



СИМУЛЯТОР НАСТРОЙКИ ГИПЕРПАРАМЕТРОВ МОДЕЛЕЙ МАШИННОГО ОБУЧЕНИЯ С ДИНАМИЧЕСКОЙ ВИЗУАЛИЗАЦИЕЙ МЕТРИК КАЧЕСТВА КАК СРЕДСТВО ФОРМИРОВАНИЯ ПРОФЕССИОНАЛЬНЫХ КОМПЕТЕНЦИЙ СТУДЕНТОВ ТЕХНИЧЕСКИХ НАПРАВЛЕНИЙ ПОДГОТОВКИ

Догадина Е. П. ORCID ID 0000-0002-9259-8702

*Федеральное государственное образовательное бюджетное учреждение высшего образования
«Финансовый университет при Правительстве Российской Федерации», Москва,
Российская Федерация;*

*Федеральное государственное бюджетное образовательное учреждение высшего образования
«МИРЭА – Российский технологический университет», Москва, Российская Федерация,
e-mail: epdogadina@fa.ru*

В статье рассматривается анализ трендов цифровизации и интерактивности в освоении технологий машинного обучения студентами технических специальностей направлений «Прикладная информатика» и «Прикладная математика и информатика», а также переход от статических к динамическим визуализациям при решении задач машинного обучения. Традиционные методы обучения, основанные на статическом анализе кода и теоретических выкладках, часто не позволяют наглядно продемонстрировать влияние гиперпараметров на качество моделей и геометрию принимаемых решений. Цель работы – обосновать педагогическую технологию использования интерактивного симулятора настройки гиперпараметров моделей машинного обучения как средства формирования профессиональных компетенций студентов технических направлений подготовки, а также экспериментально проверить ее эффективность в системе профессионального образования на основе квазиэкспериментального исследования. В работе представлена архитектура приложения и графический пользовательский интерфейс реализованных алгоритмов машинного обучения. Ключевой особенностью разработанной системы является возможность динамической визуализации процессов обучения в реальном времени: при изменении пользователем значений гиперпараметров мгновенно перестраиваются графики аппроксимации (для задач регрессии) и карты границ принятия решений (для задач классификации), а также обновляются метрики качества. Актуальностью работы является внедрение разработанного симулятора в процесс обучения студентов для формирования у них профессиональных компетенций в области машинного обучения на основе принципа динамической визуализации, обеспечивающий переход от репродуктивного освоения алгоритмов к осознанному причинно-следственному моделированию их поведения. Результаты исследования показывают, что использование интерактивного симулятора повышает качество формируемых у студентов технических направлений подготовки профессиональных компетенций и способствует сокращению времени на усвоение таких концепций машинного обучения, как визуализация разделяющих поверхностей и кривых регрессии, диагностика переобучения через геометрические артефакты, а также контекстуальная интерпретация метрик качества. Разработанное программное средство рекомендуется к использованию в учебном процессе вузов при изучении дисциплин, связанных с интеллектуальным анализом данных и искусственным интеллектом.

Ключевые слова: вуз, студенты, технические направления подготовки, машинное обучение, симулятор, когнитивная нагрузка, квазиэксперимент, система оценки образовательных ресурсов

HYPERPARAMETER TUNING SIMULATOR FOR MACHINE LEARNING MODELS WITH DYNAMIC VISUALIZATION OF QUALITY METRICS AS A MEANS OF DEVELOPING PROFESSIONAL COMPETENCIES OF STUDENTS IN TECHNICAL FIELDS

Dogadina E. P. ORCID ID 0000-0002-9259-8702

*Federal State Educational Budgetary Institution of Higher Education
“Financial University under the Government of the Russian Federation”,
Moscow, Russian Federation;*

*Federal State Budgetary Educational Institution of Higher Education
“MIREA – Russian Technological University”, Moscow, Russian Federation,
e-mail: epdogadina@fa.ru*

This article examines the trends in digitalization and interactivity in the development of machine learning technologies by students majoring in Applied Informatics and Applied Mathematics and Informatics, as well as the transition from static to dynamic visualizations in solving machine learning problems. Traditional teaching methods, based on static code analysis and theoretical derivations, often do not allow for a clear demonstration of the impact of hyperparameters on model quality and the geometry of decision-making. The aim of the work is to justify a pedagogical technology for using an interactive simulator for tuning machine learning model

hyperparameters as a means of developing professional competencies in students of technical training programs, as well as to experimentally test its effectiveness in the professional education system based on a quasi-experimental study. The work presents the architecture of the application and the graphical user interface of the implemented machine learning algorithms. A key feature of the developed system is the ability to dynamically visualize training processes in real time: when the user changes hyperparameter values, approximation graphs (for regression tasks) and decision boundary maps (for classification tasks) are instantly rebuilt, and quality metrics are also updated. The relevance of this study lies in the implementation of the developed simulator in student learning to develop professional machine learning competencies based on the principle of dynamic visualization, which facilitates the transition from the repetitive acquisition of algorithms to conscious, cause-and-effect modeling of their behavior. The study's results demonstrate that the use of an interactive simulator improves the quality of students' professional competencies in technical training and helps reduce the time required to master machine learning concepts such as visualization of separating surfaces and regression curves, overfitting diagnostics using geometric artifacts, and contextual interpretation of quality metrics. The developed software tool is recommended for use in the educational process of universities when studying disciplines related to data mining and artificial intelligence.

Keywords: university, students, technical fields of study, machine learning, simulator, cognitive load, quasi-experiment, educational resource assessment system

Введение

Современный этап развития информационных технологий характеризуется стремительным ростом объема данных и повсеместным внедрением методов машинного обучения (МО) в различные сферы человеческой деятельности. Алгоритмы регрессии и классификации являются базовыми инструментами принятия решений в большом спектре задач: от прогнозирования финансовых показателей до медицинской диагностики [1, 2]. В связи с этим возрастает потребность в подготовке высококвалифицированных специалистов в области интеллектуального анализа данных и разработки алгоритмов МО. В условиях цифровой трансформации экономики ключевым результатом обучения становится способность выпускника не только применять готовые библиотеки МО и анализа данных, но и осознанно настраивать гиперпараметры моделей, интерпретировать геометрию принимаемых решений и диагностировать качество прогнозов. Разработанный симулятор с динамической визуализацией выступает методическим средством, обеспечивающим переход от формального применения алгоритмов к глубокому пониманию их внутреннего поведения. На текущем этапе развития IT-инфраструктуры методы машинного обучения стали неотъемлемой частью прикладных задач – от финансового прогнозирования до систем поддержки принятия решений [3, 4]. Это требует подготовки специалистов, способных не просто вызывать функции готовых библиотек, а понимать внутренние механизмы алгоритмов. Однако на практике переход от теории к реализации вызывает существенные затруднения. Студентам любых направлений подготовки, чье обучение связано с изучением методов машинного обучения, трудно интуитивно связать математическую модель с ее поведением в коде. Речь идет о таких базовых методах, как k-ближайших сосе-

дей (k-NN), случайный лес, логистическая и линейная регрессия, деревья решений. В учебном процессе при преподавании дисциплины «Машинное обучение» они часто подаются как набор формул, которые нужно запомнить, а не как инструменты, поведение которых можно контролировать.

Система дидактических препятствий, возникающих в процессе усвоения учебного материала и препятствующих формированию устойчивых профессиональных компетенций, заключается в том, что студент зачастую не понимает суть выполненных преобразований, а просто подбирает исходные значения гиперпараметров для получения оптимального результата. В зависимости от используемого метода машинного обучения в качестве гиперпараметров могут быть представлены количество соседей, глубина дерева, коэффициент регуляризации [5]. В настоящее время обучение студентов проходит с использованием сред программирования, в которых процесс настройки гиперпараметров, их визуализация, а также сам процесс обучения остается неохваченным. Это препятствует глубокому усвоению материала и формированию профессиональных компетенций.

Для качественного понимания принципов машинного обучения важно видеть, как меняется качество модели (точность, F1-мера, ошибка аппроксимации) непосредственно во время изменения параметров [6]. Без использования интерактивного инструмента, позволяющего отслеживать эту динамику в реальном времени, материал запоминается формально [7].

На современном рынке образовательных и исследовательских программных продуктов представлен ряд инструментов, поддерживающих работу с методами машинного обучения [8–10]. Их можно условно разделить на три категории: промышленные платформы автоматизированного машинного обучения (AutoML), веб-

ориентированные образовательные визуализаторы и специализированные библиотеки для анализа данных.

Промышленные платформы и среды разработки (Jupyter Lab, Google Colab, Scikit-learn) обладают гибкостью, имеют возможность работы с большими данными и обширную документацию. Однако данные среды требуют навыков программирования. Стремление понятно донести до студентов принципы реализации задач машинного обучения привело к появлению инструментов, изначально ориентированных на визуализацию принципов машинного обучения. Наиболее известный из них – TensorFlow Playground – позволяет в интерактивном режиме наблюдать за обучением нейросети прямо в браузере. Другой подход реализован в Orange Data Mining, где модель строится по принципу визуального программирования через соединение блоков (виджетов). Эти инструменты действительно дают возможность быстро сформировать базовое понимание процессов.

Тем не менее за наглядность приходится платить гибкостью. Анализ их архитектурных решений выявляет ряд системных ограничений. Во-первых, подавляющее большинство подобных сред являются веб-ориентированными, что делает их использование проблематичным в условиях отсутствия устойчивого сетевого соединения. Во-вторых, акцент сделан на нейросетевые архитектуры или базовую классификацию. Ключевые методы, такие как k-NN с варьируемыми метриками расстояния или деревья решений с настройкой критериев разделения, реализованы схематично, а их специфические параметры либо скрыты от пользователя, либо заблокированы на уровне интерфейса.

Чтобы оценить методический потенциал использования разработанного симулятора в учебном процессе, целесообразно сопоставить его с доступными программными средствами, обратив внимание на степень интерактивности, целевое назначение, набор поддерживаемых алгоритмов и возможности внедрения в учебный процесс.

В области подбора гиперпараметров преобладают решения, созданные под нужды AutoML-исследований [11]. Показательным здесь выступает проект DeepCAVE, предназначенный для ретроспективного изучения результатов оптимизации. Система корректно обрабатывает выводы таких оптимизаторов, как SMAC или Optuna, и строит детальные графики траекторий поиска, чувствительности параметров и распределения вычислительного бюджета. При этом архитектура платформы заточена под анализ завершенных запусков, что подразумевает

уверенное владение студента форматами логов и понимание процедур оптимизации гиперпараметров. Возможность «живого» экспериментирования с параметрами в ней не предусмотрена, а высокий уровень знаний делает DeepCAVE неудобным для студентов, которым на начальном этапе важнее не получить оптимальную конфигурацию, а проследить причинно-следственную связь между настройкой алгоритма и изменением его прогнозирующей поверхности.

Библиотека Yellowbrick работает несколько иначе, чем стандартные инструменты, расширяя scikit-learn визуальными средствами для диагностики [12]. В ней находятся множество визуализаторов: от оценки качества разбиения данных до анализа важности признаков и остатков. Однако статичные отчеты не дают нужной гибкости. Чтобы посмотреть новый график, нужно каждый раз писать вызов в коде, потом перезапускать ядро и только после этого вникать в суть. Студент вместо наблюдения за тем, как меняется граница при сдвиге гиперпараметра, вынужден отвлекаться на отладку скриптов. Аналогичные проблемы есть и у научной литературы по данной тематике, а также у официальной документации scikit-learn, обладающей хорошей теоретической базой. Однако для начального обучения этого недостаточно, и подбор параметров превращается в механический перебор, а визуальная сторона работы алгоритма так и остается скрытой за числовыми отчетами.

Что касается систем TensorBoard, то они ориентированы на нейросети и мониторинг их обучения [13]. Они эффективно визуализируют распределение градиентов и архитектуру вычислительных графов, но жестко привязаны к фреймворку TensorFlow и не работают с табличными моделями машинного обучения. В качестве альтернативы для новичков часто упоминают TensorFlow Playground – браузерный тренажер, позволяющий в реальном времени менять структуру сети и наблюдать за реакцией модели. Действительно, мгновенная обратная связь подтверждает дидактическую ценность интерактивности. Вместе с тем узкий фокус на многослойных перцептронах, обязательное наличие интернета и отсутствие поддержки классических алгоритмов (деревьев, ансамблей, метрических методов) сужают область применения Playground. Для полноценного курса по машинному обучению, где требуется последовательное освоение базовых методов, этого недостаточно.

Для решения педагогической задачи формирования интуитивного понимания влияния гиперпараметров на поведение моделей машинного обучения была разработа-

на программа на Python (фреймворк PyQt5), которая позволяет визуально сопоставлять настройку гиперпараметров с результатами предсказания в режиме реального времени. Это дает возможность проводить эксперименты в режиме реального времени и наблюдать за геометрией моделирования наглядно.

Цель исследования – научно-методическое обоснование педагогической технологии использования интерактивного симулятора настройки гиперпараметров моделей машинного обучения как средства формирования профессиональных компетенций студентов технических направлений подготовки, а также экспериментальная проверка ее эффективности в системе профессионального образования на основе квазиэкспериментального исследования.

Материалы и методы исследования

Педагогические исследования последних лет последовательно указывают на преимущества сред, построенных по принципу прямой манипуляции объектами. Когда студент видит немедленную реакцию системы на изменение входных условий, формируется более устойчивая ментальная модель изучаемого явления. При этом отмечается системная проблема: большинство доступных решений либо требуют уверенного владения программированием, либо концентрируются исключительно на глубоком обучении (подмножестве машинного обучения), игнорируя фундаментальные алгоритмы, составляющие основу академических программ.

В работе [14] по цифровой дидактике авторы подчеркивают, что на начальном этапе важно снижать когнитивную нагрузку на студентов. Интерфейс должен брать на себя рутину, освобождая место для анализа того, как ведет себя модель. Поэтому разрыв между профессиональными аналитическими платформами и примитивными веб-демонстрациями остается довольно заметным. Симулятор, о котором идет речь в статье, как раз закрывает эту нишу. Акцент здесь не на инженерной автоматизации, а на наглядности обучения. В отличие от систем типа DeepCAVE или Yellowbrick, представленный в работе симулятор действует в реальном времени и не требует от студентов навыков программирования. А от браузерных приложений для нейросетей его выгодно отличает автономность, работа с локальными датасетами и поддержка классических алгоритмов обучения с учителем.

Поэтому разработка универсальной настольной программы с широким набором алгоритмов и удобным графическим интерфейсом становится вполне актуальной задачей.

Главная цель такого приложения будет заключаться в мгновенной обратной связи. Если студент меняет значение параметра, система должна обновлять графики и метрики качества без задержек. Только такой формат позволит по-настоящему увидеть, как меняются разделяющие границы и кривые машинного обучения, превращая абстрактные цифры в наглядную модель поведения алгоритма.

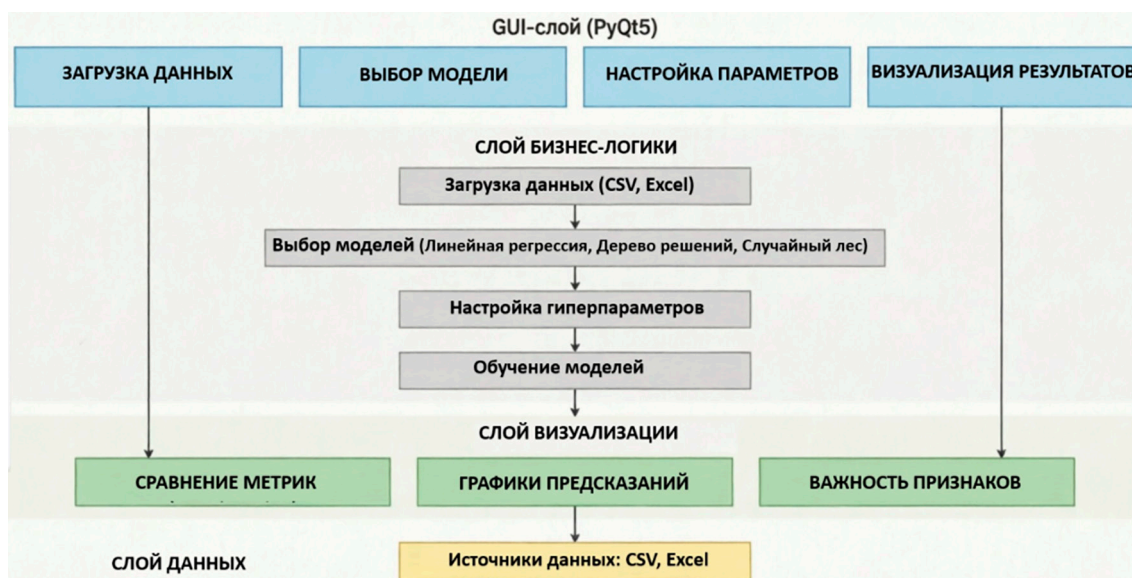


Рис. 1. Структурная схема работы приложения
Примечание: составлен автором на основе анализа предметной области

Разработанное приложение построено по модульной архитектуре с четким разделением ответственности между компонентами, что соответствует принципам MVC (Model View Controller) и обеспечивает масштабируемость кода. Общая структура системы представлена на рис. 1.

Такой интерфейс обеспечивает баланс между функциональностью и простотой, позволяя студентам сосредоточиться на анализе результатов, а не на отладке кода.

Результаты исследования и их обсуждение

В данной работе представлен симулятор настройки гиперпараметров моделей машинного обучения с динамической визуализацией метрик качества, написанный на Python с использованием фреймворка PyQt5 и библиотеки scikit-learn [15, 16].

Это приложение реализует:

- Выбор алгоритма: линейная регрессия, дерево решений, k-ближайших соседей (k-NN), случайный лес, логистическая регрессия.

- Генерацию синтетических данных (для регрессии и классификации), а также загрузку готового датасета.

- Динамическую настройку гиперпараметров через ползунки.

- Мгновенную перерисовку графика предсказаний и отображение метрик качества.

Центральным элементом симулятора является модуль обучения моделей, реализующий ключевые [17, 18] алгоритмы контролируемого обучения.

Традиционная оценка качества модели через числовые метрики (ассурасу, f1-score и т. д.) скрывает характер ошибок. Две модели с одинаковой точностью могут вести себя принципиально по-разному: одна может проводить гладкую разделяющую границу,

другая способна огибать каждый шумовой выброс. Для учебного процесса эта разница принципиальна. Поэтому в основу визуального ядра разрабатываемого инструмента положен анализ решающих поверхностей. Формируется композитное изображение, позволяющее одновременно оценить и структуру данных, и логику работы алгоритма.

Информативность такого представления выходит за рамки простой «картинки». Студент получает возможность напрямую наблюдать, как изменение гиперпараметров трансформирует геометрию решающего правила (рис. 2). Увеличение глубины дерева решений приводит к тому, что гладкая граница начинает дробиться, обрастая изолированными «карманами» – классический признак переобучения. Связь между действием (изменением параметра) и результатом (изменением поверхности) становится прозрачной, что принципиально отличает такой формат от работы со статичными таблицами метрик.

Для задач регрессии описанный выше подход требует модификации, поскольку здесь объектом анализа выступает не разделяющая поверхность, а форма восстанавливаемой зависимости. Прогноз вычисляется вдоль одномерной сетки (по выбранной пользователем оси признакового пространства), после чего строится непрерывная кривая регрессии, на которую накладываются реальные наблюдения (рис. 3). Подобная визуализация дает возможность непосредственно проанализировать масштаб и характер распределения остаточных ошибок. Кроме этого, можно проследить систематические сдвиги, скрытые нелинейные зависимости или, наоборот, избыточную «волнистость» аппроксимирующей кривой, что обычно свидетельствует о переобучении на случайные колебания данных.

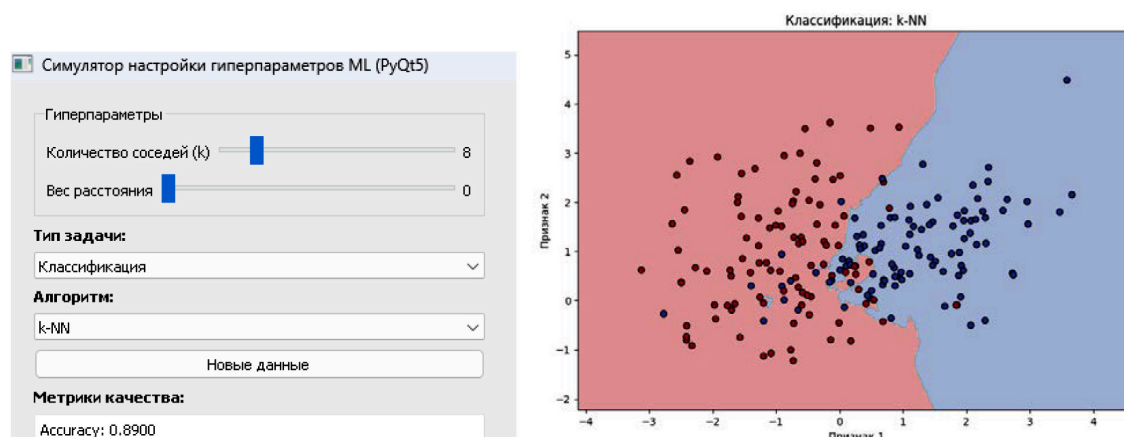


Рис. 2. Симулятор настройки гиперпараметров и отображения данных для задачи классификации
Примечание: составлен автором по результатам данного исследования

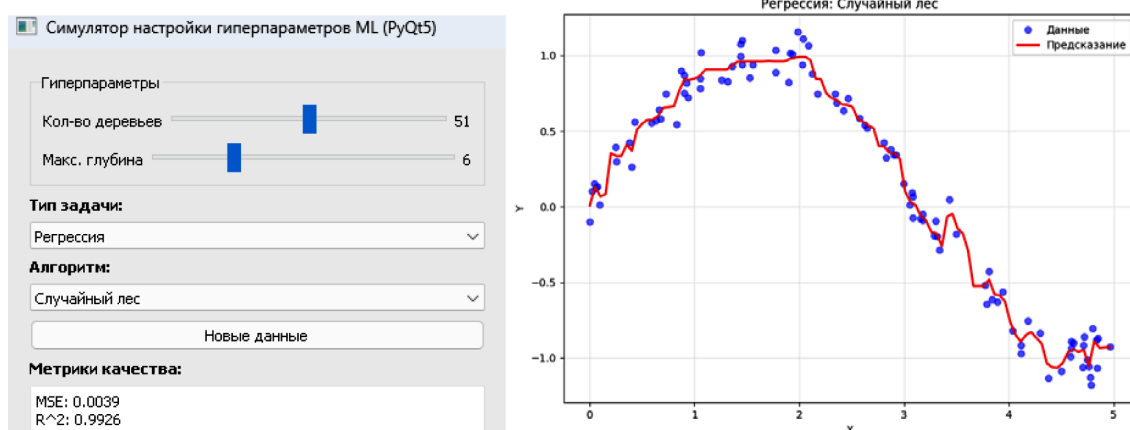


Рис. 3. Симулятор настройки гиперпараметров и отображения данных для задачи регрессии
Примечание: составлен автором по результатам данного исследования

Интеграция указанных режимов в одном программном окне формирует рабочую среду, где учебный эксперимент приобретает конкретный, прикладной характер. В таких условиях студент получает возможность поэтапно верифицировать рабочие гипотезы: сопоставлять устойчивость алгоритмов k-ближайших соседей и случайного леса к аномальным наблюдениям, фиксировать точку баланса между недообучением и переобучением при наращивании сложности модели, а также отслеживать влияние выбранной метрики близости на конфигурацию разделяющих границ в задачах классификации. Поддержание непрерывности исследовательского цикла требует минимальных задержек при обработке запросов системой. Для замеров производительности использовались следующие характеристики тестовой системы: процессор Intel Core i5-1135G7 @ 2.40 GHz, оперативная память 16 ГБ DDR4, операционная система Windows 11, версия Python 3.10.6. Проведенные замеры показали, что на выборках объемом до 1000 объектов полный цикл, включающий в себя пересчет модели, построение сетки, интерполяцию и перерисовку графики, укладывается в 150 мс. Практически это означает, что задержка становится незаметной для пользователя, и настройка гиперпараметров воспринимается как работа в режиме реального времени, без пауз, разрушающих исследовательский поток.

Для проверки дидактической эффективности предложенного программного решения в учебный процесс был внедрен квазиэкспериментальный дизайн, предполагающий сравнение трех форматов проведения лабораторного занятия. Выбор

именно квазиэксперимента обусловлен организационными ограничениями образовательной среды: полная рандомизация участников без нарушения академического расписания невозможна, однако строгий контроль внешних переменных и стандартизация процедур позволяют обеспечить внутреннюю валидность исследования [19]. Экспериментальная часть работы была интегрирована в модуль дисциплины «Машинное обучение», читаемой для студентов 3-го курса направлений технической и информационной подготовки. Общая численность выборки составила 92 чел. Темой занятия была настройка гиперпараметров базовых алгоритмов контролируемого обучения, что соответствует учебному плану и требует от обучающихся перехода от формального применения библиотечных функций к осознанному выбору конфигураций модели.

Формирование учебных групп осуществлялось по принципу стратифицированного распределения. В качестве страт выступали академическая успеваемость за предыдущий семестр (средний балл) и результаты стартового диагностического тестирования, охватывающего базовые понятия теории вероятностей, линейной алгебры и алгоритмизации. Такой подход позволил выровнять группы по уровню исходной подготовки до начала эксперимента.

В зависимости от применяемого инструментария участники были разделены на три потока. Первая группа (n = 31, где n – объем подвыборки) работала в рамках традиционного формата. Обучение строилось на основе лекционного материала и стандартных интерактивных веб-сред, где визуализация результатов носила статиче-

ский характер. Графики и метрики формировались после завершения обучения модели и требовали ручного запуска ячеек кода. Вторая группа ($n = 30$) использовала общедоступные интерактивные платформы. Все участники использовали веб-платформу TensorFlow Playground для демонстрации механизмов обучения нейросетей, а также пакет Yellowbrick для графической оценки моделей, построенных в scikit-learn. Подобный подход дает наглядную картину процесса, однако сохраняет зависимость от навыков программирования и не обеспечивает полноценного изучения классических методов машинного обучения. Третья группа ($n = 31$) работала непосредственно с авторским симулятором.

Чтобы уменьшить влияние сторонних факторов, все группы выполняли строго идентичный комплекс практических заданий. Аппаратное обеспечение компьютерных классов приведено к единому стандарту, благодаря чему различия в вычислительной мощности не могли исказить результаты эксперимента.

Диагностика образовательных достижений реализована через многоуровневую схему, затрагивающую когнитивный, поведенческий и эмоционально-ценностный компоненты учебной деятельности. Изменения в когнитивной сфере отслеживались посредством трехэтапного тестирования. Начальный срез фиксировал исходный уровень владения терминологией, заключительный контроль, проведенный непосредственно по завершении занятия, включал задачи на анализ графиков и аргументацию выбора гиперпараметров, а повторное тестирование через неделю позволяло оценить долговременную сохранность усвоенных концепций [14].

На рис. 4 представлено сравнение групп по ключевым метрикам.

Процессуальные метрики включали хронометраж выполнения лабораторной работы и анализ журналов взаимодействия с программным интерфейсом, в которых фиксировались последовательность изменения параметров, частота возвратов к предыдущим значениям и время принятия решений.

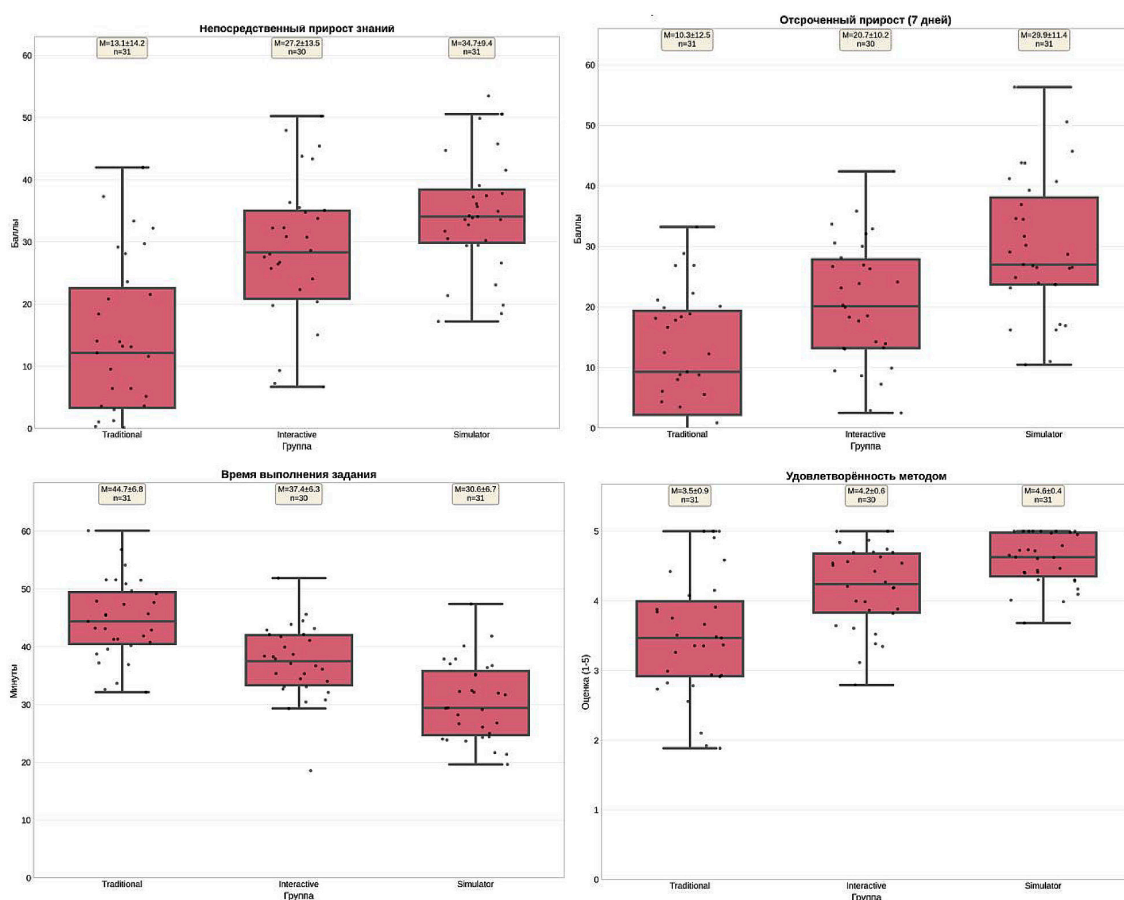


Рис. 4. Сравнение групп по ключевым метрикам

Примечание: составлен автором по результатам данного исследования

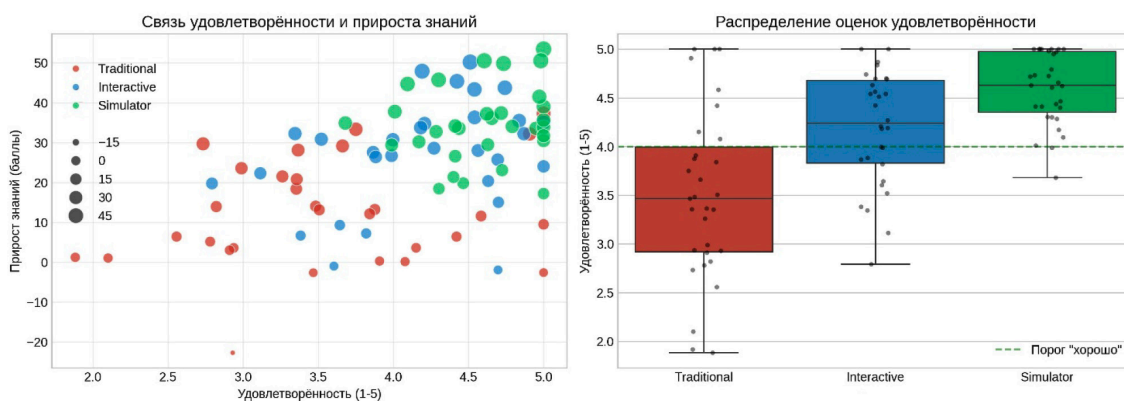


Рис. 5. Сравнение групп по ключевым метрикам
Примечание: составлен автором по результатам данного исследования

Субъективная обратная связь собиралась посредством анкетирования с использованием 5-балльной шкалы Лайкерта. Данное анкетирование представляет собой порядковую шкалу, где респондент выражает степень согласия с утверждением от «полностью не согласен» до «полностью согласен». Вопросы касались наглядности материала, понятности интерфейса и уверенности в интерпретации результатов [11].

На рис. 5 представлена связь удовлетворенности и прироста знаний, а также распределение оценок удовлетворенности по группам.

Обработка полученных массивов данных включала в себя несколько этапов. На первом этапе проверялось соответствие распределений показателей закону нормальности с помощью критерия Шапиро – Уилка. При выполнении условий параметрической статистики прирост учебных достижений оценивался посредством однофакторного дисперсионного анализа (ANOVA). Для локализации значимых межгрупповых различий использовался пост-хок тест через критерий Тьюки, минимизирующий вероятность ошибок первого рода при многократном сравнении. В случаях нарушения предположения о нормальности распределения данных применялся непараметрический критерий Краскела – Уоллиса. Кроме установления статистической значимости, в методику анализа был включен расчет величин эффекта: частичного η^2 для оценки доли дисперсии, приходящейся на фактор принадлежности к группе, и стандартизированной разности средних для попарных сопоставлений [14, 15]. Такой подход обеспечил комплексную интерпретацию результатов, позволив отделить статистически значимые сдвиги от педагогически релевантных.

Заключение

Результаты исследования подтвердили целесообразность внедрения интерактивного симулятора в учебный процесс как средства формирования компетенций в сфере машинного обучения. Анализ существующих программных сред выявил их ограниченную применимость в образовательном контексте, поскольку профессиональные платформы требуют от обучающихся продвинутой технической подготовки, а узкоспециализированные визуализаторы, как правило, не демонстрируют причинно-следственных связей между изменением гиперпараметров и поведением алгоритма. Указанный недостаток обусловил разработку приложения, ориентированного на наглядное моделирование процессов настройки моделей.

Программная реализация выполнена на языке Python с применением фреймворка PyQt5. Архитектура приложения поддерживает работу с широким спектром алгоритмов от базовых линейных методов до ансамблевых моделей. Интерфейс обеспечивает визуальный контроль над ключевыми явлениями, такими как переобучение при увеличении глубины решающего дерева или изменение геометрии границ разделения в алгоритме k-ближайших соседей. В перспективе развития проекта предусмотрена интеграция методов неконтролируемого обучения (кластеризация, снижение размерности), а также модуль интеллектуальной поддержки, генерирующий рекомендации на основе анализа траектории действий обучающегося.

Список литературы

1. Седых И. Ю., Хрипунова М. Б. Технологии искусственного интеллекта в современном высшем образовании России // Цифровые решения и технологии искусственного интеллекта. 2025. Т. 1. № 1. С. 20–27. URL: <https://>

www.digitarin.ru/jour/article/view/4?ysclid=mmwg3ua1dn671013592 (дата обращения: 12.03.2026).

2. Сальников Е. А., Мунинова С. Р. Эволюция искусственного интеллекта: от современных технологий к будущим инновациям. Цифровые решения и технологии искусственного интеллекта. 2025. Т. 1. № 2. С. 64–71. URL: <https://www.digitarin.ru/jour/article/view/14?ysclid=mmwg567s7t715657797> (дата обращения: 12.03.2026).
3. Догадина Е. П., Суханова У. Ю., Ищенко М. А., Веселов Д. И. Распознавание документов в различных условиях с помощью методов машинного обучения // Нелинейный мир. 2025. Т. 23. № 3. С. 45–53. URL: <https://www.elibrary.ru/item.asp?id=82864377> (дата обращения: 11.03.2026). DOI: 10.18127/j20700970-202503-06. EDN: CNSOCL.
4. Догадина Е. П., Бочаров М. И. К вопросу об эффективной проверке качества знаний обучающихся в соответствии с их личностными особенностями // Стандарты и мониторинг в образовании. 2023. Т. 11. № 3. С. 9–13. URL: <https://www.elibrary.ru/item.asp?id=53958906> (дата обращения: 11.03.2026). DOI: 10.12737/1998-1740-2023-11-3-9-13. EDN: IEWREX.
5. Красовская Л. В., Жуков Н. Р. Оценка различий между метриками качества классификаций при различных условиях формирования исходных данных // Фундаментальные и прикладные проблемы математики и информатики в современной науке: теория и практика актуальных исследований: материалы III Всероссийской научно-технической конференции (Махачкала, 02–03 апреля 2025 г.). Махачкала: Дагестанский государственный технический университет, 2025. С. 235–242. [Электронный ресурс]. URL: <https://www.elibrary.ru/item.asp?id=89089519> (дата обращения: 03.03.2026). EDN: RHYUNB.
6. Туманян П. И., Сарджишвили С. Э. Архитектура геймифицированной программной платформы для подготовки данных для обучения AI моделей // Современная наука: актуальные проблемы теории и практики. Серия: Естественные и технические науки. 2023. № 8–2. С. 132–136. URL: <https://www.elibrary.ru/item.asp?id=54769147> (дата обращения: 16.03.2026). DOI: 10.37882/2223-2982.2023.8-2.33. EDN: FMOQTA.
7. Панжиева Н. Н. Использование онлайн-платформы в дидактической модели обучения, стимулирующей навыки критического мышления будущих учителей информатики // Информатика и образование. 2024. Т. 39. № 2. С. 69–77. URL: <https://www.elibrary.ru/item.asp?id=67878735> (дата обращения: 03.03.2026). DOI: 10.32517/0234-0453-2024-39-2-69-77. EDN: SWNXDH.
8. Трунов А. А., Эггарт В. А. Разработка образовательной платформы для обучения программированию в школе // Информационные технологии в образовании. 2020. № 3. С. 245–248. URL: <https://www.elibrary.ru/item.asp?id=44637432> (дата обращения: 16.03.2026). EDN: ZUODFY.
9. Петров А. С., Огнева М. В. Разработка интерактивной платформы для обучения информатике, основанной на игровых элементах // Информационные технологии в образовании. 2023. № 6. С. 267–270. URL: <https://www.elibrary.ru/item.asp?id=58734366> (дата обращения: 16.03.2026). EDN: WPPQMS.
10. Бабкина Е. А. Автоматическое машинное обучение (Automl): алгоритмы и инструменты для снижения порога входа // Международный журнал гуманитарных и естественных наук. 2024. № 6–1 (93). С. 175–178. URL: <https://www.elibrary.ru/item.asp?id=68437737> (дата обращения: 03.03.2026). DOI: 10.24412/2500-1000-2024-6-1-175-178. EDN: AJZVAQ.
11. Понамарев В. В., Китов В. В. Автоматический подбор гиперпараметров с помощью модифицированного алгоритма TPE // Информационные технологии и математические методы в экономике и управлении (ИТиММ-2024): сборник статей XIII Международной научно-практической конференции имени А. И. Китова. В 3 т. (г. Москва, 14–15 марта 2024 г.). Т. 1. М.: Российский экономический университет им. Г. В. Плеханова, 2024. С. 260–267. [Электронный ресурс]. URL: <https://www.elibrary.ru/item.asp?id=73237371> (дата обращения: 03.03.2026). EDN: CLFLSP.
12. Bengfort V., Bilbro R. Yellowbrick: Visualizing the Scikit-Learn Model Selection Process. Journal of Open Source Software. 2019. Vol. 4. Is. 35. P. 1075. URL: <https://joss.theoj.org/papers/10.21105/joss.01075> (дата обращения: 03.03.2026). DOI: 10.21105/joss.01075.
13. Нажимова Н. А., Наумова Е. Г. Формирование навыков построения нейронных сетей в различных программных средах // Современные проблемы науки и образования. 2024. № 3. С. 98. URL: <https://science-education.ru/ru/article/view?id=33522> (дата обращения: 03.05.2026). DOI: 10.17513/spno.33522. EDN: HLAAWU.
14. Новиков А. В., Слабкая Д. Н. Подходы к применению критериев дидактической эффективности в образовательном процессе // Педагогический журнал. 2020. Т. 10. № 5–1. С. 248–254. URL: <https://www.elibrary.ru/item.asp?id=44727766> (дата обращения: 15.03.2026). DOI: 10.34670/AR.2020.64.53.024. EDN: TYDMGL.
15. Нурмагомедов Т. Н., Волкова М. М., Илькевич Б. В. Оценка эффективности дидактической модели преподавания дисциплины «Начертательная геометрия и инженерная графика» с элементами активизации творческого мышления // Международный научно-исследовательский журнал. 2023. № 10 (136). URL: <https://www.elibrary.ru/item.asp?id=54725365> (дата обращения: 15.03.2026). DOI: 10.23670/IRJ.2023.136.23. EDN: NBLEVT.
16. Петров С. В. Влияние настройки параметров метода «случайный лес» на качество регрессий // Информатика и вычислительная техника: сборник научных трудов XV Всероссийской научно-технической конференции аспирантов, студентов и молодых ученых (г. Ульяновск, 14–15 июня 2023 г.). Ульяновск: Ульяновский государственный технический университет, 2023. С. 251–255. [Электронный ресурс]. URL: <https://www.elibrary.ru/item.asp?id=75059272> (дата обращения: 15.03.2026). EDN: EWSZWT.
17. Le L., Xie Y., Raghavan V. V. KNN Loss and Deep KNN // Fundamenta Informaticae. 2021. Vol. 182. Is. 2. P. 95–110. URL: <https://www.elibrary.ru/item.asp?id=69932274> (дата обращения: 15.03.2026). DOI: 10.3233/fi-2021-2068. EDN: RISSXO.
18. Ашуров Х. Е. Линейная регрессия – один из базовых алгоритмов машинного обучения // Наука и технология XXI века. 2025. № 2 (16). С. 52–62. URL: <https://www.elibrary.ru/item.asp?id=82740490> (дата обращения: 15.03.2026). EDN: IRNFVD.
19. Каримов Р. М. Оценка эффективности организационно-педагогических условий формирования поликультурной компетентности: квазиэксперимент в творческом высшем учебном заведении // Современное профессиональное образование. 2025. № 11. С. 29–32. URL: <https://www.elibrary.ru/item.asp?id=87555252> (дата обращения: 15.03.2026). EDN: NINMGU.

Конфликт интересов: Автор заявляет об отсутствии конфликта интересов.

Conflict of interest: The author declares that there is no conflict of interest.

Финансирование: Автор заявляет об отсутствии внешнего финансирования.

Financing: The research was performed without external funding.