,

– Выполняется подготовка системы путем установки необходимых пакетов из файла requirements,

– Для всех bash-скриптов назначаются права на исполнение,

– Запускается основной скрипт работы ПО — main.sh, с правами суперпользователя,

– Затем преподаватель получает сообщение о завершении автоматизированной проверки, проводит корректировку сомнительных результатов вручную и запускает продолжение выполнения программы,

– После - работа ПО завершается автоматически.

Стоит отметить еще несколько важных условий для корректной работы разработанного ПО. А именно: в своем репозитории преподаватель должен создать папки для каждой практической работы для загрузки результатов выполнения работы студентов. Данные папки носят названия, состоящие из префикса «lab» и далее номер практического задания, по которому студентам необходимо отправить результаты. Такое название папок заложено работой ПО. Для студентов работа выстраивается следующим образом — они выполняют fork нужной папки по номеру практического задания, там создают папку со своей фамилией и инициалами — это также обязательное условие для корректной проверки результатов, предварительно могут склонировать репозиторий к себе в локальное хранилище. Далее выполняют само задание и

все необходимые файлы добавляют в свою папку в репозитории. Далее выполняют pull на свой удаленный репозиторий в GitHub (если предварительно репозиторий был склонирован) и затем выполняют pull request обратно вы репозиторий преподавателя. При верном выполнение конфликтов в репозитории не возникает по причине создания студентами личных папок.

3.2 Разработка основного функционала

Для реализации основного файла работы ПО была определена следующая функциональность, включенная в файл main.sh: обозначение интерпретатора для выполнения скрипта, подключение всех файлов-плагинов, содержащих функции проверок, запрос ссылки на основной GitHub репозиторий, куда присылаются работы, запрос номера лабораторной работы у пользователя, клонирование репозитория со сданными практическими заданиями студентов, создание файлов хороших результатов и сомнительных результатов, требующих ручной перепроверки, вызов функции предобработки (preproccessN()) по конкретному номеру для конкретной практической работы, затем начинается цикл прохода по всем склонированным директориям студентов. В теле цикла программа переходит в подкаталог студента и вызывает функцию проверки (auditN()) по конкретному номеру для конкретной практической работы, в которой и содержатся все проверки для оценки данной работы. В теле функции проверки результаты студента определяются как хорошие, либо сомнительные, также заполняются файлы промежуточных результатов. Далее цикл завершается, и программа сообщает пользователю о завершении процесса автоматической проверки.

В качестве отдельных достоинств такого решения стоит отметить, во-первых, наличие вызова функции предобработки для каждой конкретной лабораторной работы. Такая функция вызывается еще при расположении в общей папке практической работы, до начала перехода по вложенным директориям студентов. Это позволяет провести подготовительные манипуляции для качественной подготовки системы к началу автоматической

проверки. Сюда входит создание файлов промежуточных результатов. Это файлы, призванные логировать этапы проверки работ студентов с целью упрощения разбора неверно выполненных заданий для преподавателя. Анализ таких файлов позволит более быстро и эффективно определять ход работы, сданной студентом отчетности и обнаружить возникшую проблему. Также функция предобработки может включать дополнительный функционал настройки системы для проверки — например, проверку статуса нужного демона или процесса, статуса сетевого соединения или порта и другие типы проверок. Исходя из разнотипности принимаемых в практических работах решений целесообразнее всего такую функцию предобработки реализовывать отдельно для каждой практической работы, что и было выполнено в рамках разработки программного обеспечения. Во-вторых, в качестве достоинства разработанного решения можно выделить наличие файлов промежуточных результатов, назначение которых было описано выше. Т.е. высокую степень логирования процесса проверки. А также — возможность просмотра «неудачных» результатов студентов. Ни один результат студента не помечается как плохой или невыполненный. Результаты лишь заносятся в файл, отмеченный как «сомнительные». Такое решение связано со сложностью автоматизации однозначной оценки выполнения работы как «плохой», так как существует большое множество различных причин, вызывающих ошибки при выполнении автоматической проверки от неправильного названия файла до неправильных результатов его работы. Преподавателю предложено самостоятельно дополнительно проконтролировать возможные причины некорректной отработки проверок, что существенно повышает объективность оценивания в отличие от полностью автоматизированных систем проверок. Наиболее же значимые критерии невыполнения задания — например, отсутствие необходимых для выполнения работы файлов, ПО фиксирует в файлах промежуточных результатов, что позволяет преподавателю легко обнаруживать невыполненные, но сданные работы. Учет всех критериев и всех вариантов

ошибок при проверке работ привел бы к чрезмерному усложнению исходного кода и работы программы, но тем не менее не гарантировал бы полную вероятность достоверных результатов. Так, было выбрано оптимальное решение этой задачи, описанное выше [25 — 26].

Сокращенная блок-схема работы файла main.sh представлена на рисунке 29.

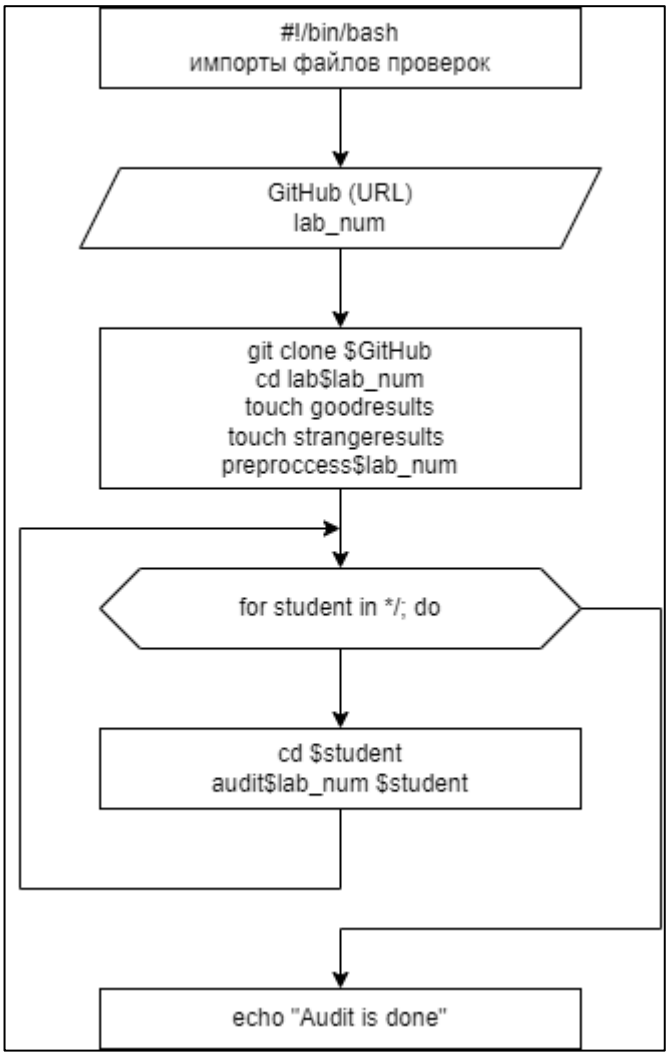


Рисунок 29 — Блок-схема работы main.sh

Листинг полного кода программы представлен в приложении А.

3.3 Разработка плагина проверки первого практического задания

Для выполнения и сдачи первой практической работы студентам необходимо правильно и полно выполнить практическое задание в модуле 1. Название файла-конвейера обязательно задавать как pipeline.sh, так как скрипт выполняет проверку наличия такого файла среди сданных. Файл pipeline.sh в качестве итога своей работы должен выводить в поток стандартного вывода единственную строку, содержащую оценку метрики работы модели. Это строка вида «Metrica is right N», где N соответствует полученному числу оценки метрики. В созданных для сдачи собственных папках студентам обязательно необходимо приложить все файлы: как python-скрипты обработки данных и работы модели, так и bash-скрипт, выполняющий функцию конвейера обучения — т.е. сам pipeline.sh. Без включения файлов-скриптов на python конвейер не сможет обеспечить корректную работу, а значит, задание не будет считаться выполненным.

Файл-плагин проверки первой практической работы студентов включает функцию предобработки системы и функцию проверки выполнения задания. Функция предобработки создает в общей директории практической работы файл pipelinecheck, в который в дальнейшем заносится факт наличия в папках студентов файла pipeline.sh. При его наличии в файл pipelinecheck заносятся фамилия и инициалы студента (переданные скрипту в качестве первого аргумента из работы основного файла приложения — main.sh) и знак «+». При отсутствии файла pipline.sh в файле промежуточных результатов pipelinecheck напротив фамилии студента будет стоять знак «-». Это позволит преподавателю быстро оценить совсем неверное выполненные работы [27].

Функция проверки работ студентов включает непосредственно проверку наличия файла pipeline.sh, описанную выше, результат которой фиксируется в файле pipelinecheck, а также проверку результата запуска файла pipeline.sh на соответствие шаблону по ожидаемой строке, которой является «Metrica is right». Полный листинг кода программы приведен в приложении Б.

3.4 Разработка плагина проверки второго практического задания

Для выполнения и сдачи второй практической работы студентам необходимо правильно и полно выполнить практическое задание в модуле 2. Название сгенерированного в среде Jenkins файла-конвейера также необходимо задавать как pipeline.sh, так как скрипт и для второй практической работы выполняет проверку наличия такого файла в директории студента. Файл pipeline.sh в качестве итога своей работы по-прежнему должен выводить в поток стандартного вывода единственную строку, содержащую оценку метрики работы модели. Это строка вида «Metrica is right N», где N соответствует полученному числу оценки метрики. Также необходимо наличие файла Jenkins, подтверждающего работу студентов в среде Jenkins. Именно с помощью данного инструмента студентам предложено выполнить задание второй практической работы. В созданных для сдачи собственных папках студентам обязательно необходимо приложить все файлы: python-скрипты обработки данных и работы модели, Jenkins файл, скрипт конвейера pipeline.sh. При отсутствии какого-либо из необходимых для сдачи файлов задание либо автоматически считается невыполненным из-за факта отсутствия, либо провоцирует некорректную работу конвейера и результаты выполнения также отмечаются как сомнительные.

Файл-плагин проверки второй практической работы включает функцию предобработки системы и функцию проверки выполнения задания. В данном файле функция предобработки создает в общей директории практической работы файлы pipelinecheck и jenkinscheck. Как и в предыдущей работе, данные файлы промежуточных результатов призваны логировать наличие главных файлов работы - pipeline.sh и Jenkins [28].

Функция проверки работ студентов включает непосредственно проверку наличия файла pipeline.sh, файла Jenkins, свидетельствующее о верном варианте работы при создании конвейера. Также происходит проверка результата запуска файла pipeline.sh на соответствие шаблону по ожидаемой

строке, которой является «Metrica is right», аналогично первой практической работе. Полный листинг кода программы приведен в приложении В.

3.5 Разработка плагина проверки третьего практического задания

Для выполнения и сдачи третьей практической работы студентам необходимо правильно и полно выполнить практическое задание в модуле 3. В данном модуле студентам необходимо создать docker-контейнер, обеспечивающий работу выбранной модели как микросервиса. Чтобы проверка при этом происходила корректно, необходимо при выполнении работы учесть следующие требования: приложить все файлы, необходимые для создания и развертывания контейнера в среде, отличной от среды разработки (это сам Dockerfile, requirments.txt, основной файл с кодом приложения, и т.д.), генерируемый контейнер обязательно должен запускаться и продолжать работу как сервис — т.е. не завершать свою работу сразу после запуска, взаимодействовать с пользователем, TCP-порт контейнера, открываемый для подключений, должен быть 8501 TCP-портом, при обращении к основной веб-странице разворачиваемого приложения сервер должен отвечать кодом успешного возвращения контента — 200.

Файл-плагин проверки третьей практической работы включает функцию предобработки системы и функцию проверки выполнения задания. В данном файле функция предобработки создает в общей директории практической работы файлы промежуточных результатов dockerfilecheck, dockercreatecheck, dockerupcheck, responsecheck, отвечающие за логирование наличия Dockerfile, корректности создания контейнера, корректности запуска контейнера и корректности ответа сервера после запуска соответственно. Данные файлы фиксируют только фамилию и инициалы студента, и знак «+» или «-» в зависимости от выполнения условия. Также функция предобработки проверяет статус службы docker в системе и в случае неактивного состояния перезапускает ее.

Функция проверки работ студентов включает первым шагом проверку наличия файла Dockerfile. Результаты при этом заносятся в промежуточный файл dockerfilecheck. Далее происходит построение контейнера командой docker build из полученного от студента докер файла. И затем происходит запуск созданного контейнера командой docker run. При построении контейнера ему присваивается тэг с фамилией и инициалами студента, передаваемыми в качестве аргумента вызываемой функции. Далее происходит проверка удачной сборки контейнера по поиску существующих в системе контейнеров командой docker ps, вывод которой отфильтрован по присвоенному ранее контейнеру тэгу. При нахождении такого контейнера — в файле промежуточных результатов dockercreatecheck студент отмечается как успешно создавший контейнер. Далее выполняется проверка на состояние работы контейнера после его запуска. Полученный и отфильтрованный на предыдущем шаге результат команды docker ps теперь фильтруется далее по статусу «Up». Таким образом, если контейнер запущен и выполняется, результат студента будет зафиксирован как положительный в файле dockerupcheck. Последняя проверка для работы микросервиса — получение верного кода ответа от приложения. Утилитой curl посылается запрос на проброшенные адрес и порт локальной системы для получения ответа от приложения. Если статус-код ответ равен 200, результат студента заносится в файл responsecheck как положительный, в противном случае — как отрицательный [29]. При этом на каждом шаге выполнения проверок переменная res, созданная в начале функции, увеличивается на единицу. Она может представлять полученный балл студента в расчёте из 4. В зависимости от значения данной переменной в конце программы будет заполнен файл goodresults с ФИО студента, либо strangeresults также с ФИО студента и с оставленными комментариями, содержание которых и зависит от переменной res. Полный листинг кода программы приведен в приложении Г.

3.6 Разработка плагина проверки четвертого практического задания

Для выполнения и сдачи четвертой практической работы студентам необходимо правильно и полно выполнить практическое задание в модуле 4. Расширение файла-инициализатора dvc обязательно задавать с расширением «.dvc». Данный файл должен присутствовать в сдаваемой работе. Также после применения указанных в инициализаторе настроек должна быть создана папка, обязательно содержащая в названии подстроку data. В данную папку впоследствии и попадут итоговые файлы работы.

Файл-плагин проверки четвертой практической работы студентов включает функцию предобработки системы и функцию проверки выполнения задания. Функция предобработки создает в общей директории практической работы файл dvccheck, в который заносится результат наличия в папках студентов файла с расширением «.dvc». При его наличии в файл dvccheck заносятся фамилия и инициалы студента (переданные скрипту в качестве первого аргумента из работы основного файла приложения — main.sh) и знак «+». При отсутствии файла с расширением «.dvc» в файле промежуточных результатов dvccheck напротив фамилии студента будет стоять знак «-». Это позволит преподавателю быстро оценить совсем неверное выполненные работы.

Функция проверки работ студентов включает непосредственно проверку наличия файла с расширением «.dvc», описанную выше, результат которой фиксируется в файле dvccheck, а также проверку результата выполнения команды dvc pull, в результате которой в текущей директории создается поддиректория, включающая в названии подстроку «data». Данная директория проверяется на существование и заполнение содержимым [30]. При прохождении всех этапов проверки работа считается успешно выполненной. Полный листинг кода программы приведен в приложении Д.

3.7 Разработка плагина проверки пятого практического задания

Для выполнения и сдачи пятой практической работы студентам необходимо правильно и полно выполнить практическое задание в модуле 5. Так, студенты выполняют данную работу, используя исключительно функционал сервиса Jupyter Notebook. Суть работы заключается при этом в создании автоматизированного выполнения тестирования работы, созданной ранее модели машинного обучения. На данном этапе разработки ПО не выявлено наиболее эффективного метода автоматизации проверки приведенного типа задания. В связи в этим, в настоящий момент решено проводить проверку пятой практической работы в ручном режиме преподавателем. Тем не менее, в дальнейшем планируется расширение функционала разработанного ПО, включающее настройку автоматизации проверки пятого практического задания.

ЗАКЛЮЧЕНИЕ

В результате выполнения был создан учебно-практический комплекс для автоматизации машинного обучение. Было создано ПО для автоматизации проверки практических работа для автоматизации машинного обучения.

В первой главе проведен сравнительный анализ LMS и облачных платформ. Были выбраны варианты которое удовлетворяют потребностям создаваемого учебно-практического комплекса.

Во второй главе описан создаваемый учебно-практический комплекс и его структура, приведены примеры тестовых и практических заданий для каждого модуля. Описаны критерии оценивая и приведен пример оценивания на LMS платформе.

В третьей главе описана общая схема работы ПО какой функционал и на каком языке были написаны скрипты. Описан основной функционал ПО для автоматизации проверки практических заданий, отмечены достоинства выбранного метода. Описаны все практические задания, что требуется выполнить студентам и как проверяются практические задания.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Гридчин В.С. Методология управления машинным обучением — MLOps / В. С. Гринд, Н. А. Чаплыгин, В. А. Балаев // Фундаментальные и прикладные научные исследования: актуальные вопросы, достижения и инновации: сб. трудов конференции / под ред. Г. Ю. Гуляев — Пенза: Издательство Наука и Просвещение, 2021. — 248 с. — URL: https://elibrary.ru/item.asp?edn=pxqqkr (дата обращения: 10.04.2023).

2. Гриншкун В. В. Отечественный и зарубежный опыт организации образовательного процесса на основе построения индивидуальных образовательных траекторий // В. В. Гриншкун, А. А. Заславский // Вестник Московского городского педагогического университета. Серия: Информатика и информатизация образования. — 2020. — № 1. — С. 8—15. — URL: https://dlt.mgpu.ru/2020/05/15/otechestvennyj-i-zarubezhnyj-opyt-organizaczii-obrazovatelnogo-proczessa-na-osnove-postroeniya-individualnyh-obrazovatelnyh-traektorij/ (дата обращения: 11.04.2023).

3. Исаева Е. С. Современные LMS платформы дистанционного обучения: анализ и сравнение / Е. С Исаева // Педагогика. Вопросы теории и практики. — 2021. — № 6. — С. 1045—1050. — URL: https://cyberleninka.ru/article/n/sovremennye-lms-platformy-distantsionnogo-obucheniya-analiz-i-sravnenie (дата обращения: 11.04.2023).

4. Колосков В. Л. Дистанционное образование. LMS и МООС / В. Л. Колосков, И. Ю. Павлов // Актуальные вопросы модернизации российского образования: сб. трудов конференции / Москва: Перо, 2015. — С. 32—34. — URL: https://www.elibrary.ru/item.asp?id=24316471&pff=1 (дата обращения: 15.04.2023).

5. Кубетаев А. С. LMS системы — инновационные образовательные технологии / А. С. Кубетаев // Непрерывность педагогического образования — залог успешности современных педагогов: сб. трудов конференции / Костанай: Костанайский областной университет им. Ахмета Байтурсынова,

2022. — С. 429—432. — URL: https:// repo.kspi.kz›handle/123456789/5793 (дата обращения: 15.04.2023).

6. Лавриненко И.Ю. Перспективы использования LMS в рамках современного высшего образования / И. Ю. Лавриненко // Концепт. — 2023. — № 1. — С. 18—35. — URL: https://cyberleninka.ru/article/n/perspektivy-ispolzovaniya-lms-v-ramkah-sovremennogo-vysshego-obrazovaniya (дата обращения: 16.04.2023).

7. Нагаева И. А. Гибридное обучение как потенциал современного образовательного процесса / И. А. Нагаева, И. А Кузнецов // Отечественная и зарубежная педагогика. — 2022. — № 3. — С. 126—139. — URL: https://cyberleninka.ru/article/n/gibridnoe-obuchenie-kak-potentsial-sovremennogo-obrazovatelnogo-protsessa (дата обращения: 16.04.2023).

8. Шатуновский В. Л. Ещё раз о дистанционном обучении (организация и обеспечение дистанционного обучения) / В. Л. Шатуновский, Е. Л. Шатуновская // Вестник науки и образования. — 2020. — № 9—1 (87). — С. 53—56. — URL: https://cyberleninka.ru/article/n/eschyo-raz-o-distantsionnom-obuchenii-organizatsiya-i-obespechenie-distantsionnogo-obucheniya (дата обращения: 21.04.2023).

9. Реализация технологии смешанного обучения средствами LMS Moodle / А. А. Шкунова, М. П. Прохорова, А. В. Лабазова [и др.]. // Инновационная экономика: Перспективы развития и совершенствования. — 2019. — № 2 (36). — С. 108—115. — URL: https://cyberleninka.ru/article/n/realizatsiya-tehnologii-smeshannogo-obucheniya-sredstvami-lms-moodle (дата обращения: 21.04.2023).

10. Use of the LMS Moodle for an effective implementation of an innovative policy in higher educational institutions / M. Zabolotniaia, Z. Cheng, E. Dorozhkin, A. Lyzhin // International Journal of Emerging Technologies in Learning (iJET). — 2020. — № 13 (15). — PP. 172—189. — URL: htpps:// https://www.researchgate.net/publication/342850447_Use_of_the_LMS_Moodle_f

or_an_Effective_Implementation_of_an_Innovative_Policy_in_Higher_Educational_Institutions (дата обращения: 21.04.2023).

11. Мингазова А. Ф. Использование LMS в условиях смешанного обучения в вузе / А. Ф. Мингазова // Инновации. Наука. Образование. — 2021. — № 34. — С. 1341—1347. — URL: https://www.elibrary.ru/item.asp?id=46320594 (дата обращения: 21.04.2023).

12. Миронова Ю. Н. Использование дистанционных технологий при проведении занятий: LMS Moodle, Google класс, Microsoft teams / Ю. Н. Миронова // Азимут научных исследований: педагогика и психология. — 2021. — № 6. — С. 177—182. — URL: https://cyberleninka.ru/article/n/ispolzovanie-distantsionnyh-tehnologiy-pri-provedenii-zanyatiy-lms-moodle-google-klass-microsoft-teams/viewer (дата обращения: 21.04.2023).

13. Karimjanovna S. V. Современные LMS платформы дистанционного обучения / S. V. Karimjanovna // Global Congress. — 2023. — № 3. — С. 13—18. — URL: https// https://www.gramota.net/materials/4/2021/6/19.html (дата обращения: 22.04.2023).

14. Мозохин А. Е. Анализ направлений развития цифровизации отечественных и зарубежных энергетических систем / А. Е. Мозохин, В. Н. Шведенко // Научно-технический вестник информационных технологий, механики и оптики. — 2019. — № 4. — С. 657—672. — URL: https://cyberleninka.ru/article/n/analiz-napravleniy-razvitiya-tsifrovizatsii-otechestvennyh-i-zarubezhnyh-energeticheskih-sistem (дата обращения: 26.04.2023).

15. Петров В. Ю. Цифровая экономика: состояние и перспективы отечественных облачных сервисов / В. Ю. Петров, А. А. Кричко // Вестник Алтайской академии экономики и права. — 2019. — № 3—2. — С. 129—135. — URL: https://https://vaael.ru/ru/article/view?id=383 (дата обращения 26.04.2023).

16. Alla S. Beginning MLOps with MLFlow / S. Alls, S. K. // Adari — Berkeley, CA: Apress, 2020. — 330 p. — URL:

https://link.springer.com/book/10.1007/978-1-4842-6549-9 (дата обращения: 26.04.2023).

17. Kreuzberger D. Machine learning operations (mlops): Overview, definition, and architecture / D. Kreuzberger, N. Kühl, S. Hirschl // IEEE Access. — 2023. — № 11. — PP. 31866—31879. — URL: htpps:// https://ieeexplore.ieee.org/document/10081336 (дата обращения: 02.05.2023).

18. Ямиков Р. Р. Анализ и разработка конвейера MLOps для развертывания моделей машинного обучения / Р. Р. Ямиков, К. А. Григорян // Электронные библиотеки. — 2022. — Т. 25. — № 2. — С. 177—196. — URL: https:// https://rdl-journal.ru/article/view/735 (дата обращения: 05.05.2023).

19. Литвяков Д. С. Разработка конфигурации ci/cd для автоматизации развёртывания и управления приложениями / Д. С. Литвяков // StudNet. — 2021. — № 6. — С. 1089—1097. — URL: https://cyberleninka.ru/article/n/razrabotka-konfiguratsii-ci-cd-dlya-avtomatizatsii-razvyortyvaniya-i-upravleniya-prilozheniyami (дата обращения: 10.05.2023).

20. Мурзин Ф. А. Облачные технологии: основные модели, приложения, концепции и тенденции развития / Ф. А. Мурзин, Т. В. Батура, Д. Ф. Семич // Программные продукты и системы. — 2014. — № 3 (107). — С. 64—72. — URL: https://cyberleninka.ru/article/n/oblachnye-tehnologii-osnovnye-modeli-prilozheniya-kontseptsii-i-tendentsii-razvitiya-1 (дата обращения: 10.05.2023).

21. Романова А. О. Виртуализация в высокопроизводительных вычислительных системах / А. О. Романова // Машиностроение и компьютерные технологии. — 2011. — № 3. — С. 1—29. — URL: https://cyberleninka.ru/article/n/virtualizatsiya-v-vysokoproizvoditelnyh-vychislitelnyh-sistemah (дата обращения: 11.05.2023).

22. Артюхова А.С. Проблемы автоматизации тестирования и подходы к их решению / А. С. Артюхова // Ceteris paribus. — 2016. — № 10. — С. 1—6. — URL: https://cyberleninka.ru/article/n/problemy-avtomatizatsii-testirovaniya-i-podhody-k-ih-resheniyu (дата обращения: 11.05.2023).

23. Оптимизация проведения практического занятия глазами студентов / Л. А. Шпагина, Л. Ю. Зюбина, С. В. Третьяков [и др.] // Медицина и образование в Сибири. — 2008. — № 2. — С. 1—11. — URL: https://cyberleninka.ru/article/n/optimizatsiya-provedeniya-prakticheskogo-zanyatiya-glazami-studentov (дата обращения: 12.05.2023).

24. Воротников И. Л. Теоретико—методологические особенности проектирования инновационной деятельности агробизнеса / И. Л. Воротников, Е. Л. Моренова // Инновационная деятельность. — 2021. — № 3. — С. 12—21. — URL: https://www.elibrary.ru/item.asp?id=47150143 (дата обращения: 12.05.2023).

25. Игнатьева О.В. Архитектурные приемы при разработке программного обеспечения, зависимого от интерфейса пользователя / О. В. Игнатьева, Н. А. Москат, С. С. Рубцова // Инженерный вестник Дона. — 2022. — № 2. — С. 1—9. — URL: http://www.ivdon.ru/uploads/article/pdf/IVD_19__2_ignateva_moskat_rubtsova.pdf_351757a28f.pdf (дата обращения: 13.05.2023).

26. Перязева Ю. В. Возможности автоматической проверки заданий в LMS Moodle / Ю. В. Перязева // Современные информационные технологии и ИТ-образование. — 2019. — № 4. — С. 876—885. — URL: https://cyberleninka.ru/article/n/vozmozhnosti-avtomaticheskoy-proverki-zadaniy-v-lms-moodle (дата обращения: 13.05.2023).

27. Тюменцев Е.А. Алгоритм загрузки плагинов, не имеющих явных зависимостей между собой / Е. А. Тюменцев, Д. Ю. Згуровец // Математические структуры и моделирование. — 2021. — № 1 (57). — С. 100—104. — URL: https://cyberleninka.ru/article/n/algoritm-zagruzki-plaginov-ne-imeyuschih-yavnyh-zavisimostey-mezhdu-soboy (дата обращения: 14.05.2023).

28. Лазарева Н.Б. Оптимальный подход к разработке программного обеспечения с использованием современных методологий и технических средств / Н. Б. Лазарева // Инженерный вестник Дона. — 2020. — № 10 (70). — С. 1—10. — URL:

http://www.ivdon.ru/uploads/article/pdf/IVD_48__9_Lazareva.pdf_740fb225f8.pdf (дата обращения: 14.05.2023).

29. Васильев П. А. Развертывание сервера с помощью технологии docker / П. А. Васильев // Вестник науки и образования. — 2016. — № 12 (24). — С. 1—25. — URL: https://cyberleninka.ru/article/n/razvertyvanie-servera-s-pomoschyu-tehnologii-docker (дата обращения: 14.05.2023).

30. Лисецкий Ю. М. Комплексный подход к управлению данными / Ю. М. Лисецкий // Математические машины и системы. — 2019. — № 4. — С. 93—99. — URL: https://cyberleninka.ru/article/n/kompleksnyy-podhod-k-upravleniyu-dannymi (дата обращения: 14.05.2023).

ПРИЛОЖЕНИЕ А

КОД ОСНОВНОЙ ПРОГРАММЫ main.sh

```
#!/bin/bash
. ./lab1.sh
. ./lab2.sh
. ./lab3.sh
. ./lab4.sh
. ./lab5.sh

echo "Set your GitHub URL, please"
read GitHub
echo "Set a lab number, please"
read lab_num
URL_GitHub = $GitHub+'/lab'+$lab_num
git clone $URL_GitHub
cd lab$lab_num
touch goodresults
touch strangeresults
preproccess$lab_num
      for student in */; do
            cd $student
            audit$lab_num $student
      done
echo "Audit is done! Please, check files goodresults and strangeresults"
```

ПРИЛОЖЕНИЕ Б

КОД ПРОГРАММЫ lab1.sh

```
#!/bin/bash

preproccess1 () {
touch pipelinecheck
}

audit1 () {
            if [[ -n "$(find . -name 'pipeline.sh')" ]]
            then
                  echo $1+" +" >> ../pipelinecheck
                  path = $(find . -name "pipeline.sh" | grep pipeline.sh)
                  chmod a+x $path
                  execution = $(/bin/bash $path)
                  result = $(awk '{print $0 $1 $2}' $execution)
                  if [[ $result = "Metrica is right" ]]
                  then
                        echo $1+" great!" >> ../goodresults
                  else
                        echo $1+"  "+$execution >> ../strangeresults
                  fi
            else
                  echo $1+" -" >> ../pipelinecheck
            fi
      done
}
```

ПРИЛОЖЕНИЕ В

КОД ПРОГРАММЫ lab2.sh

```
#!/bin/bash

preproccess2 () {
touch pipelinecheck
touch jenkinscheck
}

audit2 () {
            if [[ -n "$(find . -name 'pipeline.sh')" ]]
            then
                  echo $1+" +" >> pipelinecheck
                  if [[ -n "$(find . -name 'jenkins')" ]]
                        then
                        echo $1+" +" >> ../jenkinscheck
                        path = $(find . -name "pipeline.sh" | grep
pipeline.sh)
                        chmod a+x $path
                        execution = $(/bin/bash $path)
                        result = $(awk '{print $0 $1 $2}' $execution)
                        if [[ $result = "Metrica is right" ]]
                        then
                              echo $1+" great!" >> ../goodresults
                        else
                              echo $1+"  "+$execution >> ../strangeresults
                        fi
                  else
                        echo $1+" -" >> ../jenkinscheck
                  fi
            else
                  echo $1+" -" >> ../pipelinecheck
            fi
      done
}
```

ПРИЛОЖЕНИЕ Г

КОД ПРОГРАММЫ lab3.sh

```
#!/bin/bash

preproccess3 () {
if [[ -z "$(sudo systemctl status docker | grep running)" ]]
then
sudo systemctl start docker
fi
touch dockerfilecheck
touch dockercreatecheck
touch dockerupcheck
touch responsecheck
}

audit3 () {
res = 0
if [[ -n "$(find . -name 'Dockerfile')" ]]
              then
                     res = 1
                     echo $1+" +" >> ../dockerfilecheck
              else
                     echo $1+" -" >> ../dockerfilecheck
fi

sudo docker build Dockerfile . -t $1
sudo docker run --rm -p "127.0.0.1:8501:8501/tcp" $1

dockercreate = $(sudo docker ps | grep $1)
if [[ -n "$dockercreate" ]]
              then
                     res = 2
                     echo $1+" +" >> ../dockercreatecheck
              else
                     echo $1+" -" >> ../dockercreatecheck
fi

dockerup = $(sudo docker ps | grep $1 | grep Up)
if [[ -n "$dockerup" ]]
              then
                     res = 3
                     echo $1+" +" >> ../dockerupcheck
              else
                     echo $1+" -" >> ../dockerupcheck
```

```
fi

response=$(curl --silent --write-out "HTTPSTATUS:%{http_code}" -X GET
http://127.0.0.1:80)
status=$(echo $response| tr -d '\n' | sed -e 's/.*HTTPSTATUS://')
if [[ "$status" -eq 200 ]]
             then
                    res = 4
                    echo $1+" +" >> ../responsecheck
             else
                    echo $1+" -" >> ../responsecheck
fi

sudo docker stop $1

if [[ "$res" -eq 4 ]]; then
       echo $1+" great!" >> ../goodresults
elif [[ "$res" -eq 3 ]]; then
       echo $1+" wrong http response code" >> ../strangeresults
elif [[ "$res" -eq 2 ]]; then
       echo $1+" docker can't be Up" >> ../strangeresults
elif [[ "$res" -eq 1 ]]; then
       echo $1+" docker can't be created" >> ../strangeresults
elif [[ "$res" -eq 0 ]]; then
       echo $1+" no Dockerfile" >> ../strangeresults
fi
```

ПРИЛОЖЕНИЕ Д

СОДЕРЖИМОЕ ФАЙЛА requirements.txt

```
python 3.11
git
openjdk-8-jre
jenkins
containerd.io
docker-ce-cli
docker-ce
dvc
pytest
```