

ОРГАНИЗАЦИЯ МУЛЬТИПЛАТФОРМЕННОГО ГИБРИДНОГО ОНЛАЙН-КУРСА ПО «МАШИННОМУ ОБУЧЕНИЮ» ДЛЯ СИСТЕМЫ ВЫСШЕГО ОБРАЗОВАНИЯ

А.А. Никандров, К.Р. Пиотровская

Аннотация. В статье рассмотрен опыт разработки и интеграции в образовательный процесс вуза авторского мультимедийного онлайн-курса «Машинное обучение». На основе мирового опыта выработаны рекомендации по интеграции контента онлайн-курса в образовательную среду вуза. Обсуждены требования к выбору образовательной платформы при обучении программированию. Отмечены ключевые моменты, касающиеся отбора содержания обучения, оценки его качества, а также внедрения в учебный процесс на примере дисциплины «Машинное обучение», разработанной на факультете математики в Российском государственном педагогическом университете им. А.И. Герцена для бакалавров, обучающихся по направлению 01.03.02 «Прикладная математика и информатика». Продемонстрировано инструментальное применение авторской композиции библиотек языка программирования Python 3 для генерации практических заданий по решению задач различной сложности и организации мини-исследований в области машинного обучения, в качестве средств организации обучения указаны платформы Jupiter Notebook, Stepik, Moodle Herzen. При отборе и создании учебного материала активно использованы отечественные ресурсы олимпиадного программирования: Kaggle и Codeforces. Обсуждены первые результаты экспериментальной проверки методики на базе факультета математики РГПУ им. А.И. Герцена.

Ключевые слова: онлайн-курс, гибридное обучение, мультимедийный курс, машинное обучение, интерактивная вычислительная платформа Jupiter Notebook, педагогический эксперимент.

Для цитирования: Никандров А.А., Пиотровская К.Р. Организация мультимедийного гибридного онлайн-курса по «Машинному обучению» для систем высшего образования // Преподаватель XXI век. 2023. № 3. Часть 1. С. 110–127. DOI: 10.31862/2073-9613-2023-3-110-127

© Никандров А.А., Пиотровская К.Р., 2023



Контент доступен по лицензии Creative Commons Attribution 4.0 International License
The content is licensed under a Creative Commons Attribution 4.0 International License

ORGANIZATION OF A HYBRID MULTI-PLATFORM ONLINE COURSE ON "MACHINE LEARNING" FOR THE SYSTEM OF HIGHER EDUCATION

A.A. Nikandrov, X.R. Piotrowska

Abstract. *The article deals with the experience of developing and integrating the author's multiplatform online course "Machine Learning" into the educational process of higher education institutions. On the basis of international experience the recommendations on the integration of the online course content into the educational environment of the university are developed. The requirements to the choice of an educational platform when teaching programming are discussed. The key points concerning the selection of training content, its quality assessment, as well as its implementation in the educational process are highlighted on the example of the discipline "Machine Learning", developed at the Department of Mathematics at A.I. Herzen Russian State Pedagogical University for bachelors studying in the direction 01.03.02 "Applied Mathematics and Informatics". The author's composition of Python 3 programming language libraries for generating practical tasks for solving problems of different complexity and organising mini-research in the field of machine learning is presented. Jupiter Notebook, Stepik, Moodle Herzen platforms are specified as the means of training organisation. When selecting and creating task material, domestic resources of olympiad programming: Kaggle and Codeforces were actively used. The first results of the experimental testing of the methodology at the Faculty of Mathematics of the Herzen Russian State Pedagogical University are discussed.*

Keywords: *online course, hybrid learning, multiplatform course, machine learning, Jupiter Notebook interactive computing platform, pedagogical experiment.*

Cite as: Nikandrov A.A., Piotrowska X.R. Organization of a Hybrid Multi-Platform Online Course on "Machine Learning" for Higher Education System. *Prepodavatel XXI vek*. Russian Journal of Education, 2023, No. 3, part 1, pp. 110–127. DOI: 10.31862/2073-9613-2023-3-110-127

Проблема исследования

Развитие информационно-коммуникационных технологий (ИКТ), как и сама среда Интернет, играют важную роль в современных образовательных изменениях, поддерживая более гибкие, более распространяемые, более открытые формы обучения. Электронное или онлайн-обучение может создать новые условия учебной работы, отличающиеся от традиционного благодаря использования элементов виртуальной реальности. Как известно, система электронного обучения должна состоять из педагогического, технологического и организационного компонентов, что ставит перед разработ-

чиком электронного курса задачу сбалансированного включения этих трех составляющих [1].

Современное образование настойчиво предлагает активно использовать преимущества онлайн-обучения и предоставлять контент, включающий короткие видеолекции, материалы в цифровом виде, интерактивные задания, дискуссионные форумы и тесты, развивающие компетентность в достаточно узких направления [2]. Помимо построения автономных курсов, университеты все чаще интегрируют контент онлайн-курсов в очное обучение, создавая программы смешанного типа [3–5]. При этом студенчество зачастую проявляет высокую удовлетворенность

предоставляемыми онлайн-материалами в качестве дополнения к традиционному очному обучению [6]. Использование онлайн-материалов позволяет преподавателю реформировать и обновлять регулярные очные программы, однако эффективная интеграция таких ресурсов остается непростым и противоречивым процессом.

Особенно ярко это явление проявляется при обучении в областях, связанных с программированием и искусственным интеллектом, тем самым формируя новую образовательную традицию, которая отличается как от принятой традиции официального очного формата, так и от дистанционных форм. Эта традиция отличается [7]:

- сосредоточением на теоретических и прикладных аспектах практического применения получаемых знаний;
- организацией активной социализированной онлайн-среды для обсуждения возникающих вопросов на форуме как с ментором, так и между обучающимися;
- активным использованием аудио- и видеоматериалов, архивированием и доступностью всех лекций и материалов занятий;
- возможностями организации адаптивного персонифицированного обучения;
- оперативной автоматической проверкой программных кодов, генерируемых обучающимися.

Хотя чаще всего онлайн-курсы создаются для решения локальной четко очерченной задачи получения/повышения квалификации в достаточно узкой области, в то же время свободное онлайн-образование постепенно все чаще включается в официальные академические программы колледжей и университетов.

Как уже было отмечено выше, онлайн-курсы в области компьютерных наук зачастую строятся на специализированных платформах с расширенным функцио-

налом проверки выполнения заданий на программирование. Одно из преимуществ этого расширения — применение автоматизированного подхода к проверке программного кода взамен ручного, т. е. осуществляется заблаговременное написание тестов, на которых происходит проверка решения задания, предоставляемого студентом.

Однако практика показывает, что функционал систем, официально используемых для управления учебным процессом вузов, например, LMS Moodle, ISpring, Mirapolis, обычно не вполне отвечает требованиям, предъявляемым профессиональным современным сообществом к онлайн-обучению в области компьютерных наук и информационных технологий. Поэтому предлагается применять гибридный подход в области организации онлайн-поддержек [5]. В контексте реализации онлайн-курсов по программированию термин *гибридность* [8] понимается нами как использование нескольких платформ обучения, т. е. привлечение мультиплатформенности с акцентом на автоматизацию учебных процессов, связанных с проверкой решений практических заданий. В настоящей статье на примере дисциплины «Машинное обучение» демонстрируется построение мультиплатформенного онлайн-курса с целью его интеграции в академический, обсуждены первые результаты экспериментальной проверки.

Для того чтобы определить основные элементы методики обучения в онлайн-курсе, необходимо выбрать платформу, являющуюся хостингом разрабатываемого онлайн-курса и оценить ее потенциал на наличие желаемых режимов обучения. Убедиться в согласованности целей онлайн-курса и основного академического учебного плана, а также учесть социально-эпистемологические аспекты онлайн-курса. Если выявлена потребность в при-

менении мультиплатформенного подхода, то осуществить подбор платформ и решить, каким образом предполагается использовать уже созданные онлайн-материалы, или же есть необходимость создать собственные, новые материалы. Проводя интеграцию онлайн-курса в систему очного обучения, фиксировать детали ее успешности на всех этапах реализации.

При изучении платформ и просмотре готовых онлайн-материалов полезно различать три режима: собственно обучения, социального взаимодействия и формирования оценки обучения. Эти возможности предлагаются практически всеми платформами, но по-разному. Для получения инструкций по работе следует обратиться к таким материалам, как короткие видеоролики, цифровые тексты, иллюстрации, презентации Power Point и ссылки на внешние веб-сайты разработчика платформы [9]. Платформы обычно имеют градацию средств для оценивания, а также для решения вопросов, связанных с организацией адаптивного и персонализированного обучения. Тем самым преподаватель во время интеграции онлайн-курса в учебную дисциплину или учебную онлайн-среду вуза имеет возможность выбора подходящих настроек с учётом предметной области.

Прежде чем начать проводить дополнительные исследования в отношении доступности платформы для размещения онлайн-курса и оценивать её качество и ключевые характеристики, важно получить быстрое первое впечатление о её рейтингах путем оценки рейтинга партнёров и преподавателей, которые уже разработали и выложили свои открытые онлайн-курсы [10].

Будем различать 2 типа онлайн-курсов: курсы, созданные в отрыве от утверждённых учебных планов (в качестве источника дополнительного образования или по-

вышения квалификации), и разработку и внедрение онлайн-курса в дисциплину с уже утвержденным рабочим планом и лимитированными часами. В данной статье сделан акцент на второй вариант.

Средства изучения курса «Машинное обучение»

В настоящее время такими корпорациями, как Yandex, Tesla, Microsoft, IBM, Google, Amazon, возлагаются большие надежды на направление «Машинное обучение» (full self-driving, Tesla Bot с использованием deep learning, Yandex Zen). *Машинное обучение* (МО) — это научная дисциплина, сосредоточенная на двух взаимосвязанных вопросах: как можно создать компьютерные системы, которые автоматически совершенствуются с помощью опыта, и каковы фундаментальные статистические законы, управляющие всеми самообучающимися системами различной природы как в производственной, так и в социальной сферах.

Для построения таких методов используются средства математической статистики, численных методов, математического анализа, методов оптимизации, теории вероятностей, теории графов, а также различные техники работы с данными в цифровой форме [11].

Рассмотрим опыт преподавания обсуждаемой научной дисциплины, полученный в РГПУ им. А.И. Герцена на направлении 01.03.02 «Прикладная математика и информатика, уровень бакалавриата», модуль «Компьютерные науки» [12].

Цель обучения состоит в развитии компетенций в области работы с цифровыми данными через овладение системой знаний, умений и навыков, связанных с особенностями математических и информационных способов представления и обработки информации.

В обсуждаемом курсе под МО будем рассматривать лишь классическое МО. Последнее обосновано тем, что на дисциплину выделено всего 36 часов, поэтому вместо распределения небольшого количества часов на несколько объемных тем, таких как классическое МО, ансамблевые методы, обучение с подкреплением, нейронные сети, было решено сконцентрировать внимание на теме «Классическое МО», но при этом дополнить этот раздел подробным рассмотрением работы с возможностями визуализации результатов, и метриках оценки качества обученных моделей.

Раздел «Классическое МО» включает следующие темы: деревья решений, случайные леса, линейная и параболическая регрессия, метод опорных векторов и многие другие, и остается по-прежнему очень распространенным в различных областях исследований. Большинство классических методов машинного обучения были разработаны с учетом структурированных данных, т. е. данных в табличной форме, где наблюдения хранятся в виде строк, а сопутствующие признаки (особенности) — в виде столбцов.

Студенты, принимающие участие в эксперименте, изучали дисциплины:

- «Язык программирования Python 3» в объеме 56 часов;
- «Теория вероятностей и математической статистики» в объеме 113 часов;
- «Алгоритмы и структуры данных» — 59 часов;
- «Интеллектуальный анализ данных» (пакет Weka) в объеме 39 часов.

Нами был выбран путь, предполагающий погружение в используемые алгоритмы МО через программирование, включая возможность демонстрационной визуализации обученных моделей. Основной акцент был сделан на задачный подход, в качестве основного средства

для решения задач машинного обучения был задействован язык программирования Python 3 [13; 14].

Исторически сложилось так, что для проведения исследований в области МО и разработки приложений использовался широкий спектр различных языков программирования и сред. Однако, поскольку за последние 10–15 лет популярность языка Python общего назначения в научном сообществе, выполняющем исследования с помощью компьютерной обработки, значительно возросла, большинство новейших библиотек МО теперь основаны на языке Python [15; 16].

Для осуществления операций линейной алгебры над многомерными массивами, представляющих собой эффективные вычислительные структуры данных для представления векторов, матриц и тензоров более высокого порядка, были созданы высокопроизводительные и удобные в использовании библиотеки NumPy и SciPy. NumPy. Это библиотека многомерных массивов с фундаментальными операциями линейной алгебры, а библиотека SciPy дополняет массивы NumPy множеством важных примитивов (от численных оптимизаторов и обработки сигналов до статистики и разреженной линейной алгебры). По состоянию на 2021 год SciPy используется почти в половине всех проектов МО на GitHub.

Библиотека Pandas [17] расширяет NumPy, предоставляя объект, подобный таблице с данными, поддерживающий разнородные типы столбцов и метаданные строк и столбцов. В последние годы библиотека Pandas стала де-факто форматом представления табличных данных в Python.

Сегодня Pandas, NumPy и SciPy остаются наиболее удобными и рекомендуемыми вариантами для многих проектов в области науки о данных, включающую в себя раздел МО.

Scikit-learn является стандартной библиотекой Python, используемой для предобработки признаков и классического МО на наборах данных малого и среднего размера с менее чем 1000 наблюдениями, что и являлось в нашем случае верхней границей при изучении дисциплины МО [18]. Кроме того, Scikit-learn предоставляет программный интерфейс для оценки обученных моделей с использованием распространенных методов, таких как кросс-валидация. Авторы Scikit-learn оставили в качестве расширений большое количество новых методов МО, не интегрируя их в ядро. В рамках данного курса

мы ограничились лишь встроенными методами Scikit-learn, хотя новые и зачастую менее известные методы МО поддерживаются пакетом “Scikit-contrib” в рамках общей организации GitHub.

Чтобы можно было лучше определять границы принятия решения, в режиме реального времени для исследования параметров обученной модели применяют интерактивные визуализации. Для решения этой задачи в сообществе машинного обучения широко используется представление результатов исследований в виде графиков и диаграмм функционала библиотек Matplotlib и Seaborn [19].

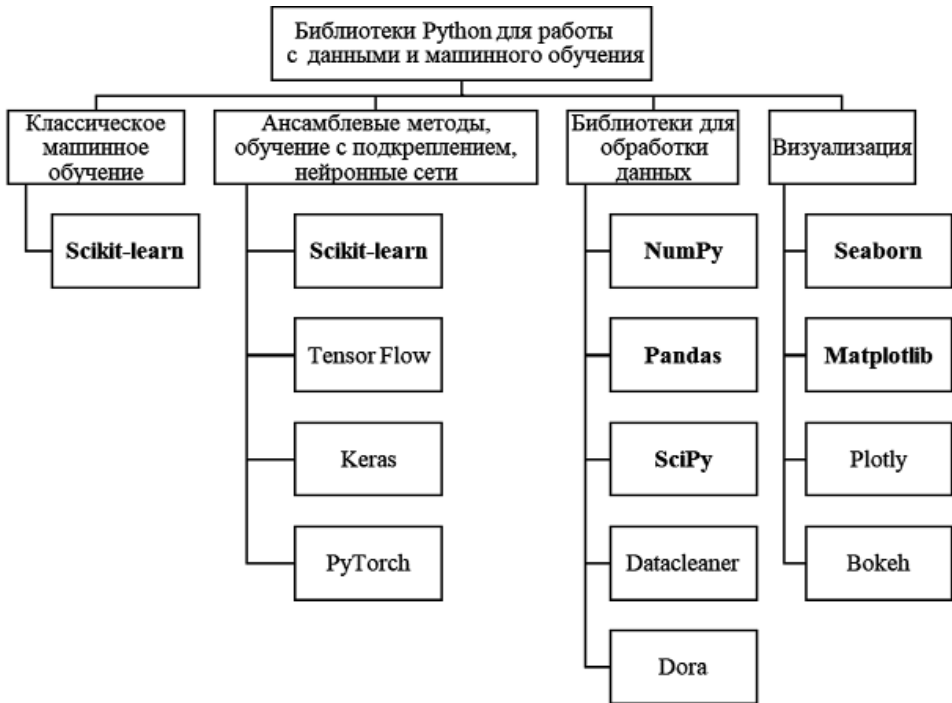


Рис. 1. Используемые библиотеки и их аналоги

Отобранные и используемые нами библиотеки Python для изучения раздела «Классическое МО» представлены на рис. 1 (выделены жирным шрифтом), к ним относятся:

- Pandas, NumPy и SciPy — для исследования данных, их структурирования, фильтрации, агрегации;
- Scikit-learn — содержит методы машинного обучения;

- Matplotlib, Seaborn — предоставляют возможность визуализировать данные или обученную модель (например, дерево решений).

Средства организации курса и содержание обучения при гибридном подходе

В настоящее время в российских вузах существует обязательное требование сопровождать любую дисциплину материалами в системе дистанционного обучения, которая определена вузом для этих целей. Однако функционал этих систем (например, система Moodle) оказывается довольно ограниченным и требует расширения за счет внешних средств, поэтому задача выбора эффективной платформы для размещения онлайн-курса по программированию достаточно нетривиальна, поскольку существует несколько десятков различных вариантов систем дистанционного обучения и поддержки образовательного процесса разной популярности (учитываются лишь платформы на русском и англоязычные). В работе [20] нами был представлен подробный обзор различных платформ для организации онлайн-курсов в области программирования и обоснован выбор в пользу платформы Stepik, которая, хотя и специализируется на технических курсах, имеет множество учебных материалов в различных областях знания. Например, существует большое количество математических курсов, рассчитанных на различный уровень подготовки слушателей. В пользу роста популярности платформы можно привести следующий факт: летом 2020 года было около 150 000 000 решений, а спустя два года — более 900 000 000.

Платформа позволяет отслеживать самостоятельность выполнения заданий, поддерживая подсчёт суммарного

количества всех имеющихся правильных решений по существующим курсам. Имеющиеся на Stepik чаты для пользователей достаточно активно модерируются, что резко сокращает количество списываний. Поэтому неважно, если два разных студента отправили идентичное решение, главное — правильность представленного решения. Мы ориентировались на безвозмездное пользование системой с ограничением несущественных функций.

Ключевое преимущество Stepik заключается в автоматической проверке задач по программированию. Оно выражается в двух вариантах:

- 1) добавлении вручную авторских тестов, на которых проверяется программный код, отправленный обучающимся;
- 2) генерировании автоматических тестов с помощью скрипта на Python.

Первый способ носит объективный характер, когда добавленные вручную тесты имеют обоснование, т. е. можно показать, что они покрывают все возможные случаи.

Второй способ при случайном генерировании может пропустить некоторые случаи (Stepik имеет ограничение на общий размер тестов для одной задачи). Для каждого теста происходит запуск программного кода, после чего передаются входные данные из теста и ожидаются выходные данные как результат работы. Далее происходит сравнение с эталонным решением. Если совпадение оказалось точным, то задача решена верно, иначе сообщается об ошибке.

В качестве дополнительного плюса стоит отметить разнообразие поддерживаемых платформой языков программирования: на платформе представлены более 10 интерпретаторов, среди которых особой популярностью пользуется язык Python.

Выбор инструмента для организации лекционной части опирается на необходимость во время лекции писать про-

граммный код, собирать и обрабатывать данные, проводить статистические тесты, осуществлять моделирование или рисовать графики.

Этим требованиям удовлетворяет интерактивная вычислительная платформа Jupyter Notebook (JN, см. <https://jupyter.org/>), предназначенная для поддержки рабочего процесса научных вычислений от интерактивного исследования до публикации подробных комментариев вычислений [21; 22]. Код в JN организован фрагментарно по ячейкам, которые могут быть автономно изменены и запущены. Выходные данные из каждой ячейки отображаются непосредственно под ней и сохраняются как часть документа. Текст JN возможно чередовать с программным кодом и сразу осуществлять вывод комментариев.

Доступ к JN осуществляется через современный веб-браузер. Это делает практичным использование одного и того же интерфейса, работающего локально (как настольное приложение), так и работающего на удаленном сервере, поэтому преподаватель может настроить программное обеспечение на сервере в целях предоставления доступа обучающимся. Файлы, созданные JN, представляют собой документированный формат JSON с расширением “.ipynb”, что позволяет написать новые программные средства, которые получают доступ к этим файлам и управляют ими.

Так как практически все лекционные часы отведены на работу с языком программирования Python и его библиотеками, то было решено использовать интерактивную вычислительную платформу JN. В таком случае весь теоретический материал удалось оформить в виде нескольких файлов с расширением .ipynb, к которым был доступ у каждого студента. Плюсом данного подхода явля-

ется добавление расширения, связанного с демонстрацией презентаций RISE (см. <https://rise.readthedocs.io/en/stable/installation.html>), т. е. в режиме реального времени был задействован функционал, который адаптирует ячейки JN в полноценные слайды, не уступающий по инструментарию известному многим PowerPoint. У студентов появляется возможность не только записывать/запоминать озвученный на лекциях синтаксис, но и параллельно локально принимать активное участие в лекциях и практических занятиях со своего компьютера, запуская исполнение ячейки JN или серии ячеек. Тем самым на лекциях удаётся избежать монолога «от одного ко многим», появляется возможность включать в процесс всю аудиторию слушателей. Ещё одно положительное дополнение заключается в возможности выполнять практические задания в интерактивном режиме.

Полный перечень всех средств обучения мультиплатформенного онлайн-курса обозначен на рис. 2.

Структурными единицами содержания обучения онлайн-курса на Stepik являются модули, уроки и степы. Модуль включает уроки, состоящие из набора фрагментов, называемых степами.

Поскольку все практические занятия проводились на платформе Stepik, то были выделены и добавлены три модуля:

- основы Python 3,
- предварительная обработка данных на Python 3,
- машинное обучение на Python 3.

Модуль «Основы Python 3» посвящён повторению абсолютно необходимого материала для дальнейшего изучения дисциплины «Машинное обучение», на основании чего были выделены следующие темы и отобрано следующее содержание обучения:

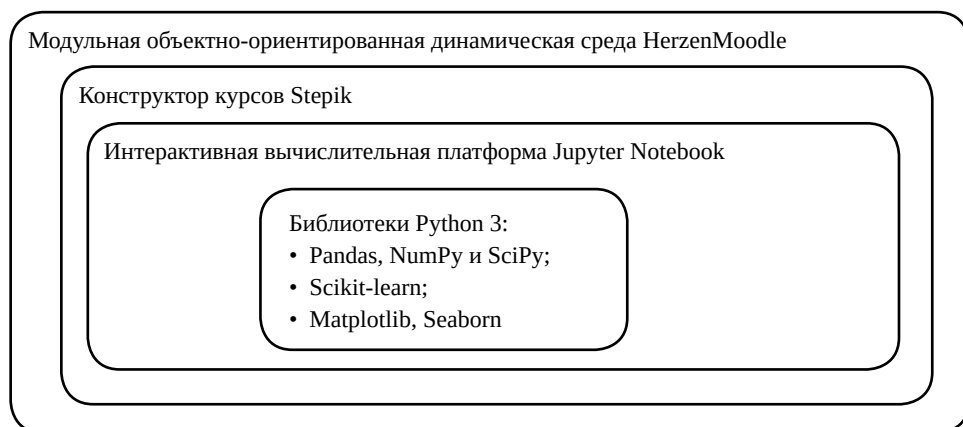


Рис. 2. Средства организации мультиплатформенного курса для изучения классического машинного обучения

1. Структуры данных, включающие List, Dictionary, Set, String, numbers (int, float), Tuple, Boolean, None. Упомянется про разницу между изменяемыми (List, Dictionary, Set) и неизменяемыми структурами данных (numbers (int, float), Tuple, Boolean, None). Для каждой структуры данных продемонстрировано следующее: инициализация экземпляра (List, Dictionary и т. д.), каким образом экземпляр хранится в памяти компьютера. Например, в случае List, Tuple выделяется непрерывный диапазон, а Dictionary, Set реализованы на хешировании, применении основных методов и атрибутов; вкратце упоминается про структуры данных Frozenset и complex numbers.

2. Функции, с помощью которых возможно считывать входные данные для исполнения программного кода и осуществления вывода результата.

3. Основные понятия, касающиеся асимптотических поведений функций. В частности, O — символика и определения O , Ω , Θ , o ; представлены таблицы временной сложности функций и методов Python, относящихся к представленным выше структурам данных.

4. Работа с файлами и файлоподобными объектами; разобраны все существующие режимы для чтения, записи, добавления, а также их комбинации.

Модуль «Предварительная обработка данных на Python 3» включала в себя использование следующих библиотек:

1. Библиотеки Pandas, NumPy и SciPy для считывания, фильтрации и преобразования к подходящему состоянию для дальнейшего использования в машинном обучении данных. Также задействованы методы и атрибуты в целях получения описательной статистики (`pandas.describe`, `pandas.dtypes`, `pandas.shape`, `pandas.size`), которая желательна до всех различных операций, связанных с преобразованием.

2. Библиотеки Matplotlib, Seaborn в целях получения визуализации исходных данных или структуры уже обученной модели. Одна значительная визуализация относилась к демонстрированию обученной модели — дерева решений, но используемый синтаксис выходил за рамки модуля «Основы Python 3» (применены библиотеки `re`, `ipywidgets`, `IPython.display`, `graphviz`), поэтому было дано лишь общее описание происходящего.

В основном из-за небольшого количества времени для проведения дисциплины визуализация относилась к описательной статистике.

Заключительный модуль «Машинное обучение на Python 3» был посвящён разбору и работе с классическими алгоритмами в МО: линейной регрессией, квадратической регрессией, деревом решений.

Остановимся подробнее на разборе работы алгоритма «Дерево решений».

Нами была задействована библиотека Scikit-learn. Так как Scikit-learn во время обучения модели для оценки качества использует метрики *Gini* или *Entropy* и зависимую от них *Information gain*, то со студентами подробно разбираются соответствующие математические основания этих метрик.

Одна значительная визуализация относилась к демонстрированию обученной модели — дерева решений, но используемый синтаксис выходил за рамки модуля «Основы Python 3» (применены библиотеки *re*, *ipywidgets*, *IPython.display*, *graphviz*), поэтому было дано лишь общее описание происходящего. В основном из-за небольшого количества времени для проведения дисциплины визуализация относилась к описательной статистике.

Обучение дерева решений (*sklearn.tree.DecisionTreeClassifier*) восходит к «жадному» алгоритму, т. е. к такому типу алгоритмов, работа которых не останавливается до тех пор, пока существует возможность улучшить показатели классификации. Это улучшение называется переобучением, и одно из распространённых решений — корректировка параметров. К таким процедурам относятся *criterion*, *max_depth*, *min_samples_split*, *min_samples_leaf*, *max_leaf_nodes*, поэтому некоторое время в курсе посвящено известным спосо-

бам поиска подходящих значений. Базовыми вариантами являются:

- *sklearn.model_selection.GridSearchCV*, перебирающий всевозможные комбинации заданных значений параметров;

- *sklearn.model_selection.RandomizedSearchCV*, который делает то же самое, но случайным образом выбирает лишь некоторые из всевозможных комбинаций (в целях экономии количества вычислений компьютера).

Представленные два способа имеют существенный недостаток, заключающийся в переобучении, потому что происходит подбор параметров на единственном определённом наборе данных, т. е. в данном подходе осуществляется не поиск значений для нахождения общей закономерности, а лишь частного случая. С помощью метода кросс-валидация демонстрируется эффективный способ подбора значений параметров дерева решений, описание которого присутствует в официальной документации Scikit-learn. Этот вариант разделяет набор данных для обучения на несколько частей необязательно одинакового размера, которые могут между собой пересекаться. Далее каждая часть всего один раз выступает в качестве тестовых данных, пока все остальные используются для обучения модели. На выходе имеется определённое количество оценок качества обученной модели, равное числу частей. В качестве общей оценки можно взять среднее от всех оценок. Описанный способ позволяет задействовать все данные и в обучении, и в тестировании. Кросс-валидация позволяет избежать переобучения.

Немаловажным моментом в использовании алгоритма дерева решений является понимание внутреннего устройства, но при этом надо уметь определять

итоговую пригодность обученной модели. Последнее покрывают метрики, демонстрирующие качество обученной модели. Хотя существует множество различных оценок, нами были выделены три наиболее активно применяющиеся, а именно: доля правильных ответов (у обученной модели метод score), точность (precision)

и полнота (recall). Также существуют другие, часть из которых представляют комбинацию названных метрик, например, F-мера (F-score), но они выходили за рамки рассматриваемой дисциплины. Общее содержание каждого из трёх перечисленных модулей продемонстрировано в табл. 1.

Таблица 1

Распределение библиотек и их содержимого по модулям

№ п/п	Название модуля	Библиотеки Python 3	Функционал
1	Основы Python 3	–	Структуры данных: List, Dictionary, Set, String, numbers (int, float), Tuple, Boolean, None, функции
2	Предварительная обработка данных на Python 3	Pandas	DataSeries
		NumPy	DataFrames
		SciPy	Алгоритмы для научных исследований
		Matplotlib	Визуализация
		Seaborn	Визуализация статистических данных
3	Машинное обучение	Scikit-learn	Регрессия, Деревья решений, Ансамблевые методы

Моделирование онлайн-курса на платформе Stepik

не исключены вопросы, связанные с добавлением используемых библиотек языка Python: Pandas, NumPy, SciPy, Scikit-learn, Matplotlib, Seaborn, поскольку по умолчанию интерпретатор Python не включает их. Авторы Stepik позаботились о необходимых зависимостях, поэтому любой обучающийся может импортировать перечисленные библиотеки для выполнения задания.

В завершение осталось добавить материалы практических занятий на платформу Stepik (см. <https://stepik.org/catalog>). Как упоминалось выше, платформа позволяет создавать задания различных типов. Для организации обучения по дисциплине «Машинное обучение» оказалось достаточно четырёх типов заданий:

- тест с выбором единственного правильного ответа,
- тест с выбором всех подходящих ответов из списка,
- численная задача,
- задача на программирование (тестируется через stdin → stdout).

Если с первыми тремя типами всё достаточно прозрачно (они моделируются функционалом Stepik), то с четвёртым

Ранее было отмечено о создании трёх модулей на Stepik (см. табл. 2) при этом тем из обучающихся, кто демонстрировал высокий уровень владения языком программирования Python и/или уже практиковался в области решения задач машинного обучения, в каждом модуле предлагалось заниматься решением задач повышенной сложности из области алгоритмов и структур данных. Поскольку обучающиеся к началу курса «Машинное обучение» уже изучали дисциплину «Алгоритмы и струк-

туры данных», то сложность представленных в этом разделе заданий была сопоставима с задачами, заключенными между уровнями 1300 и 1600 на всемирно известном Российском ресурсе по спортивному программированию Codeforces (см. <https://codeforces.com>). Комплекс задач является авторской разработкой, для их решения студенту необходимо иметь базовые знания по основным алгоритмам и структурам данных языка Python 3, например, достаточно материала из книги [23].

Если же студент осознал, что ему трудно самостоятельно справиться с задачами из области алгоритмов и структур данных, то была предусмотрена возможность

проконсультироваться с преподавателем или вернуться к базовому варианту, который целиком опирался на материалы лекций. Обязательным требованием в этом случае было выполнение всех пропущенных практических заданий в базовой части. Тем самым были добавлены элементы адаптивного обучения, студент мог скорректировать свой образовательный маршрут.

Соотношение между задачами базового уровня и заданиями повышенной сложности составляло примерно 5 к 1, поэтому на одну учебную неделю выделялось 15–16 базовых задач и 3 повышенной сложности.

Таблица 2

Структура курса на Stepik

№ п/п	Название модуля	Номера лекций	Общее количество задач базового уровня (по 15–16 задач после каждой лекции)	Задачи повышенной сложности на знание алгоритмов и структур данных (по 3 задачи после каждой лекции)		Проектная задача на семестр
				Сложность по шкале ресурса Codeforces	Общее количество задач	
1	Основы Python 3	1, 2, 3	47	1000–1200	9	1 задача с платформы Kaggle, связанная с классическим машинным обучением
2	Предварительная обработка данных на Python 3	4, 5	31	1300–1400	6	
3	Машинное обучение Python 3 (регрессия, деревья решений)	6, 7, 8	45	1500–1600	9	

121

Актуальность привлечения задач, посвящённых алгоритмам и структурам данных, подтверждается активным применением методов машинного обучения в области искусственного интеллекта. Например, студенты должны познакомиться с алгоритмом Дейкстры, а после возможен сравнительно плавный переход к алгоритму A^* , который чаще всего

обсуждается в книгах по искусственному интеллекту [24]. Общая структура разработанного курса приведена в табл. 2.

Третий вариант: в начале семестра было предложено выбрать на своё усмотрение индивидуальную задачу, связанную с МО. Подобный подход позволял засчитать успешное выполнение предложенной задачи для закрытия дисциплины.

Требовалось самостоятельно выделить набор данных, должным образом предобработать и использовать для обучения модели с дальнейшим оцениванием качества. Выбор метода согласовывался с преподавателем. Естественно, для подтверждения самостоятельного выполнения существовало условие, касающееся объяснения программного кода с демонстрацией результата. В целях сокращения затраченного времени на этап поиска и разметки случайных данных предоставлялась информация о платформе Kaggle (см. <https://www.kaggle.com/>). Это онлайн-платформа, посвящённая науке о данных, на которой проводятся соревнования для решения бизнес-задач, подбора персонала и научных исследований, предназначенной для специалистов по обработке данных и энтузиастов машинного обучения. При выборе индивидуальной задачи существовала альтернатива: предлагалось обучить модель для классификации данных или распознавания объектов, в качестве которых могли выступать рукописные числа, символы алфавита (латиница или кириллица), по желанию студента можно было обобщить и перейти к еще более сложным объектам.

Студентам рекомендовалось привлечь ресурсы онлайн-платформы, чтобы сразу перейти к этапам предобработки и обучения модели. Как и во второй траектории обучения, был возможен вариант возвращения к базовому варианту.

В пилотном эксперименте приняли участие 18 студентов 3 курса факультета математики, обучающихся в РГПУ им. А.И. Герцена по направлению 01.03.02 «Прикладная математика и информатика», уровень бакалавриата. 67% студентов решили придерживаться базового варианта обучения на основе лекций и заданий; 27,5% сначала решали задачи на знание алгоритмов и структур данных; только

5,5% обучаемых согласовали индивидуальное задание, используя ресурсы, предоставленные платформой Kaggle, а именно: набор данных по определённой тематике, безвозмездный доступ к графическим процессорам (GPUs) для обучения модели, основанной на нейронных сетях.

Все 27,5% испытуемых в среднем после попытки выполнения половины предложенных задач на знание алгоритмов и структур данных вернулись в раздел базовой части. В итоге 94,5% обучающихся успешно выполнили базовую часть до конца, а лишь 5,5% — индивидуальное задание.

Следует предположение о лёгкости базовой части или завышенной сложности задач на алгоритмы и структуры данных. В дальнейшем запланировано содержание базовой части оставить прежним и снизить уровень сложности задач на алгоритмы и структуры данных, а именно: 800–900 для первого модуля, 1000–1200 — второго и 1300–1400 — третьего.

Заключение

В настоящей статье на примере дисциплины «Машинное обучение» представлен вариант мультиплатформенного авторского онлайн-курса. Описанный подход позволил сократить время, затрачиваемое на проверку практических заданий преподавателем, а у студентов предположительно повысилась производительность, т. к. они могли выполнять задания как в аудитории, так и удаленно и быстро получать обратную связь. Этого удалось достичь благодаря гибриднему задействованию современных средств: интерактивной вычислительной платформы Jupyter notebook для оформления и проведения лекций, а также многофункциональной и гибкой платформы Stepik для создания образовательных материалов: для добав-

ления практических заданий с автоматической проверкой программного кода.

Обучающимся было предложено несколько вариантов развития собственной образовательной траектории в рамках дисциплины «Машинное обучение»:

1) базовый вариант, выполнение практических заданий которого опиралось на лекции;

2) продвинутый вариант, включающий задачи на алгоритмы и структуры данных;

3) индивидуальное задание по машинному обучению, согласованное с преподавателем.

Было решено не устанавливать жёсткие рамки, поэтому каждый студент выбирал собственную траекторию. В течение семестра существовала возможность вернуться к базовому варианту, опирающемуся на материал лекций.

В дальнейшем планируется развить адаптивные характеристики курса за счет функционала платформы Stepik для оперативного автоматического перестраивания образовательного маршрута обучающегося на основании его активностей и объективных результатов, а не только личных предпочтений.

СПИСОК ИСТОЧНИКОВ И ЛИТЕРАТУРЫ

1. Каракозов, С.Д., Рыжова, Н.И., Уваров, А.Ю. Условия результативности системной трансформации учебного процесса на основе ИКТ в образовательном учреждении // Инновационные технологии в медиаобразовании: материалы II Международной научно-практической конференции. СПб.: Санкт-Петербургский государственный институт кино и телевидения, 2018. С. 227–235.
2. Беспалько, А.А. Особенности подготовки учебно-методических материалов к дистантным занятиям по программированию / А.А. Беспалько, Н.В. Сочнева, А.П. Тощев, О.И. Тутынина // Современные наукоемкие технологии. 2021. № 9. С. 154–159.
3. Рыжова, Н.И., Филимонова, Е.В., Королева, Н.Ю. Составляющие информационно-вычислительной деятельности как основа решения цифровых образовательных задач // Информатизация образования и методика электронного обучения: цифровые технологии в образовании. Материалы V Международной научной конференции / общ. ред. М.В. Носкова. Красноярск, 2021. С. 290–296.
4. Каракозов, С.Д. Подготовка бакалавров педагогического образования к реализации проектно-исследовательской деятельности в условиях цифровизации школы / С.Д. Каракозов, Н.И. Рыжова, Н.Ю. Королева, Е.В. Филимонова // Вестник Российского университета дружбы народов. Серия: Информатизация образования. 2021. Т. 18. № 2. С. 115–127.
5. Пиотровская, К.Р. Экспериментальное исследование применения корпусного инструментария при моделировании предметно-языкового скаффолдинга // Новые образовательные стратегии в современном информационном пространстве: сборник научных статей по материалам международной ежегодной научно-практической конференции. СПб.: Астерион, 2021. С. 61–70.
6. Bralić, A., Divjak, B. Integrating MOOCs in Traditionally Taught Courses: Achieving Learning Outcomes with Blended Learning // International Journal of Educational Technology in Higher Education. 2018. Vol. 15 (2). P. 1–16.

7. Jung, Y, Lee, J. Learning Engagement and Persistence in Massive Open Online Courses (MOOCs) // Computers & Education. 2018. Vol. 122. P. 9–22.
8. Ананин, Д.П., Стрикун, Н.Г. Гибридное обучение в структуре высшего образования: между онлайн и офлайн // Преподаватель XXI век. 2022. № 4–1. С. 60–74.
9. Lujak, M. Objective Tests in Automated Grading of Computer Science Courses: An Overview / M. Lujak, M. Slavkovik, A. Lebis, M. Vermeulen, A. Doniec // Handbook on Intelligent Techniques in the Educational Process. Learning and Analytics in Intelligent Systems. 2022. Vol. 29. P. 239–268.
10. Chapman, S.A. A Strategy for Monitoring and Evaluating Massive Open Online Courses / S.A. Chapman, S. Goodman, J. Jawitz, A. Deacon // Eval Program Plann. 2016. Vol. 57. P. 55–63.
11. Jordan, M.I., Mitchell, T.M. Machine Learning: Trends, Perspectives and Prospects // Science Review. 2015. Vol. 349 (6245). P. 255–260.
12. Piotrowska, X., Terbusheva, E. Educational Data-Mining for Future Educational Employees // New Educational Strategies in Modern Information Space: Proceedings of the XIV International Conference. St. Petersburg, 2019. P. 38–49.
13. Козьмо, Л., Ричарт, В. Построение систем машинного обучения на языке Python. М.: ДМК Пресс, 2022. 302 с.
14. Воронцов, К.В. Машинное обучение. URL: <https://ya-r.ru/2020/05/07/vorontsov-kurs-mashinnoe-obuchenie-2019-shkola-analiza-dannyh/> (дата обращения: 03.01.2023).
15. Raschka, S, Mirjalili, V. Python Machine Learning: Machine Learning and Deep Learning with Python, Scikit-Learn, and Tensor Flow 2. Third Edition. Birmingham: Packt Publishing Ltd., 2019.
16. The State of the Octoverse 2020. URL: <https://octoverse.github.com> (дата обращения: 22.01.2023).
17. Hagedorn, S, Kläbe, S, Sattler, K.U. Putting Pandas in a Box // The Annual Conference on Innovative Data Systems Research (CIDR). 2021. URL: https://www.cidrdb.org/cidr2021/papers/cidr2021_paper07.pdf (дата обращения: 23.01.2023).
18. Bisong, E. Introduction to Scikit-Learn // Building Machine Learning and Deep Learning Models on Google Cloud Platform. Berkeley: Apress, 2019. P. 215–229.
19. Iyer, S.G., Pawar, A.D. Machine Learning Model for Predicting Price of Processors Using Multivariate Linear Regression // International Conference on Smart Systems and Inventive Technology (ICSSIT). 2019. P. 52–56.
20. Никандров, А.А. Многофункциональные и гибкие онлайн-платформы для создания образовательных материалов // Информатика и образование. 2022. Т. 37. № 6. С. 22–29.
21. Johnson, J.W. Benefits and Pitfalls of Jupyter Notebooks in the Classroom // Proceedings of the 21st Annual Conference on Information Technology Education. 2020. P. 32–37.
22. Pérez, F., Granger, B.E. IPython: A System for Interactive Scientific Computing // Computing in Science & Engineering. 2007. Vol. 9 (3). P. 21–29.
23. Бхаргава, А. Прокаем алгоритмы. Иллюстрированное пособие для программистов и любопытствующих. СПб.: Питер, 2017. 288 с.
24. Hetland, M.L. Python Algorithms: Mastering Basic Algorithms in the Python Language. NY: Apress, 2014.

REFERENCES

1. Karakozov, S.D., Ryzhova, N.I., Uvarov, A.Yu. Usloviya rezultativnosti sistemnoj transformacii uchebnogo processa na osnove IKT v obrazovatelnom uchrezhdenii [Conditions for the Effectiveness of the Systemic Transformation of the Educational Process Based on Information and Communication Technologies in an Educational Institution]. In: *Innovacionnye tekhnologii v mediaobrazovanii* [Innovative Technologies in Media Education: Materials of the II International Scientific and Practical Conference]. St. Petersburg, Sankt-Peterburgskij gosudarstvennyj institut kino i televideniya, 2018, pp. 227–235. (in Russ.)
2. Bsepalko, A.A., Sochneva, N.V., Toshchev, A.P., Tutynina, O.I. Osobennosti podgotovki uchebno-metodicheskikh materialov k distantnym zanyatiyam po programmirovaniyu [Features of the Preparation of Teaching Materials for Remote Programming Classes], *Sovremennye naukoemkie tekhnologii* = Modern High-Tech Technologies, 2021, No. 9, pp. 154–159. (in Russ.)
3. Ryzhova, N.I., Filimonova, E.V., Koroleva, N.Yu. Sostavlyayushchie informacionno-vychislitelnoj deyatel'nosti kak osnova resheniya cifrovyyh obrazovatelnyh zadach [Components of Information and Computing Activity as a Basis for Solving Digital Educational Problems]. In: *Informatizatsiya obrazovaniya i metodika elektron'nogo obucheniya: cifrovye tekhnologii v obrazovanii* [Informatization of Education and Methods of E-Learning: Digital Technologies in Education. Materials of the V International Scientific Conference], ed. by M.V. Noskov. Krasnoyarsk, 2021, pp. 290–296. (in Russ.)
4. Karakozov, S.D., Ryzhova, N.I., Koroleva, N.Yu., Filimonova, E.V. Podgotovka bakalavrov pedagogicheskogo obrazovaniya k realizacii proektno-issledovatel'skoj deyatel'nosti v usloviyakh cifrovizatsii shkoly [Preparation of Bachelors of Pedagogical Education for the Implementation of Design and Research Activities in the Conditions of School Digitalization], *Vestnik Rossijskogo universiteta druzhby narodov. Seriya: Informatizatsiya obrazovaniya* = Bulletin of the Peoples' Friendship University of Russia. Series: Informatization of Education, 2021, vol. 18, No. 2, pp. 115–127. (in Russ.)
5. Piotrovskaya, K.R. Eksperimental'noe issledovanie primeneniya korpusnogo instrumentariya pri modelirovanii predmetno-yazykovogo skaffoldinga [Experimental Study of the Use of Corpus Tools in Modeling Subject-Language Scaffolding]. In: *Novye obrazovatel'nye strategii v sovremennom informacionnom prostranstve* [New Educational Strategies in the Modern Information Space: A Collection of Scientific Articles Based on the Materials of the International Annual Scientific and Practical Conference]. St. Petersburg, Asterion, 2021, pp. 61–70. (in Russ.)
6. Bralić, A., Divjak, B. Integrating MOOCs in Traditionally Taught Courses: Achieving Learning Outcomes with Blended Learning, *International Journal of Educational Technology in Higher Education*, 2018, vol. 15 (2), pp. 1–16.
7. Jung, Y, Lee, J. Learning Engagement and Persistence in Massive Open Online Courses (MOOCS), *Computers & Education*, 2018, vol. 122, pp. 9–22.
8. Ananin, D.P., Strikun, N.G. Gibridnoe obuchenie v strukture vysshego obrazovaniya: mezhdru onlajn i oflajn [Hybrid Learning in the Structure of Higher Education: Between Online and Offline], *Prepodavatel XXI vek* = Russian Journal of Education, 2022, No. 4–1, pp. 60–74. (in Russ.)

9. Lujak, M., Slavkovik, M., Lebis, A., Vermeulen, M., Doniec, A. Objective Tests in Automated Grading of Computer Science Courses: An Overview. *Handbook on Intelligent Techniques in the Educational Process. Learning and Analytics in Intelligent Systems*, 2022, vol. 29, pp. 239–268.
10. Chapman, S.A., Goodman, S., Jawitz, J., Deacon, A. A Strategy for Monitoring and Evaluating Massive Open Online Courses, *Eval Program Plann*, 2016, vol. 57, pp. 55–63.
11. Jordan, M.I., Mitchell, T.M. Machine Learning: Trends, Perspectives and Prospects, *Science Rewie*, 2015, vol. 349 (6245), pp. 255–260.
12. Piotrowska, X., Terbusheva, E. Educational Data-Mining for Future Educational Employees. *New Educational Strategies in Modern Information Space: Proceedings of the XIV International Conference*. St. Petersburg, 2019, pp. 38–49.
13. Koelo, L., Richart, V. *Postroenie sistem mashinnogo obucheniya na yazyke Python* [Building Machine Learning Systems in Python]. Moscow, DMK Press, 2022, 302 p. (in Russ.)
14. Voroncov, K.V. *Mashinnoe obuchenie* [Machine Learning]. Available at: <https://ya-r.ru/2020/05/07/vorontsov-kurs-mashinnoe-obuchenie-2019-shkola-analiza-dannyh/> (accessed: 03.01.2023). (in Russ.)
15. Raschka, S, Mirjalili, V. *Python Machine Learning: Machine Learning and Deep Learning with Python, Scikit-Learn, and Tensor Flow 2. Third Edition*. Birmingham, Packt Publishing Ltd., 2019.
16. *The State of the Octoverse 2020*. Available at: <https://octoverse.github.com> (accessed: 22.01.2023).
17. Hagedorn, S, Kläbe, S, Sattler, K.U. Putting Pandas in a Box. *The Annual Conference on Innovative Data Systems Research (CIDR)*, 2021. Available at: https://www.cidrdb.org/cidr2021/papers/cidr2021_paper07.pdf (accessed: 23.01.2023).
18. Bisong, E. Introduction to Scikit-Learn. *Building Machine Learning and Deep Learning Models on Google Cloud Platform*. Berkeley, Apress, 2019, pp. 215–229.
19. Iyer, S.G., Pawar, A.D. Machine Learning Model for Predicting Price of Processors Using Multivariate Linear Regression. *International Conference on Smart Systems and Inventive Technology (ICSSIT)*, 2019, pp. 52–56.
20. Nikandrov, A.A. Mnogofunkcionalnye i gibkie onlajn-platformy dlya sozdaniya obrazovatelnykh materialov [Multifunctional and Flexible Online Platforms for Creating Educational Material], *Informatika i obrazovanie* = Informatics and Education, 2022, vol. 37, No. 6, pp. 22–29. (in Russ.)
21. Johnson, J.W. Benefits and Pitfalls of Jupyter Notebooks in the Classroom. *Proceedings of the 21st Annual Conference on Information Technology Education*, 2020, pp. 32–37.
22. Pérez, F., Granger, B.E. IPython: A System for Interactive Scientific Computing, *Computing in Science & Engineering*, 2007, vol. 9 (3), pp. 21–29.
23. Bhargava, A. *Grokaem algoritmy. Illyustrirovannoe posobie dlya programmistov i lyubopystvuyushchih* [Grokaem Algorithms. Illustrated Manual for Programmers and Curious]. St. Petersburg, Piter, 2017, 288 p. (in Russ.)
24. Hetland, M.L. *Python Algorithms: Mastering Basic Algorithms in the Python Language*. NY, Apress, 2014.

Никандров Алексей Андреевич, аспирант, кафедра цифрового образования, Российский Государственный педагогический университет им. А.И.Герцена, nik190397@mail.ru

Alexey A. Nikandrov, Postgraduate Student, Digital Education Department, A.I. Herzen Russian State Pedagogical University, nik190397@mail.ru

Пиотровская Ксения Раймондовна, доктор педагогических наук, кандидат технических наук, доцент, профессор, кафедра методики обучения математике и информатике, Российский Государственный педагогический университет им. А.И. Герцена; кафедра математической лингвистики, Санкт-Петербургский государственный университет, krp62@mail.ru

Xsenia R. Piotrowska, ScD in Education, PhD in Engineering, Associate Professor, Professor, Methods of Teaching Mathematics and Computer Science Department, A.I. Herzen Russian State Pedagogical University, Mathematical Linguistics Department, Saint-Petersburg State University, krp62@mail.ru

Статья поступила в редакцию 05.05.2023. Принята к публикации 30.06.2023

The paper was submitted 05.05.2023. Accepted for publication 30.06.2023