МИНИСТЕРСТВО ПРОСВЕЩЕНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«РОССИЙСКИЙ ГОСУДАРСТВЕННЫЙ
ПЕДАГОГИЧЕСКИЙ УНИВЕРСИТЕТ им. А. И. ГЕРЦЕНА»

Направление подготовки
09.03.01 Информатика и вычислительная техника

Направленность (профиль)
«Технологии разработки программного обеспечения»

Выпускная квалификационная работа

Использование машинного обучения для анализа эффективности цифровизации учебного процесса

Обучающегося 4 курса
очной формы обучения
Воложанина Владислав Олеговича

Руководитель выпускной квалификационной работы:
кандидат физико-математических наук,
доцент кафедры ИТиЭО
Власов Дмитрий Викторович

Санкт-Петербург
2025

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

В последние годы цифровизация образовательного процесса становится одним из ключевых направлений развития высшей школы. Внедрение современных цифровых технологий не только расширяет возможности получения и обработки образовательных данных, но и предъявляет новые требования к эффективности учебных практик и организации образовательной среды. Для вузов и преподавателей становится крайне важным обладать инструментами, позволяющими не только собирать информацию о цифровой активности студентов, но и проводить глубокий анализ полученных данных для совершенствования учебного процесса.

Вместе с тем традиционные методы анализа образовательных данных часто оказываются недостаточно эффективными при работе с большими объемами информации и не позволяют выявлять скрытые закономерности в поведении обучающихся. Решением этих задач становится применение методов машинного обучения, способных автоматизировать процесс анализа, выделять типовые профили студентов и формировать индивидуальные рекомендации по развитию цифровых компетенций. Именно поэтому вопрос о внедрении машинного обучения в образовательную аналитику приобретает особую актуальность для современных образовательных организаций.

Целью дипломного проекта является использование методов машинного обучения для анализа эффективности цифровизации учебного процесса на основе анкетных данных студентов.

Предметом исследования выступают процессы цифровизации высшего образования, методы обработки и анализа образовательных данных, а также алгоритмы построения цифровых профилей студентов и генерации персонализированных рекомендаций.

Для достижения поставленной цели требуется выполнить следующие задачи:

1. Изучить методы машинного обучения, используемые в образовательной аналитике, с акцентом на кластеризацию, классификацию и методы визуализации.
2. Провести сбор, обработку и разведочный анализ анкетных данных студентов.
3. Разработать и реализовать модели кластеризации для выделения цифровых профилей студентов.
4. Построить классификатор для автоматического определения профиля нового студента.
5. Разработать и внедрить модуль генерации персонализированных рекомендаций с использованием API GigaChat.
6. Оценить распределение цифровых профилей среди студентов различных факультетов и провести автоматическую интерпретацию результатов.

Данная работа состоит из введения, первой главы, посвящённой теоретическим аспектам цифровизации образования и методам машинного обучения, и второй главы, в которой представлены этапы практической реализации анализа, построения моделей и генерации рекомендаций.

INTRODUCTION

In recent years, the digitalization of the educational process has become one of the key areas in the development of higher education. The introduction of modern digital technologies not only expands the possibilities for collecting and processing educational data, but also sets new requirements for the effectiveness of teaching practices and the organization of the educational environment. For universities and instructors, it is becoming critically important to have tools that enable not only the collection of information about students' digital activity but also the in-depth analysis of the obtained data to improve the educational process.

At the same time, traditional methods of educational data analysis often prove insufficiently effective when working with large volumes of information and do not allow for the identification of hidden patterns in student behavior. The solution to these challenges lies in the application of machine learning methods, which are capable of automating the analysis process, identifying typical student profiles, and generating individualized recommendations for the development of digital competencies. For this reason, the implementation of machine learning in educational analytics is of particular relevance for modern educational institutions.

The aim of this thesis project is to apply machine learning methods to analyze the effectiveness of the digitalization of the educational process based on student survey data.

The subject of the research includes the processes of digitalization in higher education, methods of processing and analyzing educational data, as well as algorithms for constructing digital student profiles and generating personalized recommendations.

To achieve this goal, the following objectives are set:

1. To study the machine learning methods used in educational analytics, with a focus on clustering, classification, and visualization techniques.
2. To collect, process, and conduct exploratory analysis of student survey data.

3. To develop and implement clustering models to identify digital student profiles.
4. To build a classifier for the automatic determination of the profile of a new student.
5. To develop and implement a module for generating personalized recommendations using the GigaChat API.
6. To assess the distribution of digital profiles among students of various faculties and perform automatic interpretation of the results.

This work consists of an introduction, the first chapter devoted to the theoretical aspects of digitalization in education and machine learning methods, and the second chapter, which presents the stages of practical implementation of data analysis, model development, and recommendation generation.

ГЛАВА I. ИССЛЕДОВАНИЕ ВОЗМОЖНОСТЕЙ АВТОМАТИЗАЦИИ В ТАБЛИЧНЫХ ПРОЦЕССОРАХ

1.1 Цифровизация в высшем образовании: понятия, инициативы и современные инструменты

Цифровизация высшего образования подразумевает интеграцию информационно-коммуникационных технологий в учебный процесс, административную деятельность и инфраструктурное обеспечение университетов. Данный процесс не ограничивается переходом на дистанционные формы, а предполагает изменение всей архитектуры образовательной среды за счёт внедрения электронных систем, автоматизации процедур и использования программных продуктов для сопровождения обучения. Применение цифровых платформ позволяет университетам управлять образовательным контентом, отслеживать прогресс студентов, формировать индивидуальные образовательные траектории и обеспечивать взаимодействие между участниками образовательного процесса. В рамках цифровой трансформации реализуются проекты по оснащению вузов высокоскоростным интернетом, созданию облачных хранилищ, внедрению электронных журналов, автоматизации расписания и документооборота, что существенно упрощает администрирование и организацию учебной деятельности. Использование LMS-платформ (Learning Management Systems) стало обязательным стандартом: программные комплексы Moodle, Blackboard, Canvas и аналогичные решения интегрируются в структуру университета, предоставляя единое пространство для размещения учебных материалов, обмена заданиями, проверки знаний, мониторинга посещаемости и фиксации результатов.

Программные инициативы федерального и международного уровня способствуют продвижению цифровых инноваций в университетской среде.

Примером служит стратегическая программа Министерства науки и высшего образования Российской Федерации, ориентированная на цифровую трансформацию вузов и рассчитанная до 2030 года, где основной акцент делается на формирование индивидуальных образовательных маршрутов, развитие научно-исследовательских компетенций и внедрение практик использования аналитики данных для совершенствования образовательных решений. Сходные задачи отражены в ряде международных документов, включая рекомендации ЮНЕСКО, где подчеркивается роль цифровизации для обеспечения инклюзивности и повышения качества образования. Мировое образовательное сообщество рассматривает развитие цифровых навыков и гибких компетенций как фундамент современного образования, что требует от вузов пересмотра структуры учебных программ, а также перестройки методов преподавания и взаимодействия с обучающимися.

Реализация цифровизации охватывает ряд ключевых направлений. Современная цифровая инфраструктура включает в себя оснащение учебных корпусов компьютерными классами, доступом к высокоскоростному интернету, средствами защиты данных и облачными сервисами. Электронные образовательные ресурсы, представленные мультимедийными курсами, виртуальными лабораториями, электронными учебниками и видео-лекциями, обеспечивают доступность учебных материалов вне зависимости от физического присутствия в аудитории. LMS-платформы служат ядром управления образовательным процессом, интегрируя функции размещения материалов, проверки заданий, организации форумов и обсуждений, а также ведения электронной отчетности. Создание и поддержка электронных портфолио и цифровых идентификаторов обучающихся позволяет автоматизировать процесс признания образовательных достижений и облегчает академическую мобильность. Появление государственных и межуниверситетских цифровых платформ, таких как "Современная цифровая образовательная среда", обеспечивает стандартизацию доступа к онлайн-

курсам, централизованный учет результатов и интеграцию с государственными системами идентификации.

Формирование новых моделей организации учебного процесса включает внедрение смешанного обучения, реализацию индивидуальных и гибких траекторий, массовых открытых онлайн-курсов и полностью дистанционных программ. Перестройка содержания образовательных программ происходит с учетом запросов рынка труда и возможностей цифровых технологий. Актуализируются задачи персонализации обучения, где студенты получают возможность самостоятельно выбирать последовательность и содержание изучаемых дисциплин, а образовательные платформы обеспечивают автоматическую адаптацию под индивидуальные интересы и уровень подготовки.

Современные инструменты цифровизации охватывают все этапы образовательного цикла. Применение электронных учебников и автоматизированных систем проверки знаний упрощает контроль усвоения материала. Видеоконференции, мессенджеры, платформы для совместной работы (Zoom, Teams, Trello, Miro) интегрируются в учебный процесс и обеспечивают коммуникацию между преподавателями и студентами, а также внутри студенческих групп. Использование сервисов онлайн-опросов и тестирования (Kahoot, Quizlet) позволяет собирать обратную связь и организовывать интерактивное взаимодействие в режиме реального времени. Программное обеспечение для инженерных, математических, языковых и других дисциплин расширяет инструментарий преподавателей, способствуя освоению практических навыков.

В образовательной среде активно внедряются инновационные технологии, включая искусственный интеллект, аналитику больших данных, адаптивные образовательные платформы, а также виртуальную и дополненную реальность. Системы аналитики позволяют собирать данные о прогрессе студентов, выявлять закономерности и предлагать индивидуальные рекомендации. Интеллектуальные агенты и чат-боты обеспечивают

автоматизированную поддержку обучающихся. AR/VR-технологии применяются для создания виртуальных лабораторий и тренажёров, которые позволяют воспроизводить профессиональные ситуации и формировать практические компетенции. Электронные сертификаты и дипломы с функцией верификации на основе блокчейна становятся стандартом для подтверждения квалификаций, облегчая процессы трудоустройства и международного признания образовательных результатов.

Цифровизация сопровождается обновлением нормативно-правовой базы, разработкой стандартов и требований к электронным образовательным ресурсам, а также совершенствованием механизмов обеспечения информационной безопасности. В российской практике формируется система обязательной аккредитации электронных информационно-образовательных сред, что гарантирует соответствие используемых цифровых инструментов установленным критериям качества и доступности. Широкое внедрение цифровых технологий приводит к изменению роли преподавателя, который становится организатором и координатором образовательного процесса, сопровождающим студента на индивидуальном маршруте.

Развитие цифровых компетенций обучающихся и преподавателей рассматривается как стратегическая задача, что обусловливает появление специализированных программ повышения квалификации, образовательных интенсивов и курсов цифровой грамотности. Важной тенденцией становится распространение практик непрерывного образования, когда цифровые платформы обеспечивают доступ к учебным ресурсам на протяжении всей профессиональной деятельности. Актуальность цифровизации подтверждается аналитическими данными о росте числа пользователей онлайн-образовательных платформ и увеличении объёма цифрового контента, что свидетельствует о трансформации образовательных практик на всех уровнях системы.

1.2 Основы и методы машинного обучения

Машинное обучение определяется как область искусственного интеллекта, фокусирующаяся на построении алгоритмов, которые извлекают зависимости и структуры из эмпирических данных и используют сформированные закономерности для решения практических задач, не предусматривающих явного программирования всех вариантов. Исходный набор данных включает объекты, представленные совокупностью признаков, и, при наличии разметки, целевыми переменными, которые подлежат прогнозированию либо категоризации. Система, обладающая способностью к машинному обучению, на этапе обучения анализирует примеры, оптимизирует внутренние параметры модели, минимизируя функцию ошибки, а затем демонстрирует способность к переносимости приобретённых знаний на новые входные данные. В определении, предложенном Томом Митчеллом, формализуется связь между качеством выполнения задачи, опытом и характеристиками модели, обучающейся на наборе примеров, что подчёркивает приоритет эмпирического подхода над детерминированным программированием.

Технологический цикл машинного обучения включает этапы сбора и структурирования обучающих данных, отбора информативных признаков, выбора и настройки алгоритма, процедуры оптимизации параметров на основании критерия качества, а также тестирования и оценки полученных моделей на независимой выборке. На практике для предотвращения переобучения и контроля обобщающей способности моделей используются методы кросс-валидации, регуляризации, фильтрации признаков и увеличение обучающего множества. Контроль качества работы моделей осуществляется с использованием метрик, отражающих способность алгоритма обобщать закономерности вне обучающей выборки.

Современная система классификации методов машинного обучения строится на принципе наличия или отсутствия априорной информации о

правильных ответах в обучающей выборке. Класс супервизированных алгоритмов (обучение с учителем) предполагает работу с размеченными данными, где для каждого объекта известна целевая переменная, подлежащая предсказанию. Типичными задачами в этой парадигме выступают классификация и регрессия. Классификация предполагает разделение объектов на конечное множество дискретных категорий на основе анализа признаков, при этом процесс обучения строится на сопоставлении реальных классов и прогнозируемых меток. Алгоритмы классификации варьируются от простых моделей, например, логистической регрессии и дерева решений, до сложных ансамблевых и нейросетевых архитектур, способных выявлять сложные, нелинейные зависимости между входными переменными и целевой переменной. Задача регрессии заключается в прогнозировании количественных показателей на основании признаков, где модель формирует функциональную зависимость между признаковым пространством и непрерывной переменной.

В рамках несупервизированного подхода к обучению используются данные, лишённые информации о целевых переменных или классах. Алгоритмы этой группы анализируют только распределение и структуру признаков объектов, осуществляя поиск внутренних закономерностей, паттернов и сегментов. Классическая задача в этом контексте — кластеризация, подразумевающая разбиение множества объектов на кластеры по критерию максимального сходства внутри группы и максимального различия между различными группами. Алгоритмы кластеризации различаются методами формирования сегментов: итеративные методы, такие как k-средних и Fuzzy C-Means, строят разбиение на фиксированное количество групп с возможностью мягкого (размытого) членства для FCM; иерархические подходы, такие как агломеративная кластеризация, организуют данные в виде дерева вложенных кластеров и допускают анализ структуры на разных уровнях детализации; плотностные алгоритмы формируют кластеры как области высокой плотности в пространстве признаков; вероятностные методы строят модели, основанные на предположении о распределении данных. Кластеризация находит применение

для сегментации пользователей, выявления профилей и предварительного анализа структуры данных. В ряде случаев используются методы снижения размерности, такие как метод главных компонент, t-SNE, UMAP, для упрощения анализа, визуализации и подготовки к последующей кластеризации, что позволяет представить многомерное пространство признаков в двумерном или трёхмерном виде без существенной потери информации о структуре выборки.

Результаты кластерного анализа формируют основу для перехода к задачам супервизированного обучения, где автоматически определённые профили или сегменты используются в качестве целевой переменной для последующего обучения классификаторов. На этапе классификации модели обучаются на выборке с присвоенными метками и верифицируются на тестовых данных по ряду метрик. Наиболее часто используются точность, полнота, F1-мера и значения макро- и микро-усреднённых показателей, особенно в условиях несбалансированных классов. Для объяснения решений и интерпретации значимости признаков в современных реализациях применяется анализ важности признаков, визуализация структуры дерева решений или оценка вклада переменных в ансамблевых моделях.

В современной образовательной аналитике машинное обучение становится инструментом выявления паттернов цифрового поведения, автоматизации профилирования обучающихся, а также предсказания индивидуальных образовательных траекторий. В процессе реализации задач на основе анкетных данных студентов в исследовании использовались и супервизированные, и несупервизированные методы: этап кластеризации позволил выделить типовые профили цифровых практик, а построение классификаторов обеспечило возможность автоматического определения профиля для новых студентов на основании их анкетных ответов. Процесс построения моделей сопровождался процедурами стандартизации признаков, отбора переменных с высокой информативностью и исключения скоррелированных либо слабо вариативных признаков, что повысило

устойчивость и качество итоговых решений. Были использованы методы предотвращения переобучения и реализованы алгоритмы интерпретации результатов, включая визуализацию структуры кластеров в пространстве UMAP и анализ важности признаков для выбранных моделей.

1.3 Программные средства, платформы и методы кластеризации и классификации

В рамках анализа образовательных данных и построения цифровых профилей студентов широко используются программные средства, разработанные с учётом специфики учебных задач, масштабов выборок и особенностей источников информации. Наиболее универсальным и применимым инструментом для реализации алгоритмов машинного обучения в образовательной среде является библиотека Scikit-Learn, включающая комплекс модулей для решения задач классификации, регрессии, кластеризации, отбора признаков, автоматизации поиска гиперпараметров и построения пайплайнов обработки. В среде научных и прикладных исследований Scikit-Learn используется для построения предиктивных моделей на основе анкетных, событийных и лог-файловых данных, анализа успеваемости, выявления факторов риска и прогнозирования образовательных траекторий. Реализация деревьев решений, логистической регрессии, метода опорных векторов, ансамблевых алгоритмов и модулей для визуализации структуры данных позволяет проводить широкий спектр исследований в области учебной аналитики.

Обработка текстовых, графических и событийных данных, а также построение моделей высокой сложности и глубины осуществляется посредством библиотек глубокого обучения, таких как TensorFlow, PyTorch, Keras. Данные фреймворки обеспечивают возможность построения, обучения и внедрения нейронных сетей с различной архитектурой для решения задач анализа ответов в свободной форме, автоматической проверки письменных работ, обработки изображений, предсказания вероятности досрочного отсева в онлайн-курсах, а также сегментации обучающихся на основе многоуровневых признаковых структур. Применение глубоких моделей повышает точность решения комплексных аналитических задач и обеспечивает обработку больших массивов неструктурированных данных.

Для аналитиков, не обладающих компетенциями в программировании, разработаны визуальные конструкторы анализа данных, примером которых выступает среда Orange. С помощью графического интерфейса и инструментов визуализации Orange используется в образовательных проектах для кластеризации студентов, построения дендрограмм, анализа коммуникаций и создания интерактивных отчётов по результатам учебной деятельности. Дополнительный набор методов для автоматизации анализа и экспериментальной проверки алгоритмов реализован в пакете Weka, который находит применение в исследованиях по Educational Data Mining и позволяет выполнять полный цикл от импорта данных до построения сложных моделей и анализа результатов в визуальной форме.

Внедрение аналитических модулей и методов машинного обучения в образовательную инфраструктуру осуществляется через современные платформы управления обучением, среди которых Moodle, Canvas, Blackboard. Указанные системы интегрируют специализированные расширения для мониторинга успеваемости, анализа рисков и раннего выявления студентов с потенциальными трудностями. На основе данных LMS строятся автоматические отчёты для преподавателей и администраторов, а также формируются уведомления о снижении активности или вероятности неуспешного прохождения курса. Для агрегации, хранения и анализа событийных данных с различных платформ применяются внешние системы учебной аналитики, осуществляющие интеграцию информации из LMS, электронных тестов, опросников, форумов и прочих цифровых следов обучающихся.

Для формирования отчётов, мониторинга и визуализации результатов образовательной аналитики используются инструменты бизнес-аналитики, такие как Power BI, Tableau, Qlik. С помощью этих платформ создаются интерактивные панели мониторинга, отражающие динамику академических показателей, посещаемости, активности в онлайн-курсах и результаты анкетирования. В образовательных организациях данные инструменты

применяются для оперативного принятия управленческих решений и выстраивания адаптивных стратегий поддержки студентов.

Работа с большими массивами событийных данных, поступающих с платформ массового открытого онлайн-обучения, ведётся посредством параллельных систем обработки, таких как Apache Spark. Для сбора и структурирования событийных данных об образовательной активности студентов используются стандарты Experience API и хранилища учебных записей (Learning Record Store), что позволяет организовать масштабируемый процесс хранения и подготовки данных для последующего машинного анализа.

Разработка адаптивных систем и индивидуализированных рекомендаций по обучению реализуется на основе специализированных библиотек и фреймворков для построения рекомендательных моделей. Среди них — TensorFlow Recommenders, LensKit, Surprise, которые используются для автоматического подбора образовательных траекторий, рекомендаций по выбору дисциплин, формированию индивидуальных учебных планов. Интеграция таких моделей в образовательные платформы позволяет учитывать индивидуальные предпочтения, уровень подготовки и динамику освоения материала каждым студентом.

Методы кластеризации в образовательной аналитике играют центральную роль в автоматическом выделении однородных групп студентов по совокупности параметров учебной активности, цифровых компетенций, стилей взаимодействия с ресурсами и результатов промежуточного тестирования. В отличие от традиционных методов сегментации, опирающихся на заранее определённые признаки или формальные критерии, кластерный анализ выявляет внутренние паттерны, сегменты и профили, отражающие реальные различия в образовательных практиках, вовлечённости, стилях освоения материала. Использование кластеризации оправдано при анализе многомерных данных, где необходимо учесть широкий спектр характеристик, включая посещаемость, участие в обсуждениях, сроки сдачи

заданий, взаимодействие с LMS, а также использование дополнительных сервисов и ресурсов. В практике аналитических исследований зафиксированы примеры применения k-средних для выделения сегментов студентов по активности во внешкольной работе, Fuzzy C-Means — для обнаружения профилей с размытыми границами между группами, агломеративной кластеризации — для построения иерархических структур и анализа динамики групп по мере изменения образовательной политики.

Кластеризация используется для диагностики академических рисков и раннего выявления студентов, испытывающих трудности с освоением программ, а также для выделения центров сетевой активности и изолированных групп. В образовательных исследованиях по моделям инверсного класса и смешанного обучения анализ активности на цифровых платформах позволяет выделять группы с разной степенью вовлечённости, на основании чего преподаватели и администрация разрабатывают адресные меры поддержки и корректируют содержание учебных курсов. Визуализация результатов кластеризации реализуется в форме тепловых карт, графов, интерактивных дашбордов, которые используются для принятия решений по персонализации образовательного процесса.

Методы построения профилей и индивидуализация сопровождения обучающихся опираются на результаты кластерного анализа и классификации, а также на средства визуализации ключевых образовательных индикаторов. В учебной аналитике полученные сегменты служат основой для дальнейшего внедрения персонализированных рекомендаций, разработки траекторий развития компетенций и адаптации учебных стратегий в зависимости от характеристик группы. Данные аналитические подходы становятся инструментом управления качеством образовательных программ и повышения эффективности работы образовательных организаций.

1.4 Кластеризация и классификация в образовательной аналитике: методы, сравнительный анализ и программные средства

В исследовании, посвящённом анализу анкет студентов и оценке уровня цифровизации образовательной среды, используется комплексная стратегия применения современных методов машинного обучения. Обработка массивов анкетных данных студентов осуществляется с использованием алгоритмов кластерного анализа, классификации, а также средств автоматической генерации индивидуализированных текстовых рекомендаций. Кластеризация служит основным инструментом для обнаружения скрытых сегментов в данных без необходимости предварительной категоризации наблюдений. В контексте образовательной аналитики данный метод позволяет выделять естественные группы студентов, обладающих сходными характеристиками цифровой активности, предпочитаемыми форматами работы с образовательными ресурсами и типичными паттернами взаимодействия с электронными платформами вуза. Применение кластерного анализа обеспечивает профилирование обучающихся на основании эмпирически выявленных моделей поведения, что служит базой для построения дифференцированных сценариев поддержки и развития цифровых компетенций.

Классификация, как метод обучения с учителем, реализуется на следующем этапе после формирования цифровых профилей с помощью кластеризации. Данная категория алгоритмов применяется для построения моделей, способных автоматически определять класс или категорию новых наблюдений на основе обучающей выборки, в которой каждая запись снабжена меткой. В образовательных исследованиях классификация используется для автоматизированного прогнозирования академической успешности, диагностики рисков отсева, идентификации студентов с потенциальной потребностью в поддержке, а также для решения задачи отнесения нового респондента к одному из ранее сформированных кластерных профилей

цифровизации. В рамках рассматриваемого проекта механизм классификации позволил реализовать технологию автоматического определения цифрового профиля студента на основании его индивидуальных ответов в анкете, что обеспечивает возможность масштабируемого и объективного профилирования при появлении новых данных.

Генерация индивидуальных рекомендаций осуществляется с привлечением моделей генеративного искусственного интеллекта, обеспечивающих формирование текстовых советов на естественном языке с учётом совокупности признаков цифрового поведения каждого студента. Персонализация советов достигается за счёт анализа результатов кластеризации, выявленных в анкетных данных закономерностей и автоматической передачи описания профиля в языковую модель. В проекте реализована интеграция с API отечественной генеративной языковой модели GigaChat, что позволяет формировать содержательные и релевантные рекомендации для студентов с различными цифровыми профилями без привлечения экспертов и ручного анализа.

Комплексная реализация методов анализа данных и построения моделей машинного обучения осуществляется с использованием современного инструментария языка Python. Для работы с табличными данными применяется библиотека Pandas, которая предоставляет средства для чтения, очистки, агрегации и трансформации структурированных массивов, полученных на основе результатов анкетирования студентов. Модули библиотеки Scikit-learn используются для построения и тестирования алгоритмов машинного обучения, в том числе кластеризации (k-means, иерархическая агломерация, нечеткие с-средние), классификации (решающие деревья, случайный лес, многослойные перцептроны), а также для проведения оценки качества моделей по стандартным метрикам. Для визуализации данных, промежуточных результатов обработки, анализа структуры кластеров и проверки корректности моделей используется библиотека Matplotlib, предоставляющая инструменты для построения различных видов графиков,

диаграмм распределения, карт признаков и прочих средств аналитического представления результатов.

Интеграция генеративных языковых моделей достигается посредством использования API GigaChat, что обеспечивает автоматическую генерацию текстовых рекомендаций по развитию цифровых навыков студентов, обладающих разными профилями цифровизации. В процессе построения такой системы индивидуальных рекомендаций программное обеспечение формирует структурированный запрос к языковой модели, включающий описание профиля, типичные характеристики цифрового поведения, потенциальные слабые и сильные стороны, выявленные на предыдущих этапах анализа. Полученный от языковой модели текст сохраняется и используется для предоставления студенту в индивидуальном отчёте, а также для накопления статистики по наиболее востребованным образовательным стратегиям.

В аналитической части работы программная реализация включает последовательное применение кластеризации для выделения профилей студентов, обучение классификатора для автоматического определения профиля на новых данных, визуализацию структуры данных и кластеров с помощью графиков, а также генерацию персонализированных рекомендаций на основе профиля, полученного в ходе анализа. Весь цикл анализа реализован в среде Python с использованием открытых библиотек и интеграции с внешними языковыми сервисами.

Вывод по главе 1

В первой главе последовательно изложены подходы к анализу образовательных данных с применением методов машинного обучения, акцентировано внимание на современных методах профилирования студентов в условиях цифровой трансформации высшей школы. Рассматривается понятие цифровизации в образовании как процесс интеграции информационно-коммуникационных технологий на всех уровнях организации учебной и административной деятельности, приводятся особенности отличия цифровизации от традиционных форм дистанционного обучения. Раскрывается структура программных инициатив и стратегических документов в сфере цифровой трансформации образования, описывается влияние государственной политики и международных докладов на внедрение цифровых инструментов, а также формирование новых моделей образовательных траекторий.

Дается характеристика ключевых категорий цифровых образовательных инструментов, включая системы управления обучением, электронные образовательные ресурсы, коммуникационные и коллаборационные сервисы, специализированные приложения для поддержки учебного процесса, инновационные технологии на базе искусственного интеллекта, расширенной и виртуальной реальности, а также электронные сертификаты и механизмы подтверждения образовательных достижений. Описывается эволюция инструментов цифровизации, анализируются технологические и организационные предпосылки для повышения доступности, гибкости и индивидуализации учебной среды.

Определяются базовые принципы машинного обучения и методологические этапы построения моделей, раскрываются особенности обучения с учителем и без учителя, формулируются задачи классификации и кластеризации, приводятся ключевые алгоритмы построения и оценки моделей, затрагиваются вопросы предотвращения переобучения и оценки

обобщающей способности. Описываются методы снижения размерности, поиска аномалий и анализа структуры данных, приводится классификация алгоритмов кластеризации, включая итеративные, иерархические, плотностные, графовые и вероятностные подходы. Детализируется применение методов кластеризации и классификации для сегментации обучающихся, анализа профилей и автоматизации диагностики на больших массивах анкетных данных.

Проводится анализ программного инструментария для образовательной аналитики: описываются функциональные возможности библиотек Scikit-learn, Pandas, Matplotlib, рассматриваются среды визуального моделирования и бизнес-аналитики, такие как Orange, Tableau, PowerBI, анализируются современные решения для параллельной обработки событийных данных, интеграции систем учебной аналитики с LMS и платформами массового онлайн-обучения. Отдельное внимание уделяется появлению фреймворков для генерации персонализированных рекомендаций, интеграции языковых моделей и искусственного интеллекта в образовательную среду.

ГЛАВА II. ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА АВТОМАТИЗИРОВАННЫЙ СИСТЕМЫ

2.1 Разведочный анализ данных (eda)

В рамках разведочного анализа анкетных данных студентов были проведены комплексные мероприятия, направленные на выявление структуры, оценку качества и изучение закономерностей исходной выборки. Начальный этап исследования включал импорт и первичную загрузку данных из файла Excel, содержащего ответы респондентов. Для корректной работы с числовыми характеристиками осуществлялась автоматическая конвертация строковых представлений чисел в числовой формат с целью предотвращения ошибок дальнейших статистических операций.

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import os
import textwrap
import numpy as np
import re
from scipy.stats import chi2_contingency

INPUT_PATH = "../data/dlia_studentov.xlsx"
OUTPUT_DIR = "../outputs/eda_students"
os.makedirs(OUTPUT_DIR, exist_ok=True)

def short_label(label, max_len=40):
    label = str(label)
    return textwrap.shorten(label, width=max_len, placeholder="...")

def safe_filename(label, max_len=40):
    label = str(label)
    label = re.sub(r'[^A-Za-zA-Яа-я0-9_]+', '_', label)
    return label[:max_len]

print("[1] Загрузка данных...")
df = pd.read_excel(INPUT_PATH)
print(f"Строк: {df.shape[0]}, Столбцов: {df.shape[1]}")
display(df.head())
```

Рисунок 1.1. Загрузка данных и просмотр первых строк анкеты

В ходе анализа структуры данных был выполнен поиск столбцов, отражающих факультетальную принадлежность студента. На основании найденного столбца построена визуализация распределения студентов по

факультетам, демонстрирующая неоднородность представительства различных подразделений в выборке.

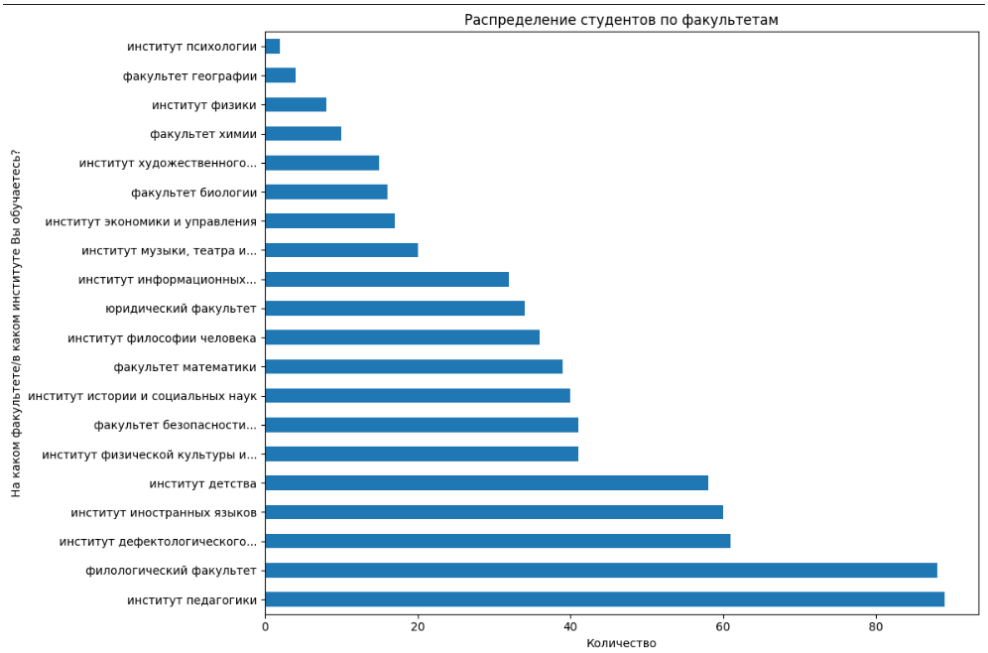


Рисунок 1.2. Распределение студентов по факультетам

Оценка полноты данных осуществлялась посредством анализа пропусков по всем признакам. Для признаков, обладающих пропущенными значениями, были построены гистограммы, отражающие долю пропусков по каждому из столбцов, что позволяет выявить потенциальные проблемные участки структуры анкеты.

```
missing = df.isnull().mean().sort_values(ascending=False)
missing_to_plot = missing[missing > 0]
if not missing_to_plot.empty:
    plt.figure(figsize=(10, 6))
    ax = missing_to_plot.plot(kind="barh", title="Доля пропусков по признакам")
    ax.set_yticklabels([short_label(x, 40) for x in missing_to_plot.index], fontsize=9)
    plt.tight_layout()
    plt.show()
else:
    print("Нет признаков с пропусками.")
✓ 0.0s
Нет признаков с пропусками.
```

Рисунок 1.3. Доля пропусков по признакам анкеты

Следующим этапом стала идентификация бинарных признаков, имеющих два дискретных значения (например, «да/нет», «использую/не использую»). Для повышения корректности анализа все подобные ответы были приведены к единой числовой шкале (0/1) посредством применения словаря соответствий.

Были выделены признаки, полностью удовлетворяющие бинарной природе, и произведён их количественный анализ.

```
binary_map = {
    "использую": 1, "не использую": 0,
    "использовать": 1, "не использовать": 0,
    "да": 1, "нет": 0,
    "Да": 1, "Нет": 0,
    "yes": 1, "no": 0,
    "Yes": 1, "No": 0,
    1: 1, 0: 0,
    "1": 1, "0": 0,
}
binary_cols = []
for col in df.columns:
    vals = df[col].dropna().unique()
    if all(str(x).strip().lower() in binary_map for x in vals):
        df[col] = df[col].map(lambda x: binary_map.get(str(x).strip().lower(), pd.NA)).astype(float)
        binary_cols.append(col)

print(f"Найдено бинарных признаков: {len(binary_cols)}")
```

✓ 0.0s

```
Найдено бинарных признаков: 19
```

Рисунок 1.4. Определение и перевод бинарных признаков анкеты

Для каждого бинарного признака рассчитана доля единичных значений, а также построена таблица частотных характеристик, что позволило выявить признаки с выраженным дисбалансом (более 95% или менее 5% положительных ответов), обозначенные как малоинформативные.

	feature	frac_ones	ones	zeros	total	min_frac
3	Необходим ли автоматический мониторинг присутс...	0.502110	357	354	711	0.497890
2	Был ли автоматический мониторинг присутствия с...	0.528833	376	335	711	0.471167
13	Была ли предусмотрена рефлексия (отзыв) после ...	0.645570	459	252	711	0.354430
15	Была ли предусмотрена рефлексия (отзыв) после ...	0.652602	464	247	711	0.347398
14	Необходима ли рефлексия (отзыв) после выполнен...	0.658228	468	243	711	0.341772
17	Было ли организовано взаимодействие с преподав...	0.738397	525	186	711	0.261603
16	Необходима ли рефлексия (отзыв) после завершен...	0.746835	531	180	711	0.253165
6	Были ли для каждого Практического задания разр...	0.776371	552	159	711	0.223629
8	Был ли встроенный электронный журнал прогресса...	0.793249	564	147	711	0.206751
11	Необходимо ли встраивать в электронный курс ви...	0.807314	574	137	711	0.192686
0	Был ли предусмотрен фидбек (отклик преподавате...	0.850914	605	106	711	0.149086
18	Необходимо ли организовывать взаимодействие с ...	0.870605	619	92	711	0.129395
5	Необходимо ли представлять материалы для практ...	0.880450	626	85	711	0.119550
12	Были ли встроенные в электронный курс тесты по...	0.890295	633	78	711	0.109705
7	Необходимы ли для каждого Практического задани...	0.904360	643	68	711	0.095640
4	Материалы, представленные для практического за...	0.914205	650	61	711	0.085795
9	Необходим ли встроенный электронный журнал про...	0.917018	652	59	711	0.082982
1	Необходим ли фидбек (отклик преподавателя на в...	0.931083	662	49	711	0.068917
10	Были ли встроенны в электронный курс видеолекции?	0.938115	667	44	711	0.061885

Рисунок 1.5. Характеристики бинарных признаков: доля единиц и информативность

Был выполнен отдельный анализ малоинформативных признаков, чьи значения практически не различаются между респондентами.

```
uninformative = bin_stats_df[(bin_stats_df["frac_ones"] < 0.05) | (bin_stats_df["frac_ones"] > 0.95)]
print("Малоинформативные бинарные признаки (перекос):")
display(uninformative)
✓ 0.0s
Малоинформативные бинарные признаки (перекос):
```

feature	frac_ones	ones	zeros	total	min_frac

Рисунок 1.6. Малоинформативные бинарные признаки с выраженным дисбалансом

Структура взаимосвязей между бинарными признаками была исследована с помощью boxplot-графика, а также корреляционной матрицы. Были определены пары признаков с максимальными абсолютными значениями корреляции, что может свидетельствовать о дублировании информации или выраженной сопряженности некоторых аспектов цифрового поведения студентов.

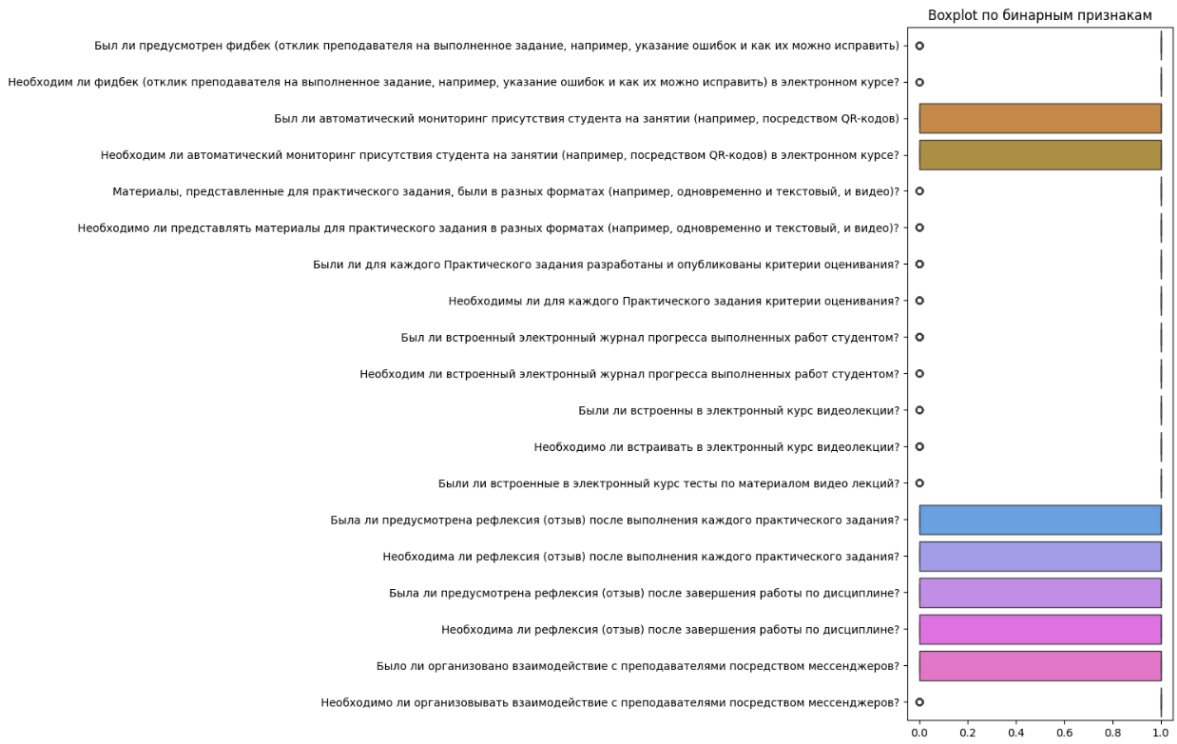


Рисунок 1.7. Boxplot по бинарным признакам анкеты

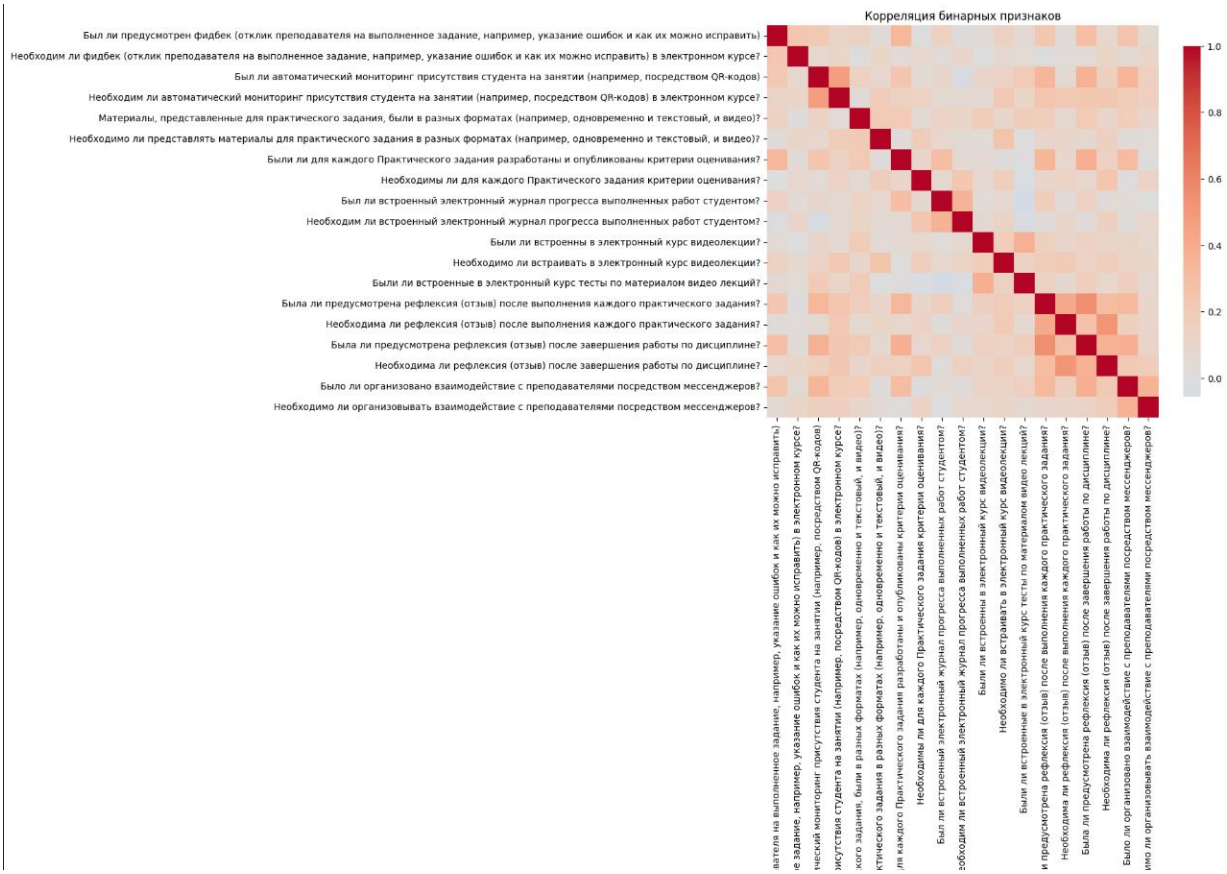


Рисунок 1.8. Матрица корреляции бинарных признаков

Для визуализации многообразия наиболее сбалансированных бинарных признаков построены парные диаграммы рассеяния (pairplot) по топ-5 признакам с максимальной информативностью.

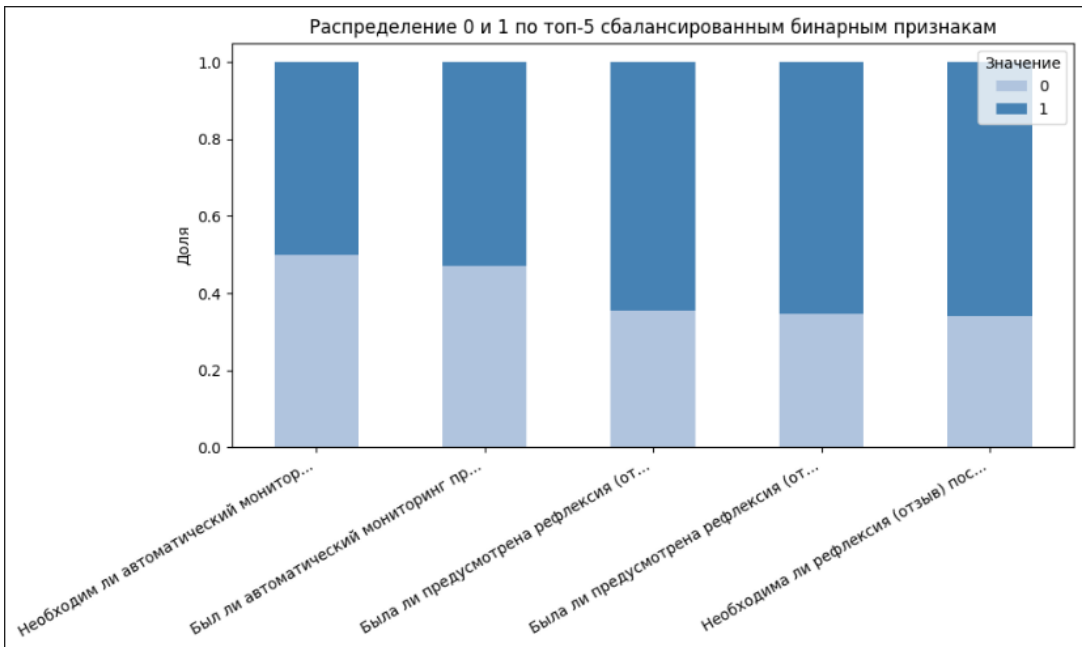


Рисунок 1.9. Pairplot по топ-5 сбалансированным бинарным признакам

В завершение была изучена взаимосвязь между бинарными признаками и факультетальной принадлежностью посредством расчёта коэффициента ассоциации Крамера (Cramér's V), что позволило количественно оценить силу связи между характеристиками цифрового поведения студентов и их учебным подразделением. Для наглядности результатов дополнительно строились столбчатые диаграммы, отражающие средние значения бинарных признаков по факультетам.

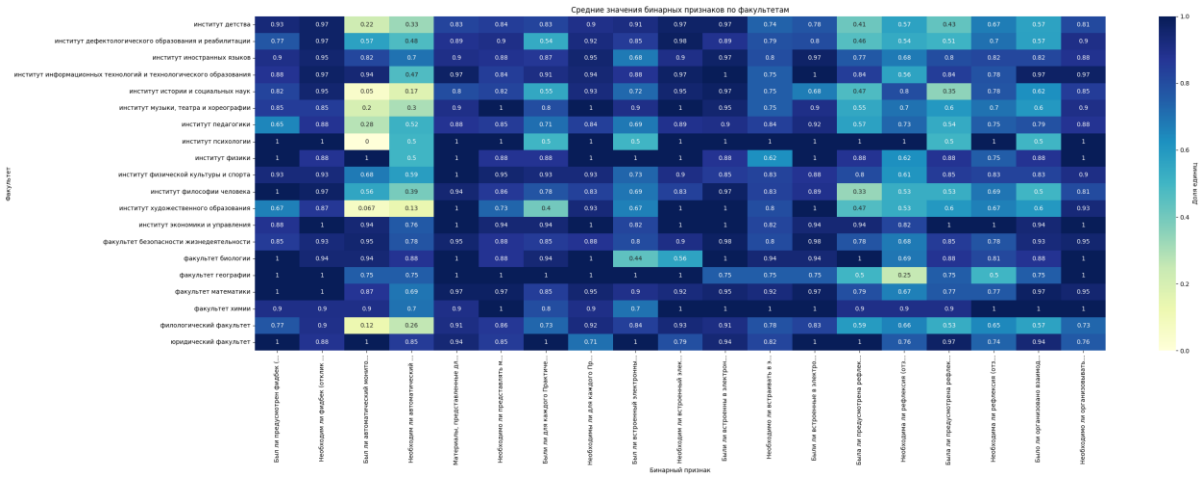


Рисунок 1.10. Оценка силы связи бинарных признаков с факультетами (Cramér's V)

В результате проведённого этапа детального анализа выявлены ключевые структурные и содержательные характеристики исходной выборки, определены признаки с максимальной информативностью, обнаружены коррелирующие и малоинформативные переменные, а также количественно оценено влияние факультетальной принадлежности на цифровое поведение студентов.

2.2 Предобработка данных и преобразование признаков

На этапе подготовки данных к дальнейшему анализу и моделированию выполнен ряд процедур, направленных на очистку, стандартизацию и преобразование исходной таблицы анкетных данных. Первоначальная загрузка данных осуществлялась из Excel-файла, содержащего ответы студентов, с последующим формированием структуры рабочей директории для хранения промежуточных и итоговых файлов.

```
import pandas as pd
import numpy as np
from sklearn.preprocessing import StandardScaler
from sklearn.feature_selection import VarianceThreshold
import umap.umap_ as umap
import os

INPUT_PATH = "../data/dlia_studentov.xlsx"
OUTPUT_PATH = "../data/students_preprocessed.csv"
BINARIES_PATH = "../data/students_binaries.csv"

os.makedirs("../data", exist_ok=True)

print("[1] Загрузка данных...")
df = pd.read_excel(INPUT_PATH)
print(f"Загружено строк: {df.shape[0]}, столбцов: {df.shape[1]}")
display(df.head())
```

Рисунок 2.1. Загрузка и первичный просмотр исходных анкетных данных студентов

В ходе первичной обработки данных были исключены все столбцы, содержащие временные метки, так как они не несут информативной нагрузки для последующего анализа. Для строковых признаков с пропущенными значениями производилась замена на стандартное значение "нет", а для числовых – заполнение средним арифметическим по признаку. После очистки из таблицы удалялись строки, полностью состоящие из пропусков, а также признаки, не имеющие достаточного заполнения (менее 30% валидных значений).

```
print("[2] Очистка данных...")
# Удаляем столбцы, содержащие 'время'
df = df[[col for col in df.columns if "время" not in col.lower()]]

# Заполняем пропуски: строки — "нет", числа — среднее
for col in df.columns:
    if df[col].dtype == object:
        df[col] = df[col].fillna("нет")
    else:
        df[col] = df[col].fillna(df[col].mean())

# Удаляем полностью пустые строки и признаки с >70% пропусков
df = df.dropna(how='all')
df = df.dropna(axis=1, thresh=0.3 * len(df))
print(f"После очистки: строк {df.shape[0]}, столбцов {df.shape[1]}")
```

✓ 0.0s

```
[2] Очистка данных...
После очистки: строк 711, столбцов 22
```

Рисунок 2.2. Очистка данных: удаление временных признаков, заполнение пропусков, фильтрация строк и столбцов

Для повышения интерпретируемости и совместимости с последующими этапами были стандартизированы бинарные признаки анкеты. Осуществлялось автоматическое приведение текстовых ответов, таких как "да", "нет", "использую", "не использую" и других, к единой числовой шкале 0/1 с помощью отображающего словаря.

```
print("[3] Бинаризация признаков...")
binary_map = {
    "использую": 1, "не использую": 0,
    "использовать": 1, "не использовать": 0,
    "да": 1, "нет": 0,
    "Да": 1, "Нет": 0,
    "yes": 1, "no": 0,
    "Yes": 1, "No": 0,
    1: 1, 0: 0,
    "1": 1, "0": 0,
}
for col in df.columns:
    if df[col].dtype == object:
        unique = df[col].dropna().unique()
        if all(x in binary_map for x in unique):
            df[col] = df[col].map(binary_map).astype(float)
```

✓ 0.0s

```
[3] Бинаризация признаков...
```

Рисунок 2.3. Преобразование бинарных признаков к числовому формату

Для организации корректного анализа факультетальных различий все обнаруженные столбцы, содержащие сведения о факультете или институте, были унифицированы в единую переменную "Факультет". При наличии дублирующих столбцов они исключались из итоговой таблицы. На основе полученного набора выделялись только признаки, значения которых

ограничены множеством {0, 1}, что позволяет сформировать отдельную таблицу бинарных признаков для последующего кластерного анализа.

```
faculty_col = [col for col in df.columns if "институт" in col.lower() or "факультет" in col.lower()]
if faculty_col:
    df["Факультет"] = df[faculty_col[0]]
    # Удаляем дубли, если есть
    df = df.drop(columns=faculty_col[1:])

print("[BINARIES] Отбор только бинарных признаков...")
# Бинарные признаки — только те, где уникальные значения ⊆ {0,1}
binaries = []
for col in df.columns:
    vals = set(df[col].dropna().unique())
    if vals <= {0, 1}:
        binaries.append(col)
bin_cols = binaries.copy()
if 'Факультет' in df.columns and 'Факультет' not in binaries:
    bin_cols.append('Факультет')
df_binaries = df[bin_cols]
df_binaries.to_csv(BINARIES_PATH, index=False)
print(f"[BINARIES] Сохранено {len(binaries)} бинарных признаков в {BINARIES_PATH}")
display(df_binaries.head())
```

Рисунок 2.4. Формирование и сохранение набора бинарных признаков

Для числовых признаков осуществлялось масштабирование с использованием стандартного нормировщика (StandardScaler). Признаки с крайне низкой дисперсией (менее 0.01) исключались посредством отбора по порогу дисперсии (VarianceThreshold), что обеспечивает удаление почти константных столбцов и уменьшение размерности признакового пространства.

```
numeric_df = df.select_dtypes(include=[np.number])
print(f"Числовых признаков для анализа: {numeric_df.shape[1]}")

print("[4] Масштабирование и отбор признаков...")
scaler = StandardScaler()
X_scaled = scaler.fit_transform(numeric_df)
selector = VarianceThreshold(threshold=0.01)
X_selected = selector.fit_transform(X_scaled)
print(f"После отбора осталось признаков: {X_selected.shape[1]}")
```

```
✓ 0.0s
Числовых признаков для анализа: 20
[4] Масштабирование и отбор признаков...
После отбора осталось признаков: 20
```

Рисунок 2.5. Масштабирование и отбор числовых признаков

Понижение размерности высокоразмерного пространства реализовывалось посредством алгоритма UMAP, который позволяет перейти к двумерному представлению данных, сохраняя наиболее значимые топологические и кластерные структуры. Полученные двумерные координаты добавлялись к основной таблице.

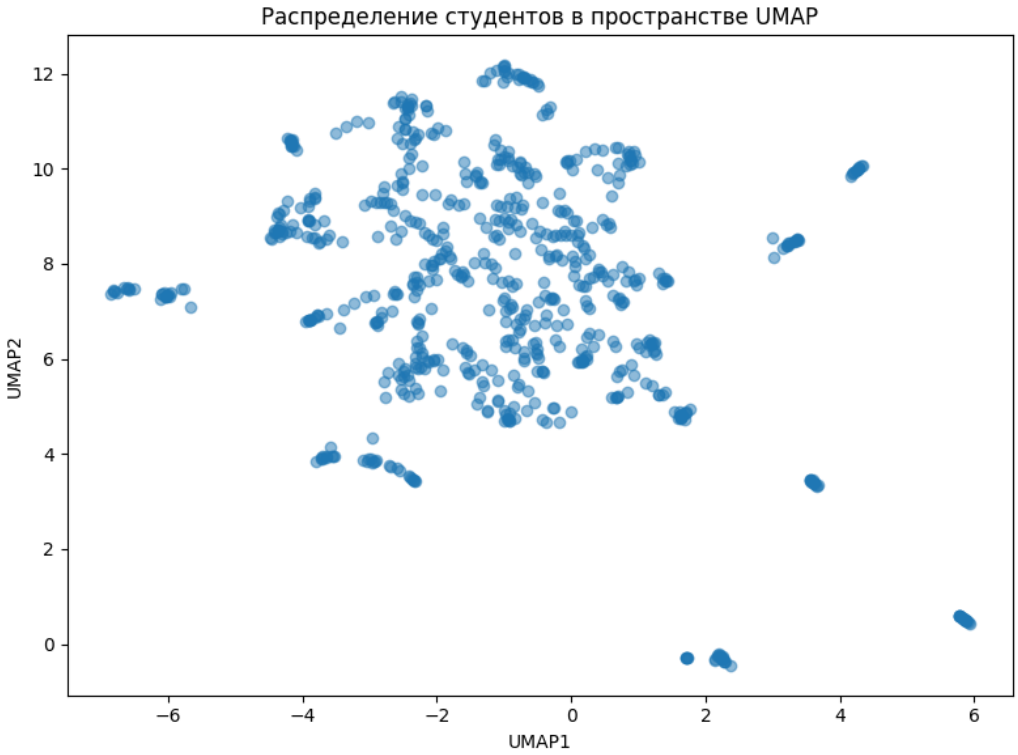


Рисунок 2.6. Визуализация двумерного пространства UMAP по результатам понижения размерности

Итоговые таблицы, включающие как исходные, так и бинарные и редуцированные данные, сохранялись на диск для обеспечения воспроизводимости дальнейших этапов анализа и возможности повторного использования в задачах моделирования.

```
print("[6] Сохранение результатов...")
df_result.to_csv(OUTPUT_PATH, index=False)
print(f"Сохранено в {OUTPUT_PATH}")
display(df_result.head())
```

Рисунок 2.7. Сохранение итогового датасета с результатами преобразования и понижения размерности

На данном этапе выполнены все процедуры по приведению данных к формату, оптимальному для последующих методов кластеризации и построения предиктивных моделей.

2.3 Кластеризация студентов (umap-признаки и матрица говера)

После завершения этапа предобработки данных следующим логическим

В ходе реализации кластерного анализа цифровых профилей студентов применялись методы, учитывающие специфику бинарных и категориальных данных. Ключевым этапом явилось формирование матрицы расстояний Говера, позволяющей корректно измерять попарные различия между объектами с бинарными признаками. Для расчёта матрицы использовались только бинарные признаки, выделенные на предыдущем этапе подготовки данных.

```
if "Факультет" in df.columns:
    df_features = df.drop(columns=["Факультет"])
else:
    df_features = df

print("[2] Расчёт матрицы расстояний Говера по бинарным признакам...")
distance_matrix = gower.gower_matrix(df_features)
np.save(OUTPUT_MATRIX, distance_matrix)
print(f"Матрица расстояний сохранена в {OUTPUT_MATRIX}")
print(f"Размерность матрицы: {distance_matrix.shape}")
```

✓ 0.0s

```
[2] Расчёт матрицы расстояний Говера по бинарным признакам...
Матрица расстояний сохранена в ../data/gower_distance_matrix.npy
Размерность матрицы: (711, 711)
```

Рисунок 3.1. Расчёт и сохранение матрицы попарных расстояний Говера для бинарных признаков

Для наглядной иллюстрации паттернов схожести и различий между студентами по бинарным характеристикам строилась тепловая карта верхней части матрицы (50×50), где каждое значение отражает степень различия между двумя студентами: от полного совпадения (0) до максимального различия (1).

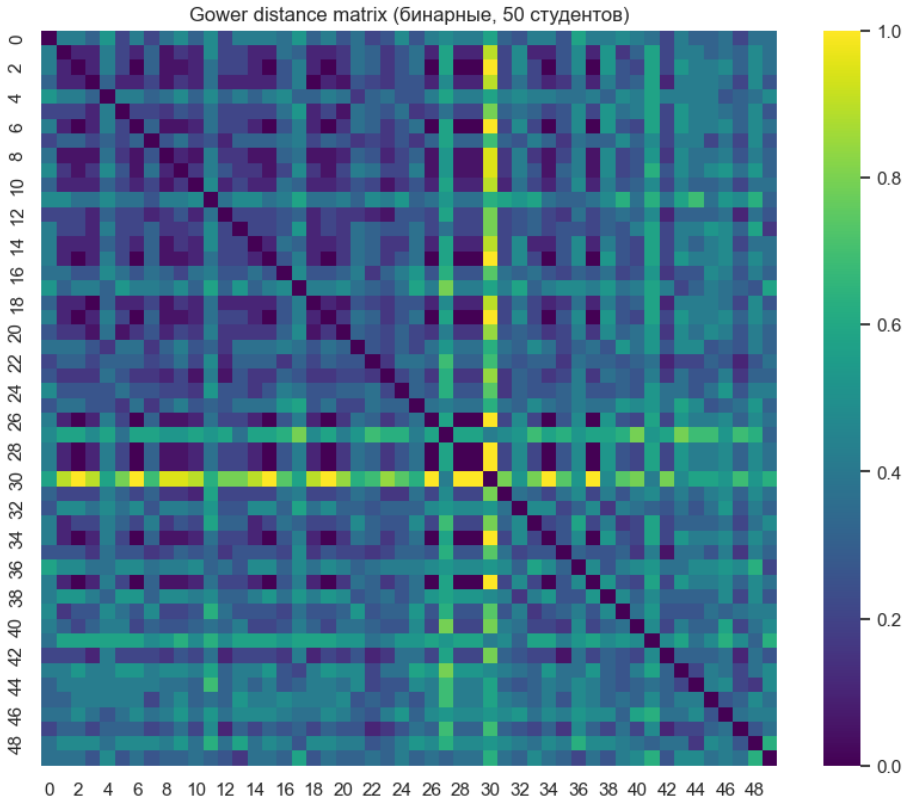


Рисунок 3.2. Фрагмент тепловой карты матрицы расстояний Говера (первые 50 студентов)

Следующим шагом стала реализация кластеризации, для чего были использованы два типа признаков: двумерные координаты, полученные посредством алгоритма UMAP, и сама матрица расстояний Говера. По UMAP-признакам выполнялись два вида кластеризации — агломеративная (иерархическая) и нечеткая c-means, что обеспечивало как жёсткое, так и мягкое разбиение на кластеры. Каждый подход проверялся на диапазоне числа кластеров от 2 до 6, а оптимальность определялась по метрикам силуэта и индексу Дэвиса–Болдуина.

```
[2] Кластеризация по UMAP-признакам...

=== Agglomerative (UMAP) ===
k=2: silhouette=0.576, db=0.562
k=3: silhouette=0.306, db=1.073
k=4: silhouette=0.333, db=0.886
k=5: silhouette=0.359, db=0.848
k=6: silhouette=0.388, db=0.785

=== Fuzzy C-Means (UMAP) ===
k=2: silhouette=0.360, db=1.074
k=3: silhouette=0.368, db=0.973
k=4: silhouette=0.381, db=0.821
k=5: silhouette=0.365, db=0.826
k=6: silhouette=0.340, db=0.859
```

Рисунок 3.3. Кластеризация по UMAP-признакам: результаты агломеративного и Fuzzy C-Means кластеризации с оценкой метрик качества

Для бинарных признаков посредством матрицы расстояний Говера применялась агломеративная кластеризация с предвычисленной матрицей расстояний и средним типом связи. Для каждого разбиения также вычислялся коэффициент силуэта.

```
print("[3] Загрузка матрицы расстояний Говера...")
distance_matrix = np.load(GOWER_MATRIX_PATH)
results_gower = []
for n_clusters in CLUSTER_RANGE:
    agglom_gower = AgglomerativeClustering(n_clusters=n_clusters, metric='precomputed', linkage='average')
    labels_gower = agglom_gower.fit_predict(distance_matrix)
    sil_gower = silhouette_score(distance_matrix, labels_gower, metric='precomputed')
    results_gower.append({
        "n_clusters": n_clusters,
        "silhouette": sil_gower,
        "labels": labels_gower.copy()
    })
    print(f"GOWER: k={n_clusters}: silhouette={sil_gower:.3f}")
✓ 0.0s
[3] Загрузка матрицы расстояний Говера...
GOWER: k=2: silhouette=0.394
GOWER: k=3: silhouette=0.345
GOWER: k=4: silhouette=0.329
GOWER: k=5: silhouette=0.268
GOWER: k=6: silhouette=0.244
```

Рисунок 3.4. Кластеризация по матрице Говера: silhouette по различным k

На основании максимальных значений силуэта для каждого метода к данным были присвоены метки кластеров. Для нечеткой кластеризации с k=3 кластерам присваивались интерпретируемые названия профилей, отражающие уровни вовлеченности студентов в цифровую образовательную среду.

```
best_ac = max(results_agglom, key=lambda x: x["silhouette"])
best_fcm = max(results_fcm, key=lambda x: x["silhouette"])
best_gower = max(results_gower, key=lambda x: x["silhouette"])

df["Кластер_AC"] = best_ac["labels"]
df["Кластер_FCM"] = best_fcm["labels"]
df["Кластер_Gower"] = best_gower["labels"]

# Названия профилей (если k=3 для FCM)
if best_fcm["n_clusters"] == 3:
    cluster_names = {0: "Цифровые скептики", 1: "Умеренно вовлечённые", 2: "Цифровые энтузиасты"}
    df["Профиль"] = df["Кластер_FCM"].map(cluster_names)
else:
    df["Профиль"] = df["Кластер_FCM"].apply(lambda x: f"Профиль {x}")
```

Рисунок 3.5. Присвоение итоговых меток кластеров и профилей студентов

Итоговая таблица с признаками, кластерными метками и профилями сохранялась для последующего анализа и визуализации.

```
print("[4] Сохранение результатов кластеризации...")
df.to_csv(OUTPUT_PATH, index=False)
print(f"Сохранено: {OUTPUT_PATH}")
display(df.head())
```

Рисунок 3.6. Итоговая таблица с метками кластеров и профилей

Для визуальной интерпретации результатов кластеризации строились диаграммы рассеяния студентов в пространстве UMAP, окрашенные по полученным профилям и меткам кластеров, а также линии зависимости silhouette score от числа кластеров для каждого метода. Дополнительно проведён кросс-анализ совпадения между разбиениями по разным методикам посредством таблицы пересечений.

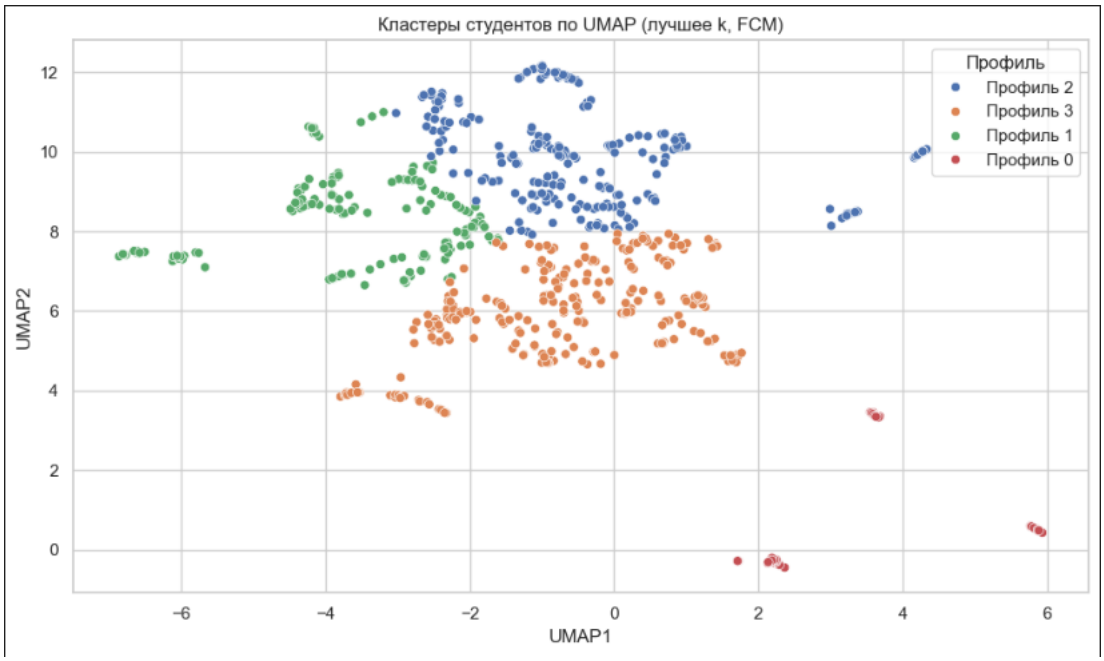


Рисунок 3.7. Визуализация кластеров по UMAP (цвет: профиль)

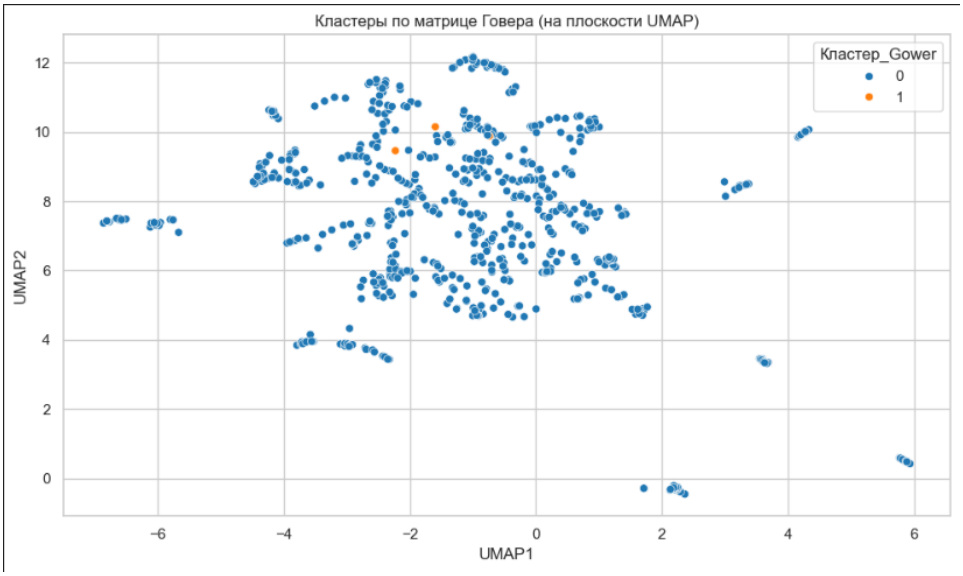


Рисунок 3.8. Визуализация кластеров по матрице Говера на плоскости UMAP

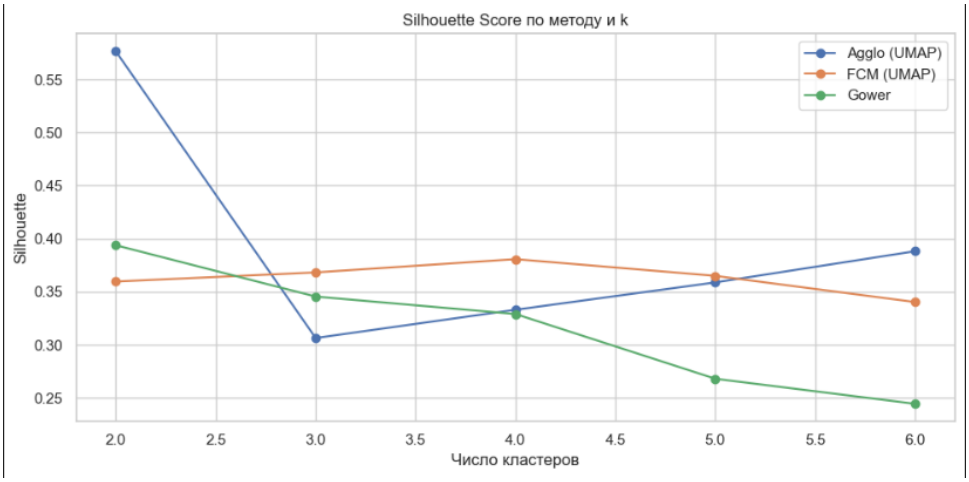


Рисунок 3.9. Зависимость silhouette score от числа кластеров для разных методов

```
crosstab = pd.crosstab(df["Кластер_FCM"], df["Кластер_Gower"])
display(crosstab)
```

✓ 0.0s

Кластер_Gower	0	1
Кластер_FCM		
0	49	0
1	169	0
2	224	3
3	266	0

Рисунок 3.10. Пересечение кластеров FCM и Говера (таблица сопряжённости)

Результаты этого этапа обеспечили основу для дальнейшей интерпретации цифровых профилей студентов, их сравнительного анализа и построения моделей автоматической классификации.

2.4 Оценка качества кластеризации

Оценка качества разбиения студентов на кластеры осуществлялась с использованием стандартных метрик кластеризации, обеспечивающих формальное измерение разделимости и однородности полученных групп. Были выбраны коэффициент силуэта (Silhouette Score) и индекс Дэвиса–Болдуина (Davies-Bouldin Index), позволяющие комплексно оценивать как внутреннюю сплочённость кластеров, так и их различимость друг от друга. Анализ охватывал разбиения, полученные агломеративной кластеризацией и методом Fuzzy C-Means по пространству UMAP, а также агломеративной кластеризацией по матрице расстояний Говера.

```
import pandas as pd
import numpy as np
from sklearn.metrics import silhouette_score, davies_bouldin_score
import matplotlib.pyplot as plt
import os

INPUT_PATH = "../data/students_with_clusters.csv"
GOWER_MATRIX_PATH = "../data/gower_distance_matrix.npy"
OUTPUT_DIR = "../outputs/evaluate_clustering"
os.makedirs(OUTPUT_DIR, exist_ok=True)

print("[1] Загрузка данных с кластерами и признаками...")
df = pd.read_csv(INPUT_PATH)
X = df[["UMAP1", "UMAP2"]].values
display(df.head())
```

Рисунок 4.1. Загрузка итоговой таблицы с метками кластеров и координатами UMAP

Для кластеров, полученных в пространстве UMAP, рассчитывались обе метрики — Silhouette Score и Davies-Bouldin Index. Для кластеров, полученных по матрице Говера, рассчитывался только коэффициент силуэта с использованием предвычисленной матрицы расстояний.

```
results = {}

# Оценка кластеров Agglomerative и FCM на UMAP
for label_col in ["Кластер_AC", "Кластер_FCM"]:
    if label_col in df.columns:
        labels = df[label_col].values
        sil = silhouette_score(X, labels)
        db = davies_bouldin_score(X, labels)
        results[label_col] = {"Silhouette": sil, "Davies-Bouldin": db}
        print(f"{label_col}: Silhouette={sil:.3f}, Davies-Bouldin={db:.3f}")
```

```
✓ 0.0s
Кластер_AC: Silhouette=0.576, Davies-Bouldin=0.562
Кластер_FCM: Silhouette=0.381, Davies-Bouldin=0.821
```

```
if "Кластер_Gower" in df.columns:
    labels_gower = df["Кластер_Gower"].values
    gower_distance_matrix = np.load(GOWER_MATRIX_PATH)
    sil_gower = silhouette_score(gower_distance_matrix, labels_gower, metric='precomputed')
    results["Кластер_Gower"] = {"Silhouette": sil_gower, "Davies-Bouldin": np.nan}
    print(f"Кластер_Gower: Silhouette Score (Gower): {sil_gower:.3f}")
```

```
✓ 0.0s
Кластер_Gower: Silhouette Score (Gower): 0.394
```

Рисунок 4.2. Расчёт метрик качества кластеризации для каждого метода

Визуальное сравнение значений Silhouette Score между методами представлено в виде столбчатой диаграммы. Чем выше значение Silhouette, тем более отчётливо выделены кластеры.

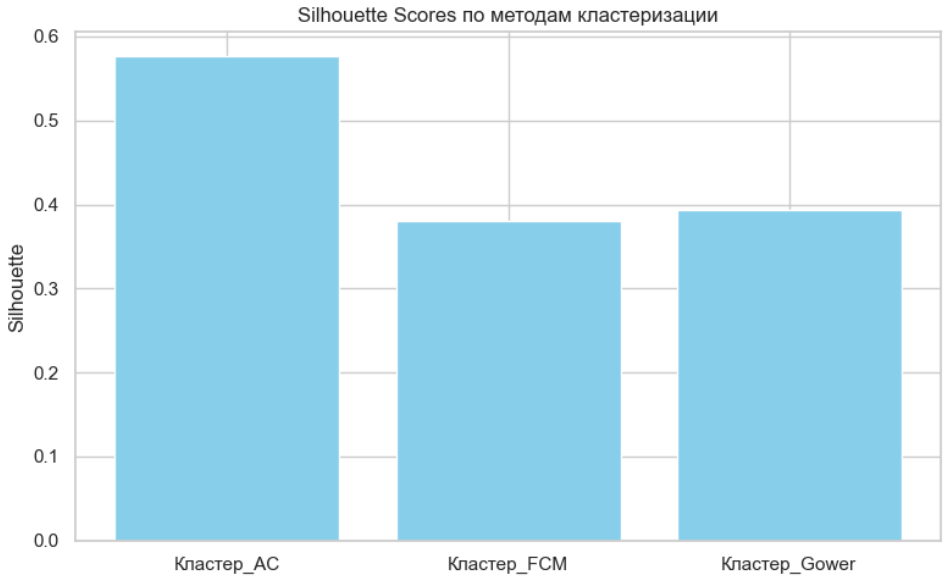


Рисунок 4.3. Сравнение Silhouette Score для разных методов кластеризации

Для методов, поддерживающих индекс Дэвиса–Болдуина, значения также были отображены на соответствующей диаграмме. Более низкое значение этой метрики свидетельствует о более качественном разбиении.

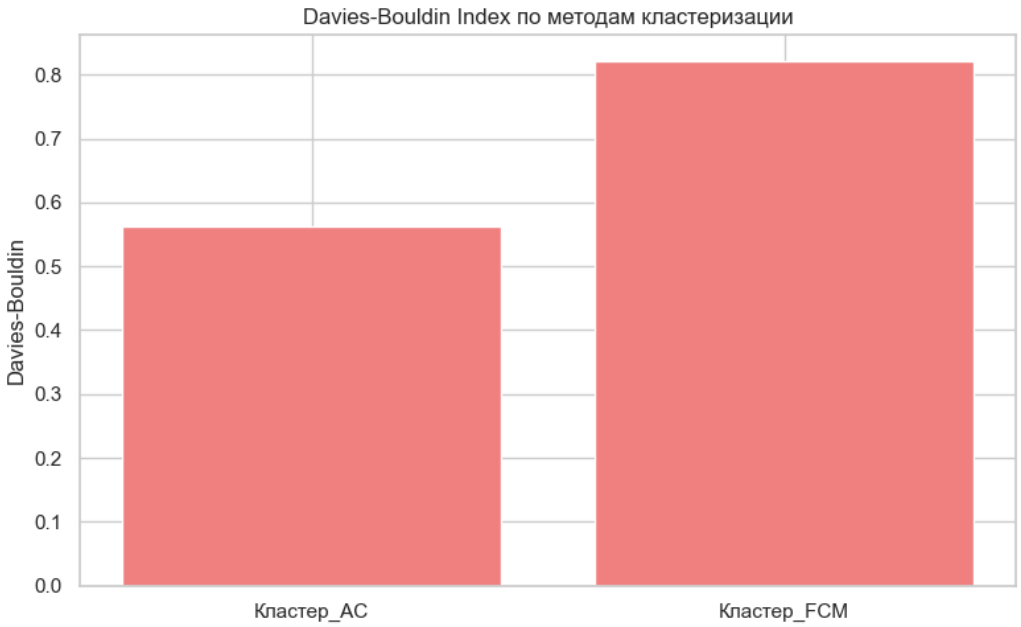
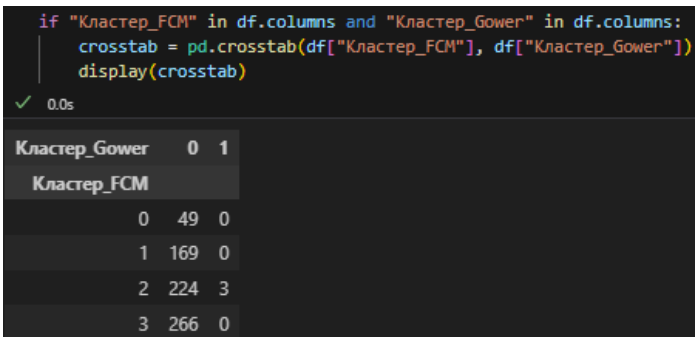


Рисунок 4.4. Сравнение Davies-Bouldin Index для разных методов кластеризации

Для анализа согласованности между разными алгоритмами проводился кросс-анализ кластеров по FCM (UMAP) и по Говеру — строилась таблица сопряжённости, отображающая, сколько объектов попало в пересечение между каждым парой кластеров двух методов.



```
if "Кластер_FCM" in df.columns and "Кластер_Gower" in df.columns:
    crosstab = pd.crosstab(df["Кластер_FCM"], df["Кластер_Gower"])
    display(crosstab)
```

✓ 0.0s

Кластер_Gower	0	1
Кластер_FCM		
0	49	0
1	169	0
2	224	3
3	266	0

Рисунок 4.5. Таблица пересечений кластеров FCM и Говера

Результаты оценки метрик свидетельствуют о приемлемом разделении кластеров по всем методам, однако максимальное значение Silhouette Score наблюдалось для агломеративной кластеризации по UMAP. Наблюдение за значениями индекса Дэвиса–Болдуина также подтвердило хорошее качество кластеризации. Кросстаб помог идентифицировать пересечения и различия разбиений, а также подтвердил устойчивость найденных профилей при использовании различных подходов.

2.5 Интерпретация и профилирование кластеров по бинарным признакам

Для разработанной автоматизированной системы генерации и проверки заданий в качестве входных данных требуются фамилия, имя обучающегося и цифра для выбора действия: «1» для генерации файла задания и «2» для его проверки (рис. 2.19).

В целях содержательной интерпретации и профилирования выделенных кластеров студентов был проведён комплексный анализ распределения бинарных признаков внутри каждого профиля. Исходные данные включали для каждого студента информацию о принадлежности к определённому кластеру, факультету, а также о значениях бинарных характеристик цифрового поведения.

```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import os
import seaborn as sns
import textwrap

INPUT_PATH = "../outputs/cluster_students_gower_binaries/students_with_clusters_binaries.csv"
CLUSTER_LABEL = "Кластер_Gower"
FACULTY_LABEL = "Факультет"
OUTPUT_DIR = "../outputs/cluster_profiles_gower"
os.makedirs(OUTPUT_DIR, exist_ok=True)

def short_label(label, max_len=30):
    label = str(label)
    return textwrap.shorten(label, width=max_len, placeholder="...")
```

Для анализа были автоматически выделены все бинарные признаки, не относящиеся к идентификаторам кластера и факультета. Для каждого кластера вычислялась доля студентов, обладающих тем или иным признаком. Это позволило получить матрицу профилей кластеров по основным характеристикам цифровой среды.

```
print("[1] Загрузка данных...")
df = pd.read_csv(INPUT_PATH)
binary_cols = [
    col for col in df.columns
    if col not in [CLUSTER_LABEL, FACULTY_LABEL]
    and set(df[col].dropna().unique()).issubset({0,1})
]
print(f"Найдено бинарных признаков: {len(binary_cols)}")

df.head().to_csv(os.path.join(OUTPUT_DIR, "students_with_clusters_binaries_head.csv"), index=False)
```

✓ 0.0s

```
[1] Загрузка данных...
Найдено бинарных признаков: 19
```

5.3 Групповая статистика по кластерам

```
profile_table = df.groupby(CLUSTER_LABEL)[binary_cols].mean().T
display(profile_table.head())
profile_table.to_csv(os.path.join(OUTPUT_DIR, "cluster_profile_table.csv"))
print(f"Таблица профилей кластеров сохранена ({profile_table.shape[0]} признаков).")
```

✓ 0.0s

Кластер_Gower	0	1
Был ли предусмотрен фидбек (отклик преподавателя на выполненное задание, например, указание ошибок и как их можно исправить)	0.850282	1.000000
Необходим ли фидбек (отклик преподавателя на выполненное задание, например, указание ошибок и как их можно исправить) в электронном курсе?	0.932203	0.666667
Был ли автоматический мониторинг присутствия студента на занятии (например, посредством QR-кодов)	0.531073	0.000000
Необходим ли автоматический мониторинг присутствия студента на занятии (например, посредством QR-кодов) в электронном курсе?	0.504237	0.000000
Материалы, представленные для практического задания, были в разных форматах (например, одновременно и текстовый, и видео)?	0.915254	0.666667

```
Таблица профилей кластеров сохранена (19 признаков).
```

Рисунок 5.1. Средние значения бинарных признаков по кластерам (матрица профиля кластеров)

Для выявления наиболее характерных различий между кластерами был рассчитан абсолютный размах долей для каждого признака — разница между максимальным и минимальным средним значением признака по всем профилям. Признаки с максимальным размахом были определены как ключевые для интерпретации различий между группами студентов. Визуализация этих признаков представлена в виде barplot-графика.

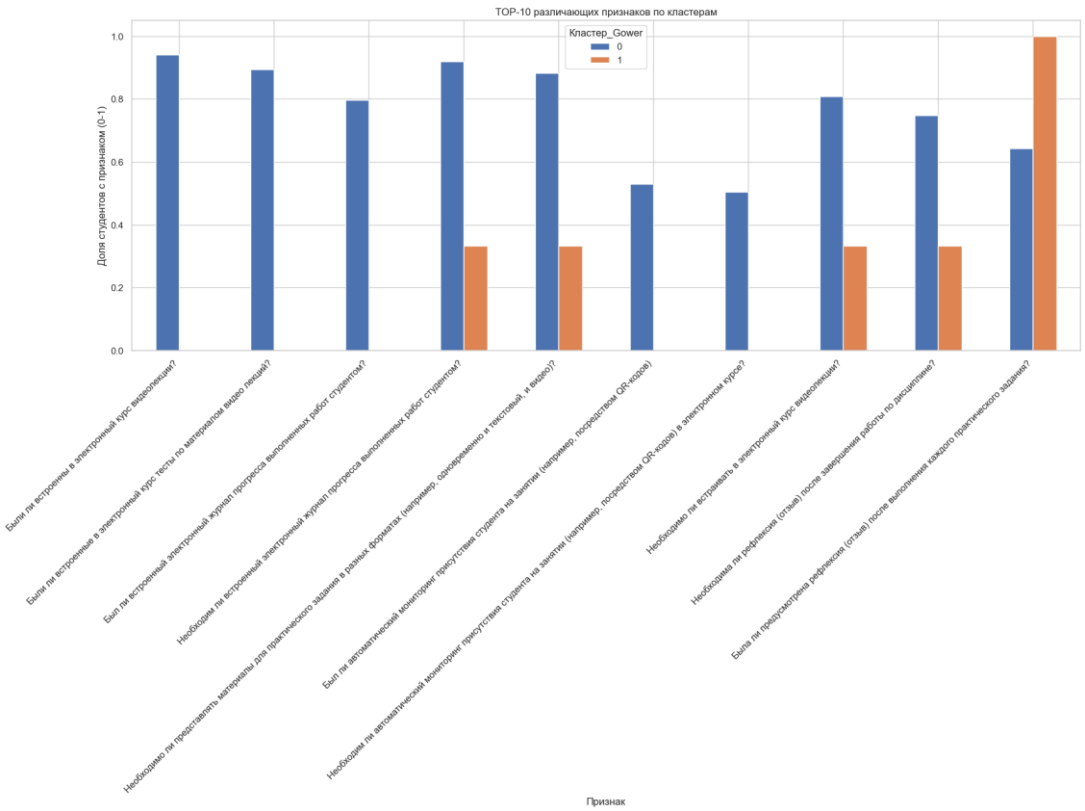


Рисунок 5.2. Barplot: топ-10 различающих бинарных признаков между кластерами

Для комплексной визуализации структурных различий между профилями студентов был построен радиальный график (radar plot), отражающий доли выраженности топовых признаков по каждому кластеру. Подобная форма визуализации позволяет сразу увидеть "силуэт" каждого цифрового профиля.

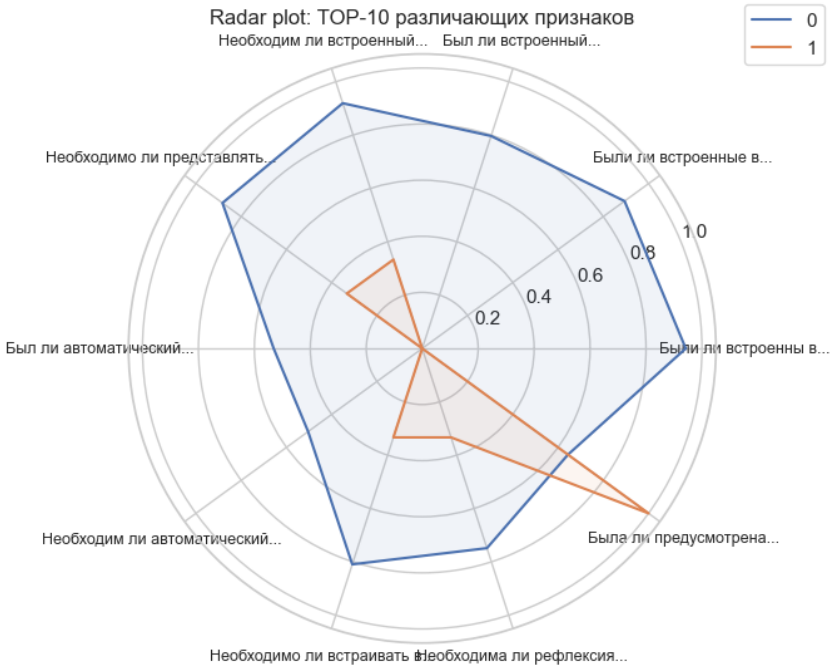


Рисунок 5.3. Радиальная диаграмма (radar plot) по топ-10 различающих признаков профилей кластеров

Для подготовки интерпретируемых текстовых описаний каждого профиля был сформирован автоматический шаблон. В нем для каждого кластера перечислялись наиболее выраженные и наименее выраженные признаки, что облегчает содержательную интерпретацию полученных групп и может служить основой для назначения содержательных названий цифровых профилей.

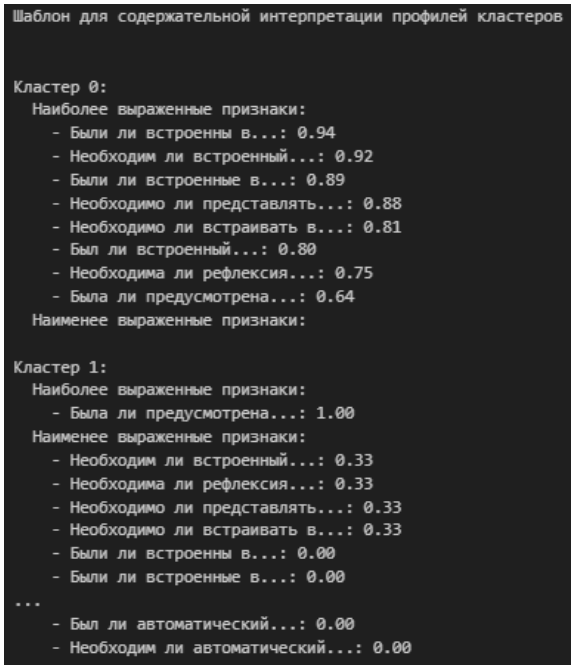


Рисунок 5.4. Автоматически сформированный шаблон интерпретации профилей кластеров

Проведённый анализ позволил не только выделить наиболее релевантные признаки для каждого цифрового профиля, но и заложил основу для дальнейшей содержательной интерпретации результатов, в том числе для построения рекомендаций и последующего внедрения результатов в образовательную аналитику.

2.6 Полярные (radar) диаграммы профилей кластеров

Для визуализации различий между цифровыми профилями студентов использовались полярные (radar) диаграммы, которые позволяют отразить средние значения бинарных признаков для каждого выделенного кластера на единой круговой сетке. Такой подход предоставляет интуитивно понятный способ сравнения ключевых характеристик профилей, выявленных в процессе кластерного анализа.

На предварительном этапе осуществлялось приведение исходных ответов к единому бинарному формату (0/1) на основании словаря сопоставления, который покрывает типовые варианты ответа («да/нет», «использую/не использую», «+/-» и аналогичные). Были автоматически выделены все признаки, удовлетворяющие бинарному критерию.

```
print("[2] Подготовка бинарных признаков...")
binary_map = {
    "использую": 1, "не использую": 0,
    "использовать": 1, "не использовать": 0,
    "да": 1, "нет": 0,
    "+": 1, "-": 0,
    "yes": 1, "no": 0
}
binary_cols = []

for col in df_raw.columns:
    unique_vals = df_raw[col].dropna().unique()
    if len(unique_vals) <= 3 and all(str(x).strip().lower() in binary_map for x in unique_vals):
        df_raw[col] = df_raw[col].map(lambda x: binary_map.get(str(x).strip().lower(), np.nan))
        binary_cols.append(col)

print(f"Найдено бинарных признаков: {len(binary_cols)}")
if not binary_cols:
    raise ValueError("Не найдено бинарных признаков для анализа профилей.")
✓ 0.0s

[2] Подготовка бинарных признаков...
Найдено бинарных признаков: 19
```

Рисунок 6.1. Определение бинарных признаков в исходных данных

На следующем этапе метки профиля, полученные по итогам кластеризации, были совмещены с исходной таблицей бинарных признаков.

Для каждого профиля (кластера) были рассчитаны средние значения бинарных признаков — это отражает долю студентов в кластере, отмечающих тот или иной признак. Далее, для каждой группы строилась индивидуальная полярная диаграмма, где каждая ось соответствует отдельному признаку, а значение по радиусу отражает частоту выраженности признака в профиле.

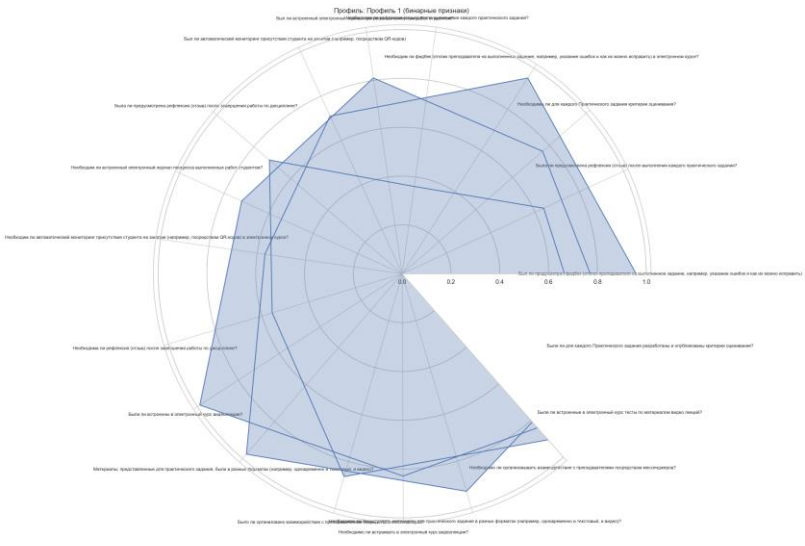


Рисунок 6.3. Полярная (radar) диаграмма: средние значения бинарных признаков для одного профиля студентов

Каждая из полученных диаграмм даёт наглядное представление о том, по каким признакам цифрового поведения тот или иной кластер отличается от остальных. Значения, максимально удалённые от центра, соответствуют наиболее характерным чертам данного профиля (например, высокая доля использования ИИ или цифровых платформ). Это облегчает интерпретацию профилей и выявление ключевых различий между группами студентов.

2.7 Построение и оценка классификатора профиля

Для решения задачи автоматической классификации цифрового профиля студента была построена серия моделей, призванных предсказывать принадлежность к кластеру на основе исходных признаков анкеты. В качестве целевой переменной выступала метка профиля, определённая по результатам Fuzzy C-Means кластеризации.

В качестве исходных переменных использовались числовые и бинарные признаки анкеты, за исключением признаков, непосредственно связанных с результатами кластеризации и координатами в пространстве UMAP. После отбора и масштабирования признаков применялся фильтр по порогу дисперсии, что позволило исключить малозначимые переменные.

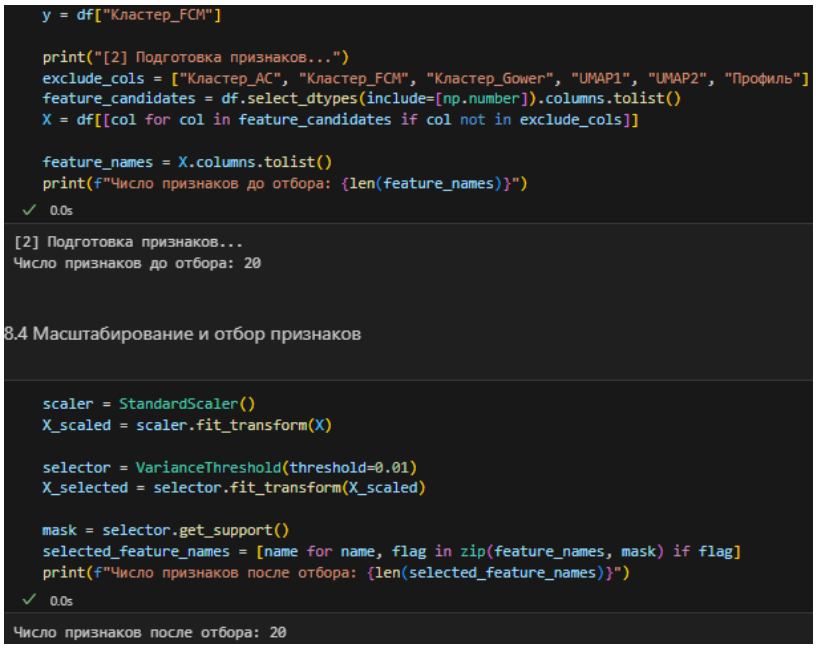


Рисунок 7.2. Предобработка: масштабирование и отбор информативных признаков

Множество данных было разделено на обучающую и тестовую выборки с сохранением пропорций классов, что обеспечивает корректную оценку качества моделей.

Для задачи мультиклассовой классификации обучались три типа моделей: решающее дерево, случайный лес и многослойный персептрон (MLP). Для

каждой модели рассчитывались метрики качества (precision, recall, f1-score) и средний F1_macro по 5-кратной кросс-валидации.

```
>>> Обучение модели DecisionTree...
              precision    recall  f1-score   support

           0       0.69      0.90      0.78        10
           1       0.82      0.82      0.82        34
           2       0.84      0.83      0.84        46
           3       0.78      0.75      0.77        53

    accuracy                           0.80       143
   macro avg       0.79      0.83      0.80       143
weighted avg       0.81      0.80      0.80       143

F1_macro (5-fold CV): 0.812 ± 0.094
Модель DecisionTree сохранена в model_DecisionTree.pkl

>>> Обучение модели RandomForest...
              precision    recall  f1-score   support

           0       0.91      1.00      0.95        10
           1       0.91      0.94      0.93        34
           2       0.96      0.96      0.96        46
           3       0.96      0.92      0.94        53
...
weighted avg       0.94      0.94      0.94       143

F1_macro (5-fold CV): 0.904 ± 0.052
```

Рисунок 7.4. Качество классификаторов: отчёты и кросс-валидация по F1_macro

Наилучшее качество демонстрировал случайный лес (RandomForest), что подтверждалось высокими значениями F1_macro и устойчивостью метрик на кросс-валидации. Для данной модели дополнительно анализировалась важность признаков — строились barplot-графики для топ-10 наиболее значимых переменных, что даёт возможность объяснить, какие характеристики анкеты вносят наибольший вклад в определение профиля.

```python
# Анализ важности признаков для RandomForest
if name == "RandomForest":
    importances = model.feature_importances_
    feat_imp = pd.Series(importances, index=selected_feature_names)
    feat_imp_sorted = feat_imp.sort_values(ascending=False)
    feat_imp_sorted.to_csv(os.path.join(OUTPUT_DIR, "feature_importance_randomforest.csv"))

    plt.figure(figsize=(10, 5))
    sns.barplot(x=feat_imp_sorted.values[:10], y=feat_imp_sorted.index[:10], orient="h")
    plt.title("Топ-10 признаков по важности (RandomForest)")
    plt.xlabel("Важность")
    plt.ylabel("Признак")
    plt.tight_layout()
    plt.show()

    plt.figure(figsize=(10, max(4, len(feat_imp_sorted)//4)))
    sns.barplot(x=feat_imp_sorted.values, y=feat_imp_sorted.index, orient="h")
    plt.title("Важность всех признаков (RandomForest)")
    plt.xlabel("Важность")
    plt.ylabel("Признак")
    plt.tight_layout()
    plt.show()
    print("[!] Важность признаков визуализирована.")
```

Рисунок 7.5. Топ-10 признаков по важности (RandomForest)

Рисунок 7.6. Важность всех признаков (RandomForest)

Полученные модели и конфигурации признаков сохранялись для последующего использования на этапе генерации персонализированных рекомендаций. Проведённая серия экспериментов продемонстрировала высокую предсказательную способность построенных моделей, а также выявила наиболее значимые характеристики, определяющие цифровой профиль студента.

2.8 Генерация персонализированных рекомендаций (gigachat api)

На заключительном этапе реализован модуль автоматической генерации персонализированных цифровых рекомендаций для студентов на основе их анкетных данных и выделенного цифрового профиля. Алгоритм строится на связке предсказания профиля с помощью обученного классификатора и последующей генерации текстовых советов с использованием API GigaChat, что обеспечивает как автоматизацию аналитики, так и индивидуальный подход в образовательной поддержке.

Процесс начинается с получения access token через OAuth-авторизацию в системе GigaChat. Токен необходим для безопасного обращения к API генерации текста.

```python
import requests
import uuid
from urllib.parse import urlencode
import urllib3

GIGACHAT_AUTH_KEY = "Basic [redacted]=="
GIGACHAT_SCOPE = "GIGACHAT_API_PERS"

print("[0] Получение Access Token от GigaChat...")
oauth_url = "https://ngw.devices.sberbank.ru:9443/api/v2/oauth"
oauth_headers = {
    "Content-Type": "application/x-www-form-urlencoded",
    "Accept": "application/json",
    "RqUID": str(uuid.uuid4()),
    "Authorization": GIGACHAT_AUTH_KEY  # НЕ МЕНЯТЬ!
}
oauth_data = {
    "scope": GIGACHAT_SCOPE
}

urllib3.disable_warnings(urllib3.exceptions.InsecureRequestWarning)

data_encoded = urlencode(oauth_data)
oauth_response = requests.post(
    oauth_url,
    headers=oauth_headers,
    data=data_encoded.encode('utf-8'),
    verify=False
)

if oauth_response.status_code != 200:
    print("Ошибка:", oauth_response.status_code)
    print(oauth_response.text)
    raise Exception(f"Не удалось получить токен: {oauth_response.status_code} - {oauth_response.text}")

GIGACHAT_ACCESS_TOKEN = oauth_response.json()["access_token"]
print("[✓] Access Token получен")
```

Рисунок 8.1. Получение OAuth токена для GigaChat API

Следующий шаг — загрузка обученной модели классификации, масштабировщика, селектора признаков и списка отобранных признаков. Профиль нового студента определяется автоматически по данным анкеты.

```python
MODEL_PATH = "../models/model_RandomForest.pkl"

print("Текущая рабочая директория:", os.getcwd())

# Проверяем наличие модели по относительному пути
if not os.path.exists(MODEL_PATH):
    print(f"Файл модели не найден: {MODEL_PATH}")
    print("Возможные причины:")
    print("- Модель не обучалась или не была сохранена")
    print("- Неправильный путь относительно рабочей директории")
    print("Содержимое ../models/:", os.listdir("../models") if os.path.exists("../models") else "нет папки")
else:
    print("[1] Загрузка модели и признаков...")
    scaler, selector, model, selected_feature_names = joblib.load(MODEL_PATH)
    print(f"[INFO] Признаки, используемые для классификации: {selected_feature_names}")

    import pandas as pd
    NEW_FORM_PATH = "../data/example_student.xlsx"
    df_new = pd.read_excel(NEW_FORM_PATH)
    for col in selected_feature_names:
        if col not in df_new.columns:
            df_new[col] = 0
    df_new = df_new[selected_feature_names]

    X_scaled = scaler.transform(df_new)
    X_selected = selector.transform(X_scaled)
    pred_cluster = model.predict(X_selected)[0]

    profile_names = {
        0: "Цифровые скептики",
        1: "Умеренно вовлечённые",
        2: "Цифровые энтузиасты"
    }
    profile = profile_names.get(pred_cluster, "Неизвестно")
    print("Профиль студента:", profile)
```

Рисунок 8.2. Автоматическое определение цифрового профиля нового студента

Для генерации персональных рекомендаций формируется детализированный prompt, в котором отражаются как тип профиля, так и индивидуальные особенности использования цифровых инструментов. Затем запрос отправляется в модель GigaChat с использованием токена.

```
prompt = f"""
Ты — цифровой наставник для студента. Твоя задача — на основе его профиля и анкеты
сформировать **3 индивидуальные рекомендации по цифровому саморазвитию** в образовательной среде.

Профиль студента: {profile}
Анкета студента (признаки использования цифровых инструментов):
{df_new.iloc[0].to_dict()}

Учитывай:
- слабые стороны профиля (например, низкое использование ИИ или LMS),
- особенности ответов анкеты (например, что студент не использует видеоплатформы или цифровые опросы),
- рекомендации должны быть реалистичны и применимы студенту,
- избегай общих фраз — используй конкретные предложения: какие инструменты, как использовать, зачем это поможет.

Выведи только 3 чёткие рекомендации.
"""

headers = {
    "Content-Type": "application/json",
    "Authorization": f"Bearer {GIGACHAT_ACCESS_TOKEN}"
}
json_data = {
    "model": "GigaChat:latest",
    "messages": [
        {"role": "system", "content": "Ты эксперт в цифровом обучении."},
        {"role": "user", "content": prompt}
    ]
}

response = requests.post(
    "https://gigachat.devices.sberbank.ru/api/v1/chat/completions",
    headers=headers,
    json=json_data,
    timeout=60,
    verify=False
)

if response.status_code == 200:
    recommendations_text = response.json()["choices"][0]["message"]["content"]
    print("\n[✓] Получены рекомендации:\n")
    print(recommendations_text)
else:
    print("[✗] Ошибка при запросе к GigaChat API:", response.status_code)
    print(response.text)
    recommendations_text = ""
```

Рисунок 8.3. Вызов GigaChat API и получение индивидуальных цифровых рекомендаций

Полученные рекомендации сохраняются как в текстовый, так и PDF-формат для последующего использования студентом или преподавателем.

```
OUTPUT_PDF = "../outputs/reports/gigachat_recommendations.pdf"
os.makedirs(os.path.dirname(OUTPUT_PDF), exist_ok=True)

font_path = "DejaVuSans.ttf"
if not os.path.exists(font_path):
    print("Файл DejaVuSans.ttf не найден. PDF не будет создан.")
else:
    from fpdf import FPDF
    pdf = FPDF()
    pdf.add_page()
    pdf.add_font("DejaVu", fname=font_path, uni=True)
    pdf.set_font("DejaVu", size=12)
    pdf.cell(200, 10, txt=f"Профиль: {profile}", ln=True)
    pdf.ln(5)
    for line in recommendations_text.split('\n'):
        pdf.multi_cell(0, 10, txt=line)
    pdf.output(OUTPUT_PDF)
    print(f"Сохранено в {OUTPUT_PDF}")
```

Рисунок 8.4. Сохранение рекомендаций в текстовый и PDF-формат

Данный этап завершает автоматизированный цикл анализа: от первичной анкеты и распознавания профиля до генерации персональных рекомендаций с помощью искусственного интеллекта. Технологическое решение интегрирует

классификацию и генерацию текста, позволяя обеспечить адресную цифровую поддержку каждому студенту на основе объективных данных.

2.9 Анализ распределения цифровых профилей по факультетам и автоматическая интерпретация

Для углублённого анализа цифровых профилей студентов по факультетам реализован автоматизированный модуль, позволяющий не только выявить различия между подразделениями, но и получить интерпретируемые характеристики каждого профиля на основании числовых и бинарных признаков анкеты.

```
import os
import pandas as pd
import matplotlib.pyplot as plt

# Пути
INPUT_PATH = "outputs/cluster_students_umap/cluster_students/students_with_clusters.csv"
OUTPUT_DIR = "outputs/analyze_faculties"
os.makedirs(OUTPUT_DIR, exist_ok=True)

print("[1] Загрузка данных...")
df = pd.read_csv(INPUT_PATH)
print("Данные загружены:", df.shape)
```

✓ 0.0s

```
[1] Загрузка данных...
Данные загружены: (711, 29)
```

Рисунок 9.1. Загрузка итогового датасета с профилями студентов и факультетальной принадлежностью

Анализ распределения профилей между факультетами строился на группировке студентов по признаку факультета и подсчёте процентного соотношения каждого цифрового профиля в пределах отдельного факультета. Визуализация выполнялась в виде stacked barplot, наглядно показывающего пропорции цифровых профилей по всем факультетам.

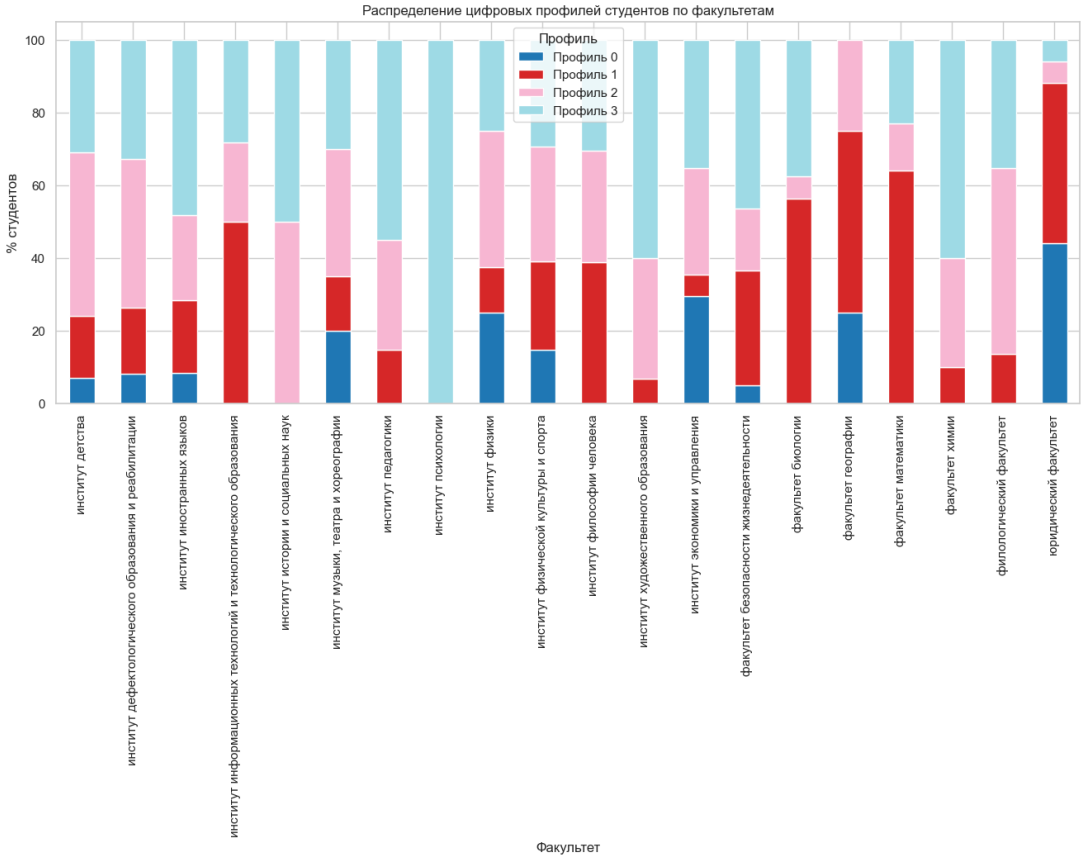


Рисунок 9.2. Распределение цифровых профилей студентов по факультетам (stacked barplot)

Для дальнейшей детализации профилей по признакам формировался список числовых переменных, не включающих технические столбцы и идентификаторы. Это обеспечивает анализ только информативных характеристик студентов.

```
technical_cols = [
    "ID", "Кластер_AC", "Кластер_FCM", "UMAP1", "UMAP2", "Факультет", "Профиль"
]
feature_cols = [
    col for col in df.columns
    if col not in technical_cols and pd.api.types.is_numeric_dtype(df[col])
]

if not feature_cols:
    print("[!] Нет числовых признаков для построения radar plot.")
else:
    print(f"[INFO] Для анализа используются признаки: {feature_cols}")
```

Рисунок 9.3. Выбор информативных числовых признаков для профилирования

По каждому профилю (кластеру) строился radar plot (полярная диаграмма), отражающая средние значения по топ-10 наиболее различающих

признаков. Такой подход позволяет увидеть силуэт и уникальные особенности каждого цифрового профиля.

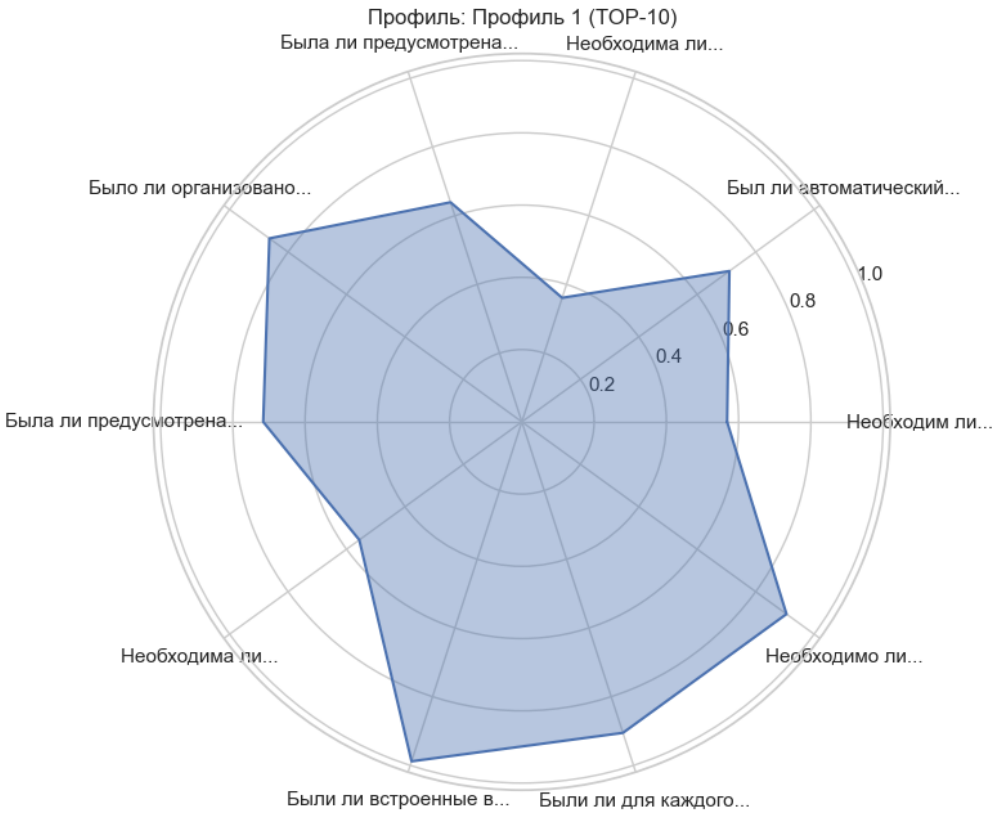


Рисунок 9.4. Полярная (radar) диаграмма цифрового профиля по факультету и признакам

Для автоматизации интерпретации были сгенерированы текстовые описания каждого профиля на основе средних значений признаков. Описания содержат общую численность студентов, топовые признаки профиля, а также указание на характерные сильные и слабые стороны.

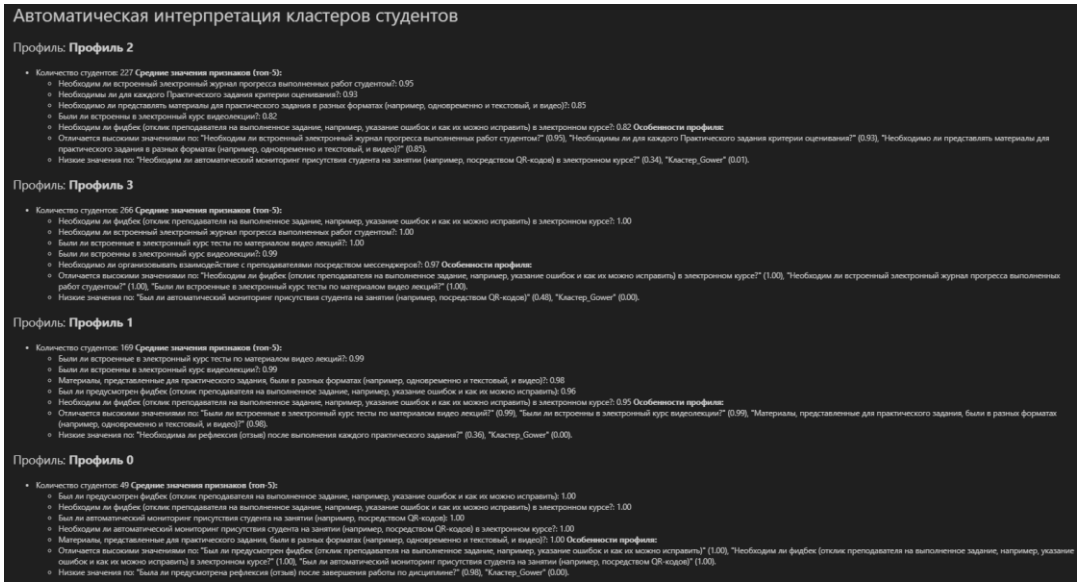


Рисунок 9.5. Автоматическая текстовая интерпретация профилей студентов

Этот этап завершает сквозную аналитику: распределение и характеристика цифровых профилей студентов по факультетам становится полностью прозрачным, автоматическая генерация текстовых интерпретаций позволяет быстро получать содержательные выводы для отчёта, принятия решений или дальнейших образовательных инициатив.

Вывод по главе 2

В ходе практической реализации поставленных задач по анализу эффективности цифровизации учебного процесса средствами машинного обучения был построен полный цикл обработки, моделирования и интерпретации анкетных данных студентов. На этапе разведочного анализа подробно изучена структура, полнота и распределение признаков, выявлены основные закономерности и потенциальные источники дисбаланса данных. Процедуры очистки и преобразования охватили формирование набора бинарных и числовых признаков, обеспечивших оптимальную основу для дальнейшего моделирования.

Кластеризация цифровых профилей студентов осуществлялась с применением современных методов пониженной размерности (UMAP), а также с использованием специализированных метрик для смешанных данных (расстояние Говера), что позволило получить интерпретируемые группы с чёткой дифференциацией по цифровому поведению. Оценка качества кластеризации базировалась на совокупности формальных метрик (Silhouette Score, индекс Дэвиса–Болдуина) и визуальном анализе распределения объектов, что обеспечило объективный контроль валидности выделенных профилей.

В результате профилирования были получены содержательные описания и визуализации, позволяющие выделять ключевые отличия между кластерами как на уровне отдельных признаков, так и на уровне факультетальных распределений. Визуальный и количественный анализ профилей с использованием barplot и полярных диаграмм подтвердил релевантность выделения цифровых типов студентов. Модели классификации профиля, построенные на основе исходных признаков, продемонстрировали высокую точность и устойчивость, а анализ важности переменных позволил выявить наиболее значимые характеристики, определяющие принадлежность к конкретному цифровому профилю.

Финальным этапом стала автоматизация генерации индивидуальных рекомендаций с помощью интеграции обученного классификатора и языковой модели GigaChat. Такой подход обеспечил возможность формировать точечные советы для новых студентов, ориентированные на сильные и слабые стороны их цифрового поведения, с автоматическим сохранением результатов для дальнейшего использования. Проведённый анализ распределения профилей по факультетам и автоматическая интерпретация результатов позволили не только оценить уровень цифровизации в разрезе подразделений, но и предложить рекомендации по дальнейшему развитию цифровых компетенций в образовательной среде.

Комплексная реализация представленных этапов продемонстрировала высокую степень автоматизации анализа и персонализации выводов, что отражает современные подходы к образовательной аналитике и цифровой трансформации высшего образования.

ЗАКЛЮЧЕНИЕ

В ходе выполнения дипломного проекта по использованию методов машинного обучения для анализа эффективности цифровизации учебного процесса были выполнены следующие задачи:

1. Проанализированы современные тенденции цифровизации высшего образования, рассмотрены основные подходы и технологии обработки образовательных данных.
2. Изучены методы машинного обучения, применяемые для анализа анкетных данных студентов, а также реализованы подходы к кластеризации и классификации цифровых профилей.
3. Проведен полный цикл обработки и разведочного анализа исходных данных, включая очистку, кодирование признаков и масштабирование.
4. Построены модели кластеризации, позволившие выделить однородные группы студентов по параметрам цифровой активности.
5. Реализован классификатор профилей, обеспечивающий автоматическое определение цифрового профиля нового студента на основании его анкетных данных.
6. Разработан и внедрён модуль автоматизированной генерации персонализированных рекомендаций с использованием API GigaChat, что позволило повысить индивидуализацию образовательной поддержки студентов.
7. Проведен анализ распределения профилей по факультетам и автоматическая интерпретация результатов, что позволило получить содержательные выводы о специфике цифрового поведения различных групп студентов.

В результате проведённой работы были достигнуты все поставленные задачи и продемонстрирована возможность практического применения машинного обучения для повышения эффективности цифровизации учебного

процесса в высшей школе. Разработанная система анализа и персонализации рекомендаций может быть использована в образовательных организациях для совершенствования учебных стратегий и поддержки студентов на основе объективных данных.

БИБЛИОГРАФИЯ

1. Коробков Н. Что такое цифровизация образования и зачем она нужна [Электронный ресурс] // Блог платформы Skillspace. – 2021. – URL: https://skillspace.ru/blog/chto-takoe-cifrovizaciya-obrazovaniya-i-zachem-ona-nuzhna/ (дата обращения: 20.05.2025).
2. UNESCO. Digital learning and transformation of education [Electronic resource]. – URL: https://www.unesco.org/en/digital-education (accessed: 20.05.2025).
3. Шпунт Я. Минобрнауки внесло программу цифровизации российского высшего образования на рассмотрение в правительство [Электронный ресурс] // ComNews. – 2023. – 12 декабря. – URL: https://www.comnews.ru/content/230626/2023-12-12/minobrnauki-vneslo-programmu-cifrovizacii-rossiyskogo-vysshego-obrazovaniya (дата обращения: 20.05.2025).
4. Научно-исследовательский институт развития образования РЭУ им. Г.В. Плеханова. Основные тренды цифровизации высшего образования: результаты мониторинга тенденций развития высшего образования в мире и в России. – Выпуск 1. – М.: РЭУ им. Г.В. Плеханова, 2021. – 46 с.
5. Итоги eSTARS: высшее электронное [Электронный ресурс] // IQ — портал НИУ ВШЭ (новостная статья). – URL: https://iq.hse.ru/news/423670621.html (дата обращения: 20.05.2025).
6. UNESCO. UNESCO spotlights how digital learning can promote equity in low-resource contexts [Electronic resource]. – 31.03.2025. – URL: https://www.unesco.org/en/articles/unesco-spotlights-how-digital-learning-can-promote-equity-low-resource-contexts (accessed: 20.05.2025).
7. UNESCO. Digital Learning Week 2025 – AI and the future of education: Disruptions, dilemmas and directions (Concept note) [Electronic resource]. –

URL: https://www.unesco.org/en/weeks/digital-learning (accessed: 20.05.2025).

8. OECD. Digital higher education: Emerging quality standards, practices and supports. – OECD Education Working Papers No. 281. – Paris: OECD Publishing, 2022. – DOI: 10.1787/f622f257-en. (доступен онлайн).
9. Флах П. Машинное обучение. — М.: ДМК Пресс, 2015. — 400 с.
10. Mitchell T. Machine Learning. — New York: McGraw-Hill, 1997. — 414 p.
11. Witten I. H., Frank E. Data Mining: Practical Machine Learning Tools and Techniques. — 2nd ed. — Morgan Kaufmann, 2005. — 560 p.
12. Hastie T., Tibshirani R., Friedman J. The Elements of Statistical Learning: Data Mining, Inference, and Prediction. — 2nd ed. — Springer, 2009. — 746 p.
13. Breiman L. Random Forests // Machine Learning. — 2001. — Vol. 45, no. 1. — pp. 5–32.
14. Bezdek J. C. Pattern Recognition with Fuzzy Objective Function Algorithms. — New York: Plenum Press, 1981. — 272 p.
15. Jain A. K., Murty M. N., Flynn P. J. Data clustering: a review // ACM Computing Surveys. — 1999. — Vol. 31, no. 3. — pp. 264–323.
16. Хайкин С. Нейронные сети: полный курс. — 2-е изд. — М.: Вильямс, 2006. — 1104 с.
17. Алпатов А. В. Применение машинного обучения для анализа образовательных результатов студентов вузов // Информационные и математические технологии в науке и управлении. – 2023. – №4(32). – С. 67–78.
18. Малышев В. В., Сливкин С. С., Рукавишников В. С., Базаркин Е. В. Применение методов машинного обучения для построения рекомендательной системы отбора анкет абитуриентов // Научный вестник НГТУ. – 2017. – Т. 67, №2. – С. 109–119.