

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение высшего образования
«Тольяттинский государственный университет»

Институт математики, физики и информационных технологий
(наименование института полностью)

Кафедра Прикладная математика и информатика
(наименование)

09.04.03 Прикладная информатика
(код и наименование направления подготовки)

Управление корпоративными информационными процессами
(направленность (профиль))

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА (МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ)

на тему «Исследование методов и технологий создания учебного контента с применением технологий искусственного интеллекта»

Обучающийся

Н.А. Моисеева

(Инициалы Фамилия)

(личная подпись)

Научный руководитель

к.п.н., доцент, Е.А. Ерофеева

(ученая степень (при наличии), ученое звание (при наличии), Инициалы Фамилия)

Тольятти 2023

Оглавление

Введение.....	3
Глава 1 Теоретические основы методов и технологий искусственного интеллекта для образования.....	8
1.1 Направления исследований искусственного интеллекта в сфере образования.....	8
1.2 Обзор подходов к построению рекомендаторов.....	12
1.3 Алгоритмы и метрики точности рекомендаторов	22
Глава 2 Практическая реализация	34
2.1 Выбор языка программирования и цифровой среды для разработки алгоритмического обеспечения рекомендатора персонального учебного контента	34
2.2 Реализация алгоритма рекомендательной системы для изучения дисциплины	38
2.3 Реализация алгоритма рекомендательной системы по выбору массового онлайн-курса.....	46
Глава 3 Анализ результатов	56
Заключение	61
Список используемой литературы	63
Приложение А Анкета.....	72

Введение

На сегодняшний день в развитие, внедрение и эволюцию технологий искусственного интеллекта (ИИ) вкладывают все больше денег и сил, т. к. ИИ стал неотъемлемой и вездесущей, хоть и не всегда явно видимой частью нашей жизни: от Siri до автоматизированной журналистики, от биржевых прогнозов до профилактики преступности, от распознавания лиц до медицинской диагностики и т.д. Необходимо заметить, что ИИ незаметно и прочно проник и в сферу образования [47]. Так называемые интеллектуальные, адаптивные или персонализированные системы обучения все чаще применяются как в школах, так и вузах по всему миру, собирают и анализируют огромные объемы данных (Big Data) и при этом существенно влияют как на процесс, так и качество обучения [21, 22, 37, 58].

На данный момент в национальной системе образования на всех уровнях, как и во всем мире, особо актуально решение проблемы проектирования персонального образовательного трека обучаемого, в том числе учебного контента с учетом его познавательных потребностей, что является одной из важных составляющих адаптивного обучения, которое определяет перспективные образовательные технологии и системы в условиях глобальной цифровой экономики. В этой связи, рекомендательные системы являются технологической составляющей в формировании персонального учебного контента обучаемого по дисциплине.

Анализ современного состояния проблемы исследования [2, 8, 12, 19, 22, 24, 33, 36, 41, 47, 58, 64] заключается в разрешении противоречия между существующим потенциалом систем рекомендаций для формирования индивидуального учебного контента обучаемого, т. е. его персонального образовательного трека по дисциплине, и недостаточным уровнем программно-алгоритмической реализации рекомендаторов в системе национального образования.

В этой связи, в настоящей работе исследуется следующий аспект: разработка прототипа РС, который опираясь на данные о результатах итогового тестирования обучаемых по некоторой дисциплине, может использоваться в формировании персонального учебного контента для обучаемого. Также, такая система сможет рекомендовать существующие массовые онлайн-курсы (МООК), основываясь на соответствующих исходных учебных данных студента по дисциплине.

Цель работы — разработка алгоритмического обеспечения прототипа рекомендательной системы на основе методов и технологий машинного обучения для формирования персонального учебного контента обучаемого, основываясь на анализе его итоговых учебных данных по дисциплине.

Объект исследования — рекомендательная система для формирования персонального учебного контента обучаемого по дисциплине.

Предмет исследования — алгоритмическое обеспечение прототипа рекомендательной системы на основе методов и технологий машинного обучения.

Для достижения вышеуказанной цели были поставлены следующие задачи:

1. Анализ влияния методов и технологий ИИ, в частности рекомендательных систем, на формирование учебного контента.
2. Аналитический обзор подходов к построению рекомендательных систем.
3. Изучить алгоритмы построения рекомендательных систем и необходимые библиотеки языка программирования Python для их разработки.
4. Выявить возможную конфигурацию библиотек анализа данных и машинного обучения Python для реализации систем рекомендации.
5. Разработать алгоритмическое обеспечение рекомендательной системы на языке программирования Python и ее прототип.
6. Экспериментально проверить результативность применения разработанного прототипа системы рекомендации в учебном процессе.

Обзор и анализ методологий и технологий по теме исследования:

- методы и технологии искусственного интеллекта (Л. С. Болотова, Т. А. Гаврилова, Д. И. Муромцев и др.);
- искусственный интеллект в образовании (Г. А. Атанов, Я. А. Ваграменко, С. Даггэн, О. П. Околелов, И. Н. Пустынникова, У. Холмс и др.);
- язык программирования Python (К. Берман, Т. Гэддис, Д. Златопольский, У. Маккинни, Н. Прохоренок, Дж. Плас, Н. Седер, С. Рашка, Д. Шихи и др.)
- алгоритмы и технологии машинного обучения (А. Н. Аверьянов, Дж. Плас, С. Рашка, В. Н. Садовский, Э. Г. Юдин и др.); рекомендательные системы (R. Burke, A. J. Fernández-García, K. Фальк и др.);
- методология объектно-ориентированного (Г. Буч, А. М. Вендров, Д. Рамбо, А. Джекобсон и др.) и структурного подхода к проектированию программного обеспечения (Г. Н. Калянов, Д. А. Марка, К. Ю. МакГоун, Ф. Тельнов, С. В. Черемных и др.).

При выборе соответствующих методов исследования ВКР учитывалась специфика проблемы исследования, в соответствии с которой необходимо было провести анализ и синтез специальной и научно-технической литературы, в том числе научных публикаций [10].

Для получения количественной оценки используемых алгоритмов и моделей машинного обучения использовались соответствующие оценки и критерии моделей, которые им присущи и подробно описаны в источниках [41, 44, 51].

Для получения качественной оценки разработанного прототипа рекомендательной системы с целью формирования индивидуального учебного материала обучаемого по дисциплине использовались непосредственно

тестирование и анкетирование потенциальных пользователей – это профессорско-преподавательский состав (ППС) и студенты.

Для решения поставленных задач данной ВКР использовались следующие методы анализа и обработки результатов исследования:

- теоретические: анализ и синтез научной и технической литературы по проблеме исследования; изучение научных публикаций;
- эмпирические: оценка моделей машинного обучения [38, 41], используемых в задаче кластеризации результатов тестирования обучаемых, и рекомендательных систем [51]; анкетирование ППС, тестирование обучаемых;
- анализ полученных результатов: статистико-математическая обработка данных и их графическая интерпретация [14, 35].

Новизна исследования. Предлагается внедрить систему рекомендаций в блок контроля и диагностики электронного учебно-методического комплекса дисциплины, что позволит формировать персональный учебный контент обучаемого.

Теоретическая значимость исследования заключается в выявлении потенциала методов и технологий ИИ для создания учебного контента по дисциплине с целью формирования персонального образовательного трека обучаемого на основе его итоговых учебных данных.

Личное участие автора. Автор самостоятельно проводила анализ литературы по тематике исследования, планировала компьютерные эксперименты и получила все представленные в работе данные и выводы. Лично публиковала результаты, представленные в данном исследовании, в сборниках трудов международных научно-практических конференций и периодическом издании.

Практическая значимость исследования. Разработан прототип рекомендатора для формирования индивидуального учебного контента обучаемого, который может быть интегрирован в электронный учебно-методический комплекс дисциплины.

На защиту выносятся

1. Произведенный отбор методов и технологий ИИ, который реализуется на языке программирования Python, позволяет реализовать программно-алгоритмическое обеспечение рекомендательной системы по формированию индивидуального учебного контента обучаемого.

2. Возможности рекомендаторов, как из одного из популярных технологий ИИ, позволяют создать учебный контент обучаемого с учетом его образовательных результатов по дисциплине.

Апробация результатов исследования осуществлялась на протяжении всего исследования. Результаты исследования были представлены на двух международных научно-практических конференциях: «Информатика: проблемы, методы, технологии», «Информатизация образования: теория и практика» и опубликованы в сборниках трудов этих конференций; две публикации в периодических изданиях.

Структура ВКР: Магистерская диссертация содержит введение, три главы, выводы по главам, заключение, библиографический список литературы, приложение.

Глава 1 Теоретические основы методов и технологий искусственного интеллекта для образования

1.1 Направления исследований искусственного интеллекта в сфере образования

На самом деле исследования искусственного интеллекта (ИИ) в образовании идут уже около 50 лет, однако за последние годы в разработку продуктов AIED¹ (Artificial Intelligence in Education – Искусственный интеллект в образовании) [47] вложили миллионы долларов такие гиганты, как Amazon, Google и Facebook, присоединившись к известным мультимиллионным профильным компаниям вроде Knewton и Carnegie Learning. Фонд Global Learning XPrize объявил приз в 15 миллионов долларов за создание программы, которая позволит детям контролировать свое обучение.

Технологии AIED в качестве научного инструмента зачастую связан с двумя другими частично перекрывающимися областями, где используются статистические методы исследования больших данных [47]:

- анализ ученических данных – подразумевается измерение, сбор, анализ и представление данных об учащихя и их окружении в целях понимания и оптимизации обучения и условий, в которых оно происходит;
- глубинный анализ информации образовательного назначения – подразумевается сбор и анализ данных для понимания, поддержки и улучшения качества обучения.

Рассмотрим понятие искусственного интеллекта.

¹ Искусственный интеллект в образовании (AIED) имеет 3 значения: 1) «обучение для понимания ИИ» – развитие компетенций понимать технологии ИИ и применять ИИ; 2) «обучение с использованием ИИ» – применение для эффективного преподавания предмета/ дисциплине; 3) «обучение экспертов по ИИ» – подготовка специалистов, разрабатывающих ИИ, и учителя, преподающие ИИ.

Под ИИ понимается способность компьютера или других технических устройств демонстрировать или моделировать интеллектуальное поведение, которое свойственно человеку [11, 47].

Обзор научных статей и материалов конференции [13, 29, 32, 47, 49, 65, 67] показывает, что AIED охватывает широкий диапазон программных продуктов от персонализированных учебных и диалоговых систем до исследовательских сред обучения, анализа письменных работ, интеллектуальных игровых сред, чат-ботов и основанной на ИИ технологии подбора репетиторов.

Некоторые общеобразовательные школы уже внедряют ИИ в качестве отдельной учебной программы, ведутся разработки по его использованию для улучшения онлайн-обучения и повышения квалификации учителей. В целом присутствие интеллектуальных технологий в образовании становится все более явным, и, по прогнозам, к 2024 году размер этого рынка достигнет около шести миллиардов долларов.

Аналитический обзор научных статей и материалов конференций [2]–[6, 8, 9, 12, 25, 26, 33, 42, 46, 62, 74] показывает, что AIED охватывает широкий диапазон продуктов от пошаговых персонализированных учебных и диалоговых систем до исследовательских сред обучения, анализа письменных работ, интеллектуальных игровых сред, чат-ботов и основанной на ИИ технологии подбора репетиторов. Технически это выглядит как самостоятельная работа обучаемых на компьютере, образовательная практика, диагностика знаний и умений и т.д.

Прежде чем рассмотреть ИИ в образовательном контексте, рассмотрим понятие искусственного интеллекта в целом.

Некоторые исследователи предпочитают термин «расширенный интеллект», в котором единственным источником интеллекта остается человеческий мозг, а компьютер и его программы – это сложный инструмент, с помощью которого люди могут расширять или усиливать свои интеллектуальные возможности [47].

Аналитический обзор научных трудов [2, 8, 12, 19, 32, 54]-[56, 60, 61]-[74], в которых исследуются проблемы и вопросы имплементации интеграции методов и средств ИИ в систему образования на всех уровнях. Один из путей интеграции технологий ИИ в сферу образования – персонализация образовательного трека обучаемого [32]. К сожалению, ученые отмечают в своих исследованиях [8, 47, 59, 61, 63] следующий факт: процедура оценки учебных заданий осуществляется по некоему шаблону, что, в свою очередь, не может позволить персонализировать процесс обучения учащегося.

В XXI веке С. Даггэн [22] констатирует, что именно технологии ИИ определяют ведущую роль в имплементации концепции инновационного адаптивного обучения и проектирования персонального образовательного трека учащегося.

На данный момент актуальны и востребованны три наиболее перспективных направления применения технологий ИИ в национальной системе образования [21]:

- «Модели персонализированного адаптивного обучения как элементы современной цифровой образовательной инфраструктуры»,
- «Использование методов искусственного интеллекта для сбора и обработки данных оценивания в образовании»,
- «"Instructional Design Dashboards" для педагогического дизайна образовательных продуктов».

Аналитический обзор научных работ и специальной литературы [54, 55, 59, 65] по тематике исследования сосредоточен на разработках рекомендаций для сферы электронного обучения, в частности адаптивного обучения в цифровой среде [56]-[59, 67, 68, 74]. В работе [54] цель оптимальной траектории обучения – предоставить обучаемым наиболее подходящий учебный объект для изучения, руководствуясь познавательными потребностями обучаемого и индикаторами знаний (компетенций).

В научной статье [60] авторами продемонстрирована модель автоматизированного мониторинга, интегрированную в Web-сервис Maths Garden (<https://math-garden.com/>), в среде которого детей обучают математике. Используя данную модель ответа на вопросы, основанную на рейтинговой системе Elo (1978) и принципе подсчета баллов для последующей оценки способностей людей. При этом уровень сложности вопросов предлагается обновлять с каждым новым ответом, что, таким образом, позволяет проводить своевременную коррекцию или модификацию соответствующего элемента учебного контента по математике.

В научной работе [62] описана разработанная адаптивная, краудсорсинговая Web-платформа образовательного назначения, которая рекомендует задать набор индивидуальных вопросов в зависимости от уровня знаний и познавательных потребностей обучаемого.

Научная статья [73] представляет концепции электронного обучения с интеграцией систем рекомендаций (или так называемых рекомендательных систем, рекомендаторов) и техник глубокого обучения. Авторы рассматривают основные подходы к разработке рекомендательной системы (РС) с использованием обратной связи, а также различные практики, реализованные для создания РС с целью повышения качества обучения.

В исследовании [65] предлагается РС для рекомендации некоторой модели курсов, которая основана на совместной фильтрации, с целью помочь студентам и академическим консультантам (ассистентам) во время их регистрации на тот или иной учебный курс. Для разработки РС авторы используют методологию совместной фильтрации, которая дает рекомендации, основанные на успеваемости обучаемых по предыдущим предметам. В качестве меры сходства была выбрана корреляция Пирсона.

В работе [74] авторами представлена реализация в Python алгоритма рекомендации образовательных ресурсов на основе потребностей пользователя, где эти ресурсы будут получены за счет использования API YouTube, который предлагает видео учебных материалов,

ориентированных на уровень знаний учащегося, для настройки рекомендаций, принимая во внимание данные, полученные в результате опроса знаний, и академические оценки, полученные из различных ресурсов, созданных на платформе системы дистанционного обучения Moodle.

На сегодняшний день нет устоявшегося определения РС.

В самом общем виде рекомендательные системы — это алгоритмы, направленные на то, чтобы предлагать пользователям релевантные элементы (предметы, которые представляют собой, например, фильмы для просмотра, книги для чтения, продукты для покупки или что-либо еще в соответствии со спецификой отрасли) [1, 51].

В качестве РС рассматривается тип алгоритма машинного обучения (Machine Learning – ML), который предоставляет потребителям «соответствующие» рекомендации. Когда ищем что-то где угодно, будь то в приложении или в нашей поисковой системе, эта система рекомендаций используется для предоставления нам релевантных результатов. Они используют класс алгоритмов, чтобы найти релевантную рекомендацию для пользователя.

1.2 Обзор подходов к построению рекомендаторов

Формально РС можно представить следующим образом. Имеется множество пользователей $u \in U$, множество объектов $i \in I$ и множество событий $(r_{ui}, u, i, \dots) \in D$ (действия, совершаемые над объектами пользователями). Пользователем u задается каждое событие, объектом i , своим результатом r_{ui} и, возможно (но не обязательно), другими характеристиками.

В соответствии с этим, в базовый функционал РС входит следующее:

– предсказывать интересы/потребности пользователя u к объекту

$$i: r_{ui} = \text{Predict}_k(u, i, \dots) \approx r_{ui};$$

– формировать персональные рекомендации для пользователя u :

$$\text{Recommend}_k(u, \dots) = (i, \dots, i_k);$$

– выявлять объекты, которые похожие на объект i :

$$\text{Similar}_M(i) = (i_1, \dots, i_M).$$

В работе РС выделяют проблему «холодного старта» (англ. Cold start), в контексте которой рассматривается событие, в случае которого пока не накоплен приемлемый объем данных для корректного функционирования РС.

Проблема «холодного старта» актуальна для новых объектов или объектов, с которыми пользователи редко совершают действия. В случае если средний рейтинг будет вычислен по оценкам, например, пяти пользователей, то такая оценка не является достоверной. Довольно часто в подобных ситуациях такого рода рейтинги корректируют искусственно.

Существуют следующие возможные способы подобной коррекции рейтингов:

– расчет сглаженного среднего (англ. Damped mean) вместо среднего значения. Идея: рейтинг при малом количестве оценок стремится к среднему показателю, и после того как набирается необходимое количество новых оценок перестает действовать «усредняющая» корректировка;

– расчет среднего рейтинги для объекта, затем определяется доверительный интервал (англ. Confidence interval) данного рейтинга. Идея: чем больше оценок, тем меньше вариация среднего и, следовательно, больше уверенность в его адекватном значении. В качестве рейтинга объекта можно принять, например, нижнюю границу интервала (англ. Low CI bound). В данном случае такая система будет

достаточно консервативной и характеризоваться тенденцией к занижению оценок по новым объектам.

Аналитический обзор специальной и научной-технической литературы [1, 13, 29, 34, 50, 51 57, 69] показал, что на сегодняшний день существует несколько подходов к построению РС (рисунок 1.).

На рисунке 1 представлены 4 типа фильтрации, используемые для реализации РС, в соответствии с которыми выделяют 4 подхода к построению РС:

- фильтрация, основанная на контенте (Content base), или фильтрация содержимого,
- совместная, или коллаборативная (Collaboration) фильтрация,
- фильтрация, базирующаяся на знаниях (Knowledge base),
- гибридная (Hybrid) фильтрация.

Необходимо отметить, что наиболее распространены 3 подхода, поэтому рассмотрим именно эти подходы к построению современных РС.

Первый подход основан на алгоритме коллаборативной фильтрации – это когда не существует структурированных данных о контенте, но есть информация об отношении пользователя с некими объектами. Второй подход основан на структурированных данных о контенте, которые предоставляются самим пользователем. Этот подход называется рекомендацией на основе контента. Третий подход объединяет два предыдущих и поэтому получил название гибридным.

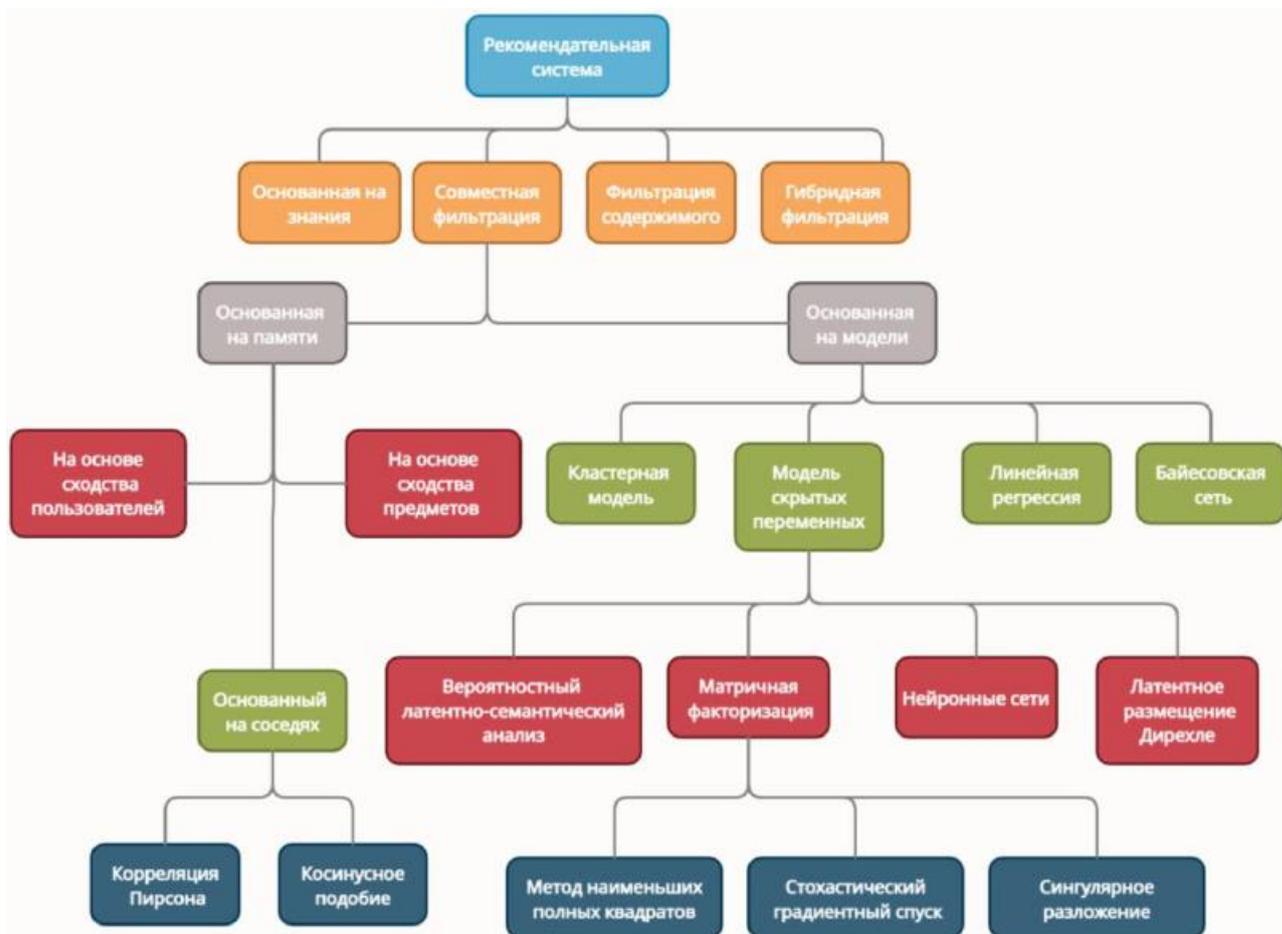


Рисунок 1 – Система методов и технологий к построению рекомендательных систем

Рекомендатор на основе коллаборативной фильтрации

Коллаборативная (или совместная) фильтрация – это когда алгоритмы рекомендуют элементы (это и есть часть фильтрации) на основе информации о предпочтениях от многих пользователей (а это совместная часть). Этот подход использует сходство поведения предпочтений пользователей, учитывая предыдущие взаимодействия между пользователями и элементами, алгоритмы рекомендаций учатся предсказывать будущее взаимодействие.

Подход коллаборативной фильтрации формирует модель с учетом «прошлого поведения» пользователя (например, ранее приобретенные или выбранные предметы и/или числовые оценки, присвоенные этим элементам), а также аналогичных решений, принятых другими пользователями.

Далее полученная модель будет использована с целью прогнозирования элементов (или рейтингов элементов).

Методы для РС, которые в основном основаны на предыдущих взаимодействиях между пользователями и целевыми элементами, известны как методы совместной фильтрации.

В результате все прошлые данные о взаимодействиях пользователя с целевыми объектами будут переданы в систему совместной фильтрации. Эта информация обычно записывается в виде матрицы, где строки представляют пользователей, а столбцы — элементы.

Идея: если некоторые люди принимали подобные решения и совершали покупки в прошлом, например, при выборе фильма, то существует высокая вероятность того, что они согласятся на дополнительные выборы в будущем. Например, если рекомендатель совместной фильтрации знает, что у вас и другого пользователя схожие вкусы в отношении фильмов,

Основная идея метода – выявить, что похожим пользователям нравятся похожие объекты.

Рассмотрим алгоритм реализации данного метода:

- Идентификация условной меры подобия (сходства) пользователей на основе их истории оценок $sim(u, v)$.
- Объединение пользователей в группы (кластеры) таким образом, чтобы похожие пользователи оказывались в одном кластере $u \in F(u)$.
- Предсказание оценки пользователя: средняя оценка кластера этому

$$\text{объекту } \bar{r}_{ui} = \frac{1}{|F(u)|} \sum_{u \in F(u)} r_{ui} .$$

Выделены проблемы [51], характерные для подхода к совместной фильтрации: «холодный» запуск, масштабируемость и разреженность:

- «Холодный старт». Для нового пользователя или элемента недостаточно данных, чтобы дать точные рекомендации;

– Масштабируемость. Возможность тиражирования и развертывания рекомендаций в других средах, которые дают миллионам пользователей;

– Разреженность. На самом деле количество товаров, продаваемых в крупных Internet-магазинах, чрезвычайно велико. Наиболее активные пользователи оценят сравнительно небольшую часть товаров из базы данных, в которой описаны все товары. Таким образом, даже самые популярные товары имеют достаточно мало оценок.

Кластеризация пользователей осуществляется в двух вариантах: user-based (сходства пользователей) и item-based (сходства элементов).

User-based алгоритм

Замена жесткой кластеризации на предположение, что объект понравится пользователю, если он понравился похожим пользователям. Тогда предпочтение пользователя u к объекту i можно записать по формуле (1) следующим образом

$$r_{ui} = r_u + \frac{\sum_{v \in U_i} sim(u, v)(r_{vi} - \bar{r}_v)}{\sum_{v \in U_i} sim(u, v)}, \quad (1)$$

где

– r_u – средняя оценка, поставленная пользователем u ,

– $sim(u, v)$ – мера схожести пользователей u и v .

Item-based алгоритм

Так называемый симметричный алгоритм – это когда объект понравится пользователю, если ему понравились похожие объекты. Предпочтение пользователя u к объекту i запишется по формуле (2) так:

$$\overline{r_{ui}} = \overline{r_i} + \frac{\sum_{j \in I_u} \text{sim}(i, j)(r_{uj} - \overline{r_j})}{\sum_{j \in I_u} \text{sim}(i, j)}, \quad (2)$$

где

- $\overline{r_i}$ – средняя оценка, проставленная объекту i ,
- $\text{sim}(i, j)$ – мера схожести пользователей i и j .

Методы совместной фильтрации делятся на две подгруппы: методы на основе памяти и методы на основе моделей.

Методы на основе памяти (memory-based methods) являются самыми основными, поскольку они вообще не используют модель. Они предполагают, что прогнозы могут быть сделаны исключительно на основе «памяти» прошлых данных, и обычно используют простой метод измерения расстояния, такой как ближайший сосед.

Подходы, основанные на моделях (memory-based methods), с другой стороны, обычно предполагают некоторую форму базовой модели и пытаются гарантировать, что любые сделанные прогнозы должным образом соответствуют модели.

Рекомендатор на основе контента

Контентная фильтрация – это один из популярных методов РС. Содержание или атрибуты вещей, которые нравятся пользователю, называются «контентом».

Методы контентной фильтрации базируются на описании элемента и профиле предпочтений пользователя. Эти методы подходят для тех случаев, когда известны данные об элементе (имя, местонахождение, описание и т. д.), но не о пользователе. РС на основе контента рассматривают рекомендацию как проблему классификации для конкретного пользователя и изучают так называемый классификатор на основе характеристик элемента предпочтений пользователя.

В такой РС ключевые слова используются для описания элементов, а профиль пользователя – для указания типа элемента, который нравится этому пользователю. Таким образом, алгоритмы РС стремятся рекомендовать товары, похожие на те, которые понравились пользователю в прошлом или которые он изучает в данный момент времени.

В частности, различные элементы-кандидаты сравниваются с элементами, предварительно оцененными пользователем, и рекомендуются наиболее подходящие элементы. Этот подход берет свое начало в исследованиях по поиску и фильтрации информации [51].

Здесь система использует особенности и лайки пользователей чтобы рекомендовать другому пользователю вещи, которые могут ему соответственно понравиться. Он использует информацию, предоставленную вами через Интернет, и ту информацию, которую они могут собрать, а затем составляет рекомендации в соответствии с этим.

Целью контентной фильтрации является классификация продуктов по определенным ключевым словам, узнать, что нравится покупателю, найти эти термины в базе данных, а затем порекомендовать похожие вещи.

Контентная фильтрация (рисунок 2) в качестве входных данных содержит множество пользователей и множество категорий (или ключевых слов), которыми были помечены как объекты интереса. Задача систем рекомендаций, основанных на контенте – вычислить множество объектов, которые наиболее близки к категориям, которыми интересуется текущий пользователь *u*.



Рисунок 2 – Контентная фильтрация

Этот тип РС зависит от входных данных, предоставленных пользователями, некоторые распространенные примеры включают Google, Wikipedia и т. д. Например, когда пользователь ищет группу ключевых слов, Google отображает все элементы, состоящие из этих ключевых слов.

Система рекомендаций на основе содержательного подхода работает по двум методам, каждый из которых использует разные модели и алгоритмы. Один использует метод векторного интервала и (vector spacing method) называется методом 1, а другой использует модель классификации – это метод 2 (с помощью которого можно, например, создать дерево решений).

Наиболее популярный алгоритм – это представление векторного пространства, или представление TF-IDF. В этом случае РС формирует профиль пользователей на базе контента с учетом взвешенного вектора характеристик элемента. Веса обозначают важность каждой функции для пользователя и вычисляются с учетом векторов контента и индивидуальной оценкой, используя различные методы.

Простые методы применяют средние значения вектора оцениваемых элементов, сложные методы базируются на основе методов ML для оценки вероятности того, какой именно элемент понравится пользователю, например, байесовский классификатор, кластерный анализ, деревья решений и искусственные нейронные сети [51].

РС решают задачи, базирующиеся на следующих классах моделей:

- offline-модели: направлены на выявление закономерностей на глобальном уровне (что зачастую понимается как коллаборативная фильтрация) с целью определить и сформировать рекомендации пользователю;
- online-модели: направлены на выявление закономерностей на данный период времени, что можно проинтерпретировать как «здесь и сейчас» с целью детерминации кратковременных трендов.

Online-модели отличаются от offline-моделей тем, что они менее персонализированы, поэтому offline-модели представляют собой класс персонализированных волнений, а online-модели направлены на поиск трендов, некоторых ниш в какой-либо сфере.

То, как рекомендательная модель дает рекомендации, будет зависеть от типа имеющихся данных. Если есть данные только о том, какие взаимодействия происходили в прошлом, то, вероятно, заинтересует совместная фильтрация. Если имеются данные, описывающие пользователя и объекты, с которыми он взаимодействовал (например, возраст пользователя, категория кухни ресторана, средний отзыв о фильме), можно смоделировать вероятность нового взаимодействия с учетом этих свойств в текущий момент путем добавления контента и контекстной фильтрации.

1.3 Алгоритмы и метрики точности рекомендаторов

1.3.1 Алгоритмы реализации систем рекомендаций

Аналитический обзор специальной и технической литературы позволил произвести отбор алгоритмов, которые используются в РС, основанных на подходе коллаборативной фильтрации: классификация, кластеризация, статистика и алгоритмы ML².

Двумя популярными подходами совместной фильтрации являются модели скрытых факторов, которые извлекают функции из матриц пользователей и элементов, и модели соседства, которые находят сходство между продуктами или пользователями.

Модель соседства – это предметно-ориентированный подход к выявлению пользовательских предпочтений на основе оценок, данных пользователем похожим предметам.

Модели скрытых факторов – это такие как разложение по сингулярным значениям (SVD), извлечь функции и корреляцию из матрицы пользовательских элементов.

Совместные методы для РС – это методы, которые основаны исключительно на прошлых взаимодействиях, записанных между пользователями и элементами, для выработки новых рекомендаций. Эти взаимодействия хранятся в так называемой «матрице взаимодействия пользователя с элементом». Так, много существующих алгоритмов используются для измерения сходства пользователей или сходства элементов в РС. Например, такая конфигурация как алгоритм k-ближайших соседей (Nearest Neighborhood, NN), или k-NN и в качестве метрики используется корреляция Пирсона или подобие косинусов или косинусное сходство [44, 45].

Алгоритм k-Nearest Neighborhood

² Алгоритмы ML анализируют данные, чтобы выявить закономерности и построить модель, которая затем используется для итоговых вычисления [45].

Стандартный метод совместной фильтрации известен как алгоритм ближайшего соседства (k-Nearest Neighborhood, k-NN). Существуют совместная фильтрация на основе пользователей и совместная фильтрация на основе элементов. Давайте сначала посмотрим на User-based совместной фильтрации. В этом случае имеется матрица оценок $n \times m$ с пользователем u_i , $i = 1, \dots, n$ и элементом p_j , $j = 1, \dots, m$. Требуется предсказать рейтинг r_{ij} , если целевой пользователь i не смотрел/оценивал элемент j . Процесс заключается в вычислении сходства между целевым пользователем i и всеми другими пользователями, выборе X самых популярных похожих пользователей и использовании в качестве весов средневзвешенных оценок этих X пользователей со сходством.

Рассмотрим пример совместной фильтрации на основе k-NN. Сводная матрица, в которой все значения являются рейтингами (рисунок 3).

ID пользователя	1	2	3	4
Название1	4	NaN	NaN	NaN
Название2	NaN	4	NaN	NaN
Название3	NaN	NaN	5	NaN
Название4	5	3	5	NaN

Рисунок 3 – Фрагмент матрицы, содержащей в качестве значений некие рейтинги

kNN – алгоритм ML, который находит кластеры похожих пользователей на основе общих рейтингов книг и делает прогнозы, опираясь на средний рейтинг лучших k ближайших соседей.

Используемая метрика в KnearestNeighbor – это подобие косинусов или косинусное сходство (Cosine Similarity). Данная техника будет использоваться, чтобы узнать оценку сходства и алгоритм вычислял косинусное сходство между векторами оценок по формуле (3).

$$sim(A,B)=cos(Theta)=\frac{A}{\|A\| \|B\|} \quad (3)$$

Далее будет использоваться kNN, работающий на евклидовом расстоянии. После расчета расстояния между векторами рейтинга, следует преобразовать значения (или рейтинги) матричного фрейма данных для проведения наиболее эффективных вычислений. Применение kNN позволит найти расстояние (или близость) экземпляров.

Затем kNN классифицирует экземпляр, находя ближайших соседей, и позволит определить наиболее близкий класс.

Алгоритм кластеризации k-means

Метод K-means (средних) – это один из методов кластеризации данных для разделения существующих данных на один или несколько кластеров или группы. Метод K-средних делит данные на кластеры, чтобы данные с одинаковыми характеристиками были сгруппированы в один и тот же кластер, а данные с разными характеристиками сгруппированы в другие группы. Цель кластеризации заключается в группировании объектов до тех пор, пока расстояние от каждого объекта до центра группы в группе не станет равным минимуму.

Процесс группировки с использованием K-средних обычно выполняется по следующему базовому алгоритму:

- Определить количество кластеров.
- Распределить данные по кластерам случайным образом.
- Рассчитать центроид (среднее значение) данных в каждом кластере.
- Распределите все данные по ближайшему центроиду.
- Вернуться к этапу 3, если все еще есть данные, которые перемещают кластеры, или если изменения в значении центроида выше заданное пороговое значение.

Для вычисления кластера центроидов используется следующая формула (4):

$$v_{ij} = \frac{\sum_{k=1}^{N_i} X_{kj}}{N_i}, \quad (4)$$

где

N_i – это объем данных, который является членом i -го кластера.

Одним из методов оценки, используемых в кластеризации К-средних, является метод коэффициента силуэта. Этот метод служит для проверки качества результирующего кластера и представляет собой метод проверки кластера, который сочетает в себе метод сцепления и разделения [3]. Сплоченность измеряет, насколько тесно связаны объекты в кластере. Разделение измеряет, насколько кластер отличается от других кластеров.

Для вычисления значения коэффициента силуэта требуется расстояние между объектами с помощью евклидовой формулы расстояния. После этого шаги для расчета значений коэффициента силуэта следующие:

1. Для каждого объекта i вычислить среднее расстояние от объекта i со всеми объектами в одном кластере. Среднее значение называется $a(i)$.

2. Для каждого объекта i вычислить среднее расстояние от объекта i до объекта в другом кластере. Из всех расстояния среднее принимает наименьшее значение. Это значение называется $b(i)$.

3. После этого для объектов вычисляется значение коэффициента силуэта по формуле (5):

$$s(i) = (b(i) - a(i)) \div \max(a(i), b(i)) \quad (5)$$

Результаты расчета значения коэффициента силуэта могут варьироваться от -1 до 1.

Результаты считаются хорошими, если значение коэффициента силуэта положительно ($a(i) < b(i)$) и $a(i)$ близко к 0, так что будет сгенерировано максимальное значение коэффициента силуэта, равное 1, когда $a(i) = 0$. Таким образом, можно сказать, что если $s(i) = 1$ означает, что объект i уже находится в правильном кластере. Если значение $s(i) = 0$, объект i находится между двумя кластерами, поэтому что объект не ясен, должен быть включен в кластер А или кластер В. Однако, если $s(i) = -1$ означает, что кластер сгенерированная структура перекрывается, поэтому объект i более точно включается в другой кластер.

Коэффициент силуэта каждого объекта в кластере является мерой, которая показывает, насколько плотно данные сгруппированы в кластере.

Ниже приведено значение силуэта, основанное на Кауфмане и Руссо.

$0.7 < SC \leq 1$	Сильная структура (Strong Structure)
$0.5 < SC \leq 0.7$	Средняя структура (Medium Structure)
$0.25 < SC \leq 0.5$	Слабая структура (Weak Structure)
$SC \leq 0.25$	Нет структуры (No Structure)

Разложение по сингулярным числам SVD

Когда дело доходит до уменьшения размерности, разложение по сингулярным значениям (SVD) является популярным методом линейной алгебры для матричной факторизации в ML.

SVD – это метод матричной факторизации, который уменьшает количество признаков в наборе данных за счет уменьшения размерности пространства с N-размерности до K-размерности (где $K < N$). SVD строит матрицу со строками пользователей и столбцами элементов, а элементы задаются рейтингами пользователей.

Разложение по сингулярным числам разлагает матрицу на три другие матрицы и извлекает факторы из факторизации матрицы высокого уровня (рейтинга пользователя) по формуле (5).

$$A = USV^T, \quad (6)$$

где

- матрица U: сингулярная матрица (пользователь*скрытые факторы),
- матрица S: диагональная матрица (показывает силу каждого скрытого фактора),
- матрица V: сингулярная матрица (элемент*скрытые факторы).

Из матричной факторизации скрытые факторы показывают характеристики предметов. Наконец, создается матрица полезности A формы m*n. Конечный результат матрицы A уменьшает размерность за счет извлечения скрытых факторов.

Из матрицы A он показывает отношения между пользователями и элементами, отображая пользователя и элемент в r-мерное скрытое пространство. Вектор X_i считается каждым элементом, а вектор Y_u считается каждым пользователем. Рейтинг выставляется пользователем по элементу как R_{ui} = X^T_i * Y_u. Потери можно свести к минимуму за счет разницы квадратов ошибок между произведением R_{ui} и ожидаемым рейтингом по формуле (6).

$$Min(x, y) \sum_{(u,i) \in K} (r_{ui} - x_i^T y_u)^2 \quad (7)$$

Регуляризация используется, чтобы избежать переобучения и обобщить набор данных, добавив штраф.

Рекомендатор на основе контента работает с данными, которые пользователь предоставляет явно (рейтинг) или неявно (нажатие на ссылку). На основе этих данных формируется профиль пользователя, который впоследствии используется для предоставления пользователю рекомендаций. По мере того, как пользователь вводит больше данных или выполняет действия в соответствии с рекомендациями, рекомендатор становится все более и более точным.

Аналитический обзор специальной и технической литературы позволил произвести отбор алгоритмов, которые используются в РС, основанных на содержательном подходе: метод векторного пространства или векторизация TF-IDF в совокупности с такими техниками обработки естественного языка (Natural Language Processing, NLP³) как извлечение стоп-слов, стемминг, лемминг – которые позволяют исключить из вектора несущественную и нерелевантную информацию.

Понятия частоты терминов (TF, Term Frequency) и обратной частоты документов (IDF, Inverse Document Frequency) используются в поиске информации, а также в механизмах фильтрации на основе контента (таких как рекомендатор на основе контента). Они используются для определения относительной важности документа/статьи/новости/фильма и т. д.

Алгоритм TF*IDF используется для взвешивания ключевого слова в документе и присвоения важности ключевому слову в зависимости от того, сколько раз оно встречается в документе. Проще говоря, чем выше показатель TF*IDF (вес), тем реже и важнее термин, и наоборот.

Каждое слово или термин имеет свою соответствующую оценку TF и IDF. Произведение оценок TF и IDF термина называется весом TF*IDF этого термина.

³ NLP – это ветвь ИИ, которая позволяет машинам интерпретировать человеческий, т. е. естественный язык. Однако то же самое не может быть использовано машиной напрямую, и нужно сначала предварительно обработать то же самое.

TF слова – это количество раз, которое оно встречается в документе. Зная это, вы сможете увидеть, используете ли вы термин слишком часто или слишком редко.

$TF(t) = (\text{Количество раз, когда термин } t \text{ появляется в документе}) / (\text{Общее количество терминов в документе})$.

IDF слова – это мера того, насколько значим этот термин во всем корпусе.

$IDF(t) = \log_e (\text{Общее количество документов} / \text{Количество документов с термином } t \text{ в нем})$.

Стемминг и лемматизация – это методы, используемые поисковыми системами и чат-ботами для анализа значения слова. Стемминг использует основу слова, а лемматизация использует контекст, в котором используется слово.

Цель обоих процессов одна и та же: свести флективные формы каждого слова к общей основе или корню. Однако эти два метода не совсем одинаковы. Основное различие заключается в том, как они работают, и, следовательно, в результате, который возвращает каждый из них. Еще одно важное отличие, которое следует подчеркнуть, заключается в том, что лемма является базовой формой всех своих флективных форм, а основа – нет. Вот почему обычные словари представляют собой списки лемм, а не основы.

Алгоритмы стемминга работают, отсекая конец или начало слова, принимая во внимание список общих префиксов и суффиксов, которые можно найти во флективном слове. Это неизбирательное сокращение может быть успешным в некоторых случаях, но не всегда, поэтому мы утверждаем, что этот подход имеет некоторые ограничения. Ниже проиллюстрирован этот метод примерами на английском языке.

Лемматизация, с другой стороны, принимает во внимание морфологический анализ слов. Для этого необходимо иметь подробные словари, которые алгоритм может просмотреть, чтобы связать форму со своей леммой.

Удаление стоп-слов – это один из наиболее часто используемых этапов предварительной обработки в различных приложениях NLP. Идея состоит в том, чтобы просто удалить слова, которые часто встречаются во всех документах корпуса. Как правило, артикли и местоимения обычно классифицируются как стоп-слова.

Эти слова не имеют значения в некоторых задачах NLP, таких как поиск и классификация информации, что означает, что эти слова не очень различительны. Наоборот, в некоторых приложениях NLP удаление стоп-слов будет иметь очень небольшое влияние. В большинстве случаев список стоп-слов для данного языка представляет собой хорошо составленный вручную список слов, наиболее часто встречающихся в корпусах.

1.3.2 Метрики

Системы рекомендаций в ML – это один из таких алгоритмов, который работает на основе схожести содержимого.

Рассмотрим наиболее часто используемые метрики для вычисления сходства между векторами.

Евклидово расстояние

Метод Евклидова расстояние используется для расчета расстояния между данными и центроидом. Это измерение на основе значения объекта в каждом измерении в обучении. Евклидово расстояние может рассчитать расстояние между данными до двух измерений и более. Например, формула (7) Евклидова расстояния с 2 объектами и 2 измерениями

$$D_{pq} = \sqrt{(p_2 - p_1)^2 + (q_2 - q_1)^2}, \quad (8)$$

где

- D_{pq} = расстояние двух объектов,
- X_1 = первое измерение первого объекта,
- X_2 = первое измерение второго объекта.

Математически для n-мерного пространства и (p_i, q_i) в качестве точек данных метрика идеального расстояния рассчитывается по формуле (8):

$$D_e = \sum_{i=1}^n (p_i - q_i)^2 \quad \frac{1}{2} \quad (9)$$

Евклидово расстояние используется во многих алгоритмах ML в качестве метрики расстояния по умолчанию для измерения сходства между двумя записанными наблюдениями. Однако сравниваемые наблюдения должны включать непрерывные функции и числовые переменные, такие как вес, рост, зарплата и т. д.

Косинусное расстояние

Косинусное сходство – это мера сходства между двумя точками данных (векторами) на плоскости. Косинусное сходство (9) используется в качестве метрики в различных алгоритмах ML, таких как KNN, для определения расстояния между соседями, в РС, оно используется для рекомендации, например, пользователей с одинаковым предпочтением и для текстовых данных, оно используется для поиска сходства тексты в документе.

$$\text{similarity}(A, B) = \frac{A \cdot B}{\|A\| \|B\|} = \frac{\sum_{i=1}^n A_i B_i}{\sqrt{\sum_{i=1}^n A_i^2} \sqrt{\sum_{i=1}^n B_i^2}} \quad (10)$$

РС в основном используют матрицу подобия, чтобы рекомендовать похожий контент пользователю на основе его характеристик доступа. Таким образом, любые рекомендательные данные могут быть получены, а необходимые функции, которые могут быть полезны для рекомендации

содержания, могут быть извлечены из данных. Как только необходимые текстовые данные будут доступны, текстовые данные должны быть векторизованы с использованием CountVectorizer для получения матрицы подобия (рисунок 4). Таким образом, как только матрица подобия получена, метрики косинусного сходства Scikit-learn можно использовать для рекомендации пользователю.

```
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.metrics.pairwise import cosine_similarity
count_vec=CountVectorizer()
sim_matrix=count_vec.fit_transform(df['text_data'])
print('Similarity Matrix',sim_matrix.toarray())
cos_sim = cosine_similarity(sim_matrix)
```

Рисунок 4 – Пример векторизации с использованием CountVectorizer

Корреляция Пирсона или корреляционное сходство

Коэффициент корреляции Пирсона представлен как «r», он измеряет, насколько сильна линейная связь между двумя непрерывными переменными.

Существуют различные типы формул для получения коэффициента корреляции, одной из самых популярных является корреляция Пирсона (также известная как R Пирсона), которая обычно используется для линейной регрессии.

Коэффициент корреляции Пирсона обозначается символом «r». Формула коэффициента корреляции (10) возвращает значение от 1 до -1.

Здесь

- 1 указывает на сильную отрицательную связь,
- 1 указывает на сильные положительные отношения,
- и нулевой результат указывает на отсутствие отношений вообще.

$$r = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum (x_i - \bar{x})^2 \sum (y_i - \bar{y})^2}} \quad (11)$$

Принимая во внимание данные метрики, можно построить РС в соответствии с концепцией их работы.

Итоги по главе 1

Подводя итоги по первой главе, следует констатировать, что на данный момент особо актуально в системе образования во всем мире решение проблемы проектирования персонального образовательного трека обучаемого, в том числе учебного контента с учетом его познавательных потребностей, что, в свою очередь, является важной частью современного адаптивного обучения, которое определяет перспективные образовательные технологии и системы в условиях глобальной цифровой экономики [32]. Таким образом, РС являются технологической составляющей в формировании персонального учебного контента обучаемого по дисциплине.

В этой связи, в настоящей работе предлагается разработка прототипа РС, который опираясь на данные о результатах итогового тестирования обучаемых по некоторой дисциплине, может использоваться в формировании персонального учебного контента для обучаемого. Также, такая система сможет рекомендовать существующие MOOK, основываясь на соответствующих исходных учебных данных студента по дисциплине.

Проведен обзор подходов к построению РС, а так же соответствующих алгоритмов ML, которые используются при построении РС, показаны способы оценки качества моделей ML.

Глава 2 Практическая реализация

2.1 Выбор языка программирования и цифровой среды для разработки алгоритмического обеспечения рекомендатора персонального учебного контента

Python – язык программирования высокого уровня общего назначения, который широко используется для программной реализации большинства технологий ИИ [17, 18, 20, 28, 38, 43, 45, 48].

Рассмотрим список причин (рисунок 5), по которым Python является предпочтительным языком программирования как для каждого основного разработчика, специалиста по данным, инженера по ML, так и реализации задач настоящей работы [4, 7, 27, 30, 41]:

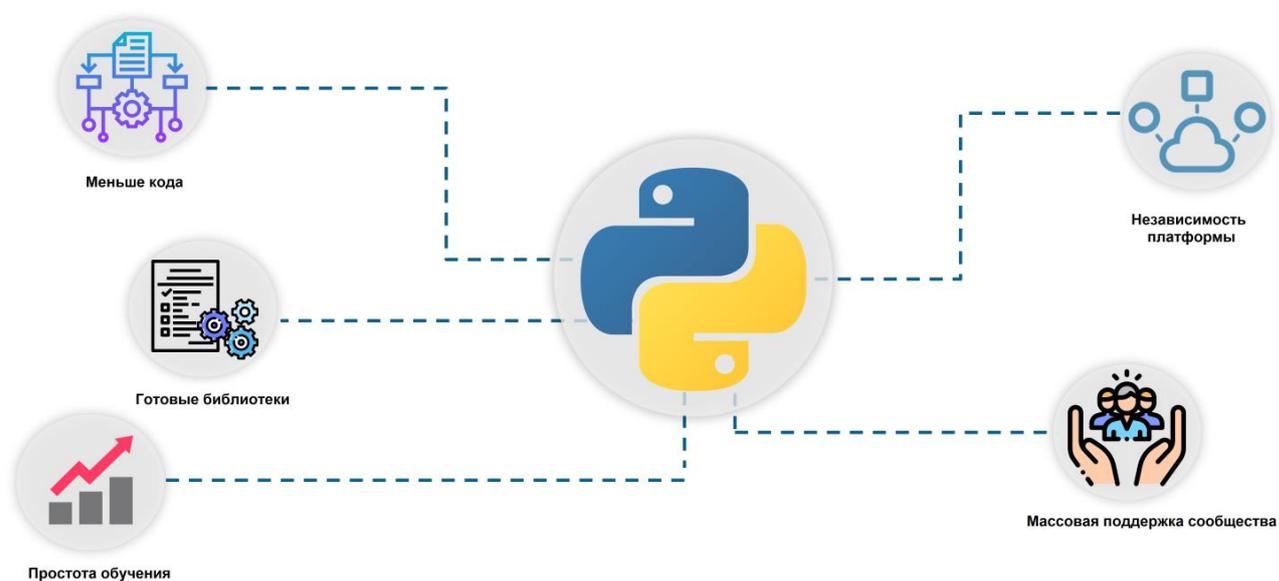


Рисунок 5 – Преимущества языка программирования Python для ИИ, в частности для ML

Мало программного кода.

Для реализации технологий ИИ требуется множество алгоритмов. Благодаря Python и наличию определенного многообразия пакетов (фреймворков)

нет необходимости программировать алгоритмы. Более того, Python поддерживает методологию «проверки в процессе написания программного кода», которая снижает нагрузку на тестирование программного кода.

Готовые библиотеки.

Python имеет сотни библиотек для реализации различных алгоритмов ML, в том числе и глубокого обучения. Каждый раз, когда требуется запустить алгоритм на наборе данных, все, что нужно сделать – это установить и загрузить нужные библиотеки. Например: NumPy, Keras, Tensorflow, Pytorch и т. д.

Простота обучения.

Python использует несложный синтаксис, который достаточно эффективно используется для реализации простых вычислений.

Независимость от платформы.

Python может работать на нескольких платформах, включая ряд сред таких операционных систем как Windows, MacOS, Linux, Unix и т. д. В случае переноса программного кода с одной платформы на другую можно воспользоваться таким пакетом, как PyInstaller, который устранит проблемы с зависимостями.

Массовая поддержка сообщества.

В качестве цифровой среды разработки алгоритмического обеспечения рекомендатора персонального учебного контента был выбран Google Colaboratory – простая в настройке интерактивная цифровая среда для запуска кода ML, который поставляется с бесплатным доступом к графическим процессорам. Colaboratory (или сокращенно Colab) – это продукт Google Research. Colab позволяет любому писать и выполнять произвольный код Python через браузер.

Технически, Colab представляет собой цифровой сервис ноутбуков Jupyter⁴, размещенный на хостинге и при этом не требующий настройки для использования, так же он предоставляет бесплатный доступ к вычислительным ресурсам, включая GPU.

Для предварительного анализа и подготовки данных так же был использован Google Colaboratory, который представляет собой бесплатный цифровой облачный сервис на основе Jupyter Notebook для МО и анализа данных. Непосредственно работа с датасетами выполнялась с использованием библиотек языка программирования Python: Pandas и Numpy .

Рассмотрим функциональную модель прототипа рекомендатора персонального учебного контента (рисунок 6).



Рисунок 6 – Контекстная диаграмма IDEF0 автоматизированного бизнес-процесса формирования персонального учебного контента обучаемого

На рисунке 7 представлена декомпозиция автоматизированного бизнес-процесса, который показывает основные технологические этапы

⁴ Project Jupyter — это бесплатное программное обеспечение для интерактивных вычислений на нескольких языках программирования.

функционирования прототипа рекомендатора персонального учебного контента обучаемого по дисциплине.

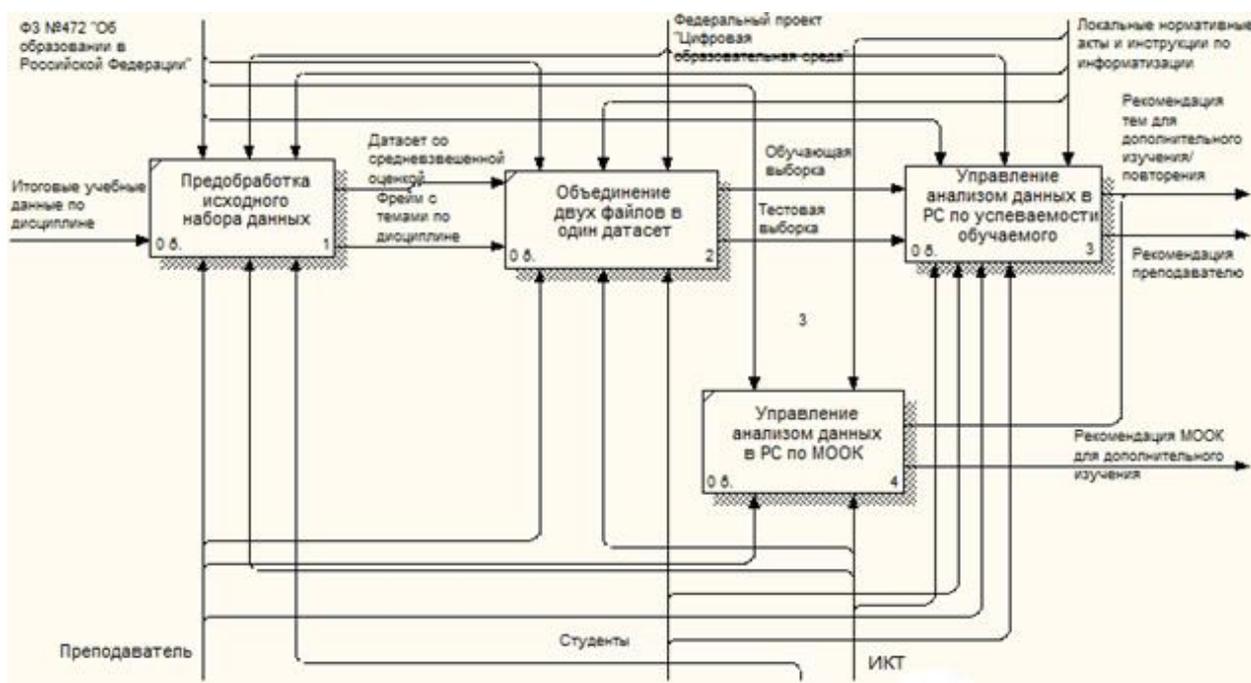


Рисунок 7 – Декомпозиция диаграммы IDEF0 автоматизированного бизнес-процесса формирования индивидуального учебного контента обучаемого

Согласно функциональной модели IDEF0 бизнес-процесс разработка прототипа рекомендатора для создания персонального контента обучаемого инициирована федеральными нормативными документами [52, 53], которые регламентируют цифровую трансформацию национальной сферы образования и внедрение разнообразных инструментов и инновационных техник реализации электронного обучения в комплексе с традиционными методами обучения, например, применение методов и средств ИИ для формирования учебного контента. Управление цифровизацией образовательного процесса в университете регламентируется соответствующими локальными нормативными актами.

На основе анализа входных данных – итоговые учебные данные по дисциплине – предлагаемый рекомендатор формирует персональный учебный

контент обучаемого, принимая во внимание сформированные системой рекомендации тем для дополнительного повторения и какие MOOK можно дополнительно изучить для повышения уровня освоения дисциплины.

2.2 Реализация алгоритма рекомендательной системы для изучения дисциплины

Рассмотрим РС с учетом данных о тестировании студентов по дисциплине «Цифровая грамотность». В настоящем исследовании для программной реализации обозначенной выше РС, за основу был взят и адаптирован под задачи исследования датасет, который находится в свободном доступе на специализированном Web-сайте Kaggle⁵ (<https://www.kaggle.com/johnweek/students-test-results>), который содержит результаты тестирования некоторых студентов.

В контексте этой работы данные датасета интерпретируются следующим образом по разнообразным темам дисциплины «Цифровая грамотность»:

- с ‘Unnamed: 0’ по ‘Unnamed: 11’ – Компьютерная грамотность,
- с ‘Unnamed: 0’ по ‘Unnamed: 3’ – Информация,
- с ‘Unnamed: 4’ по ‘Unnamed: 11’ – Алгебра логики,
- с ‘Unnamed: 12’ по ‘Unnamed: 34’ – Техническая грамотность,
- с ‘Unnamed: 12’ по ‘Unnamed: 13’ – Текстовый процессор,
- с ‘Unnamed: 14’ по ‘Unnamed: 22’ – Электронные таблицы,
- с ‘Unnamed: 23’ по ‘Unnamed: 32’ – Базы данных,
- с ‘Unnamed: 33’ по ‘Unnamed: 34’ – Мультимедиа,
- с ‘Unnamed: 35’ по ‘Unnamed: 71’ – Программирование,

⁵ Kaggle – открытая для всех англоязычная Web-платформа с целью организации международных конкурсов по исследованию данных. В эту цифровую платформу интегрирована социальная сеть специалистов международного уровня по обработке данных и машинному обучению. Kaggle принадлежит корпорации Google (с марта 2017 г.)

- с ‘Unnamed: 35’ по ‘Unnamed: 40’ – Линейные вычисления,
- с ‘Unnamed: 41’ по ‘Unnamed: 47’ – Ветвление,
- с ‘Unnamed: 48’ по ‘Unnamed: 59’ – Циклы,
- с ‘Unnamed: 60’ по ‘Unnamed: 65’ – Массивы,
- с ‘Unnamed: 66’ по ‘Unnamed: 71’ – Функции Модули,
- с ‘Unnamed: 72’ по ‘Unnamed: 87’ – Кибербезопасность,
- с ‘Unnamed: 88’ по ‘Unnamed: 99’ – Интернет грамотность.

Датасет testdata.csv включает 1000 записей о студентах и их результативных ответов на 100 вопросов итогового теста. Так датасет testdata.csv содержит информацию о правильности (1), неправильности (0) или отсутствия ответа (NaN) на тестовое задание обучаемым. Под отсутствием ответа в значении NaN будем понимать то, что студент не успел решить тестовое задание.

Ниже показан фрагмент программного кода на Python для обработки датасета, а также показан фрагмент выборки этого набора данных (рисунок 8).

```
# Display top of dataframe
data.head()
```

	Unnamed: 0	question 0	question 1	question 2	question 3	question 4	question 5	question 6	question 7	question 8	...	question 90	question 91	question 92	question 93	question 94	question 95	question 96	question 97	question 98	question 99
0	Charlene Jackson	1.0	NaN	NaN	NaN	NaN	0.0	0.0	1.0	0.0	...	0.0	0.0	0.0	NaN	0.0	NaN	NaN	1.0	0.0	1.0
1	Eric Maldonado	NaN	1.0	NaN	NaN	NaN	0.0	0.0	1.0	1.0	...	1.0	1.0	NaN	0.0	0.0	NaN	NaN	0.0	1.0	1.0
2	Samantha Mendoza	1.0	NaN	1.0	0.0	0.0	NaN	1.0	1.0	0.0	...	NaN	1.0	NaN	NaN	NaN	0.0	NaN	1.0	NaN	NaN
3	Linda Francis	0.0	0.0	0.0	0.0	1.0	0.0	0.0	1.0	1.0	...	NaN	NaN	1.0	1.0	1.0	1.0	1.0	1.0	1.0	NaN
4	Ana Peterson	1.0	0.0	1.0	1.0	0.0	1.0	0.0	0.0	0.0	...	NaN	1.0	0.0	1.0	0.0	NaN	0.0	0.0	NaN	1.0

Рисунок 8 – Фрагмент выборки набора данных testdata.csv

Данные представлены в виде датафрейма,

где

- индексы – это фамилии студентов,
- названия столбцов (признаков), т. е. это идентификаторы тестовых заданий.

Для описания последовательности выполняемых действий разрабатываемой РС по успеваемости обучаемого была построена UML диаграмма активности (рисунок 9), раскрывающая детали алгоритмической реализации операций, выполняемых этой РС.

Функционал РС по успеваемости обучаемого:

- Формирование перечня тем по каждому обучаемому с процентом освоения.
- Формирование перечня тем дисциплины, которые необходимо изучить.
- Прогнозирование паттерна успешного освоения соответствующих тем/ разделов/ модулей для освоения дисциплины
- Прогнозирование паттерна разделов или модулей или тем дисциплины, которые вызывают затруднения у большинства обучаемых

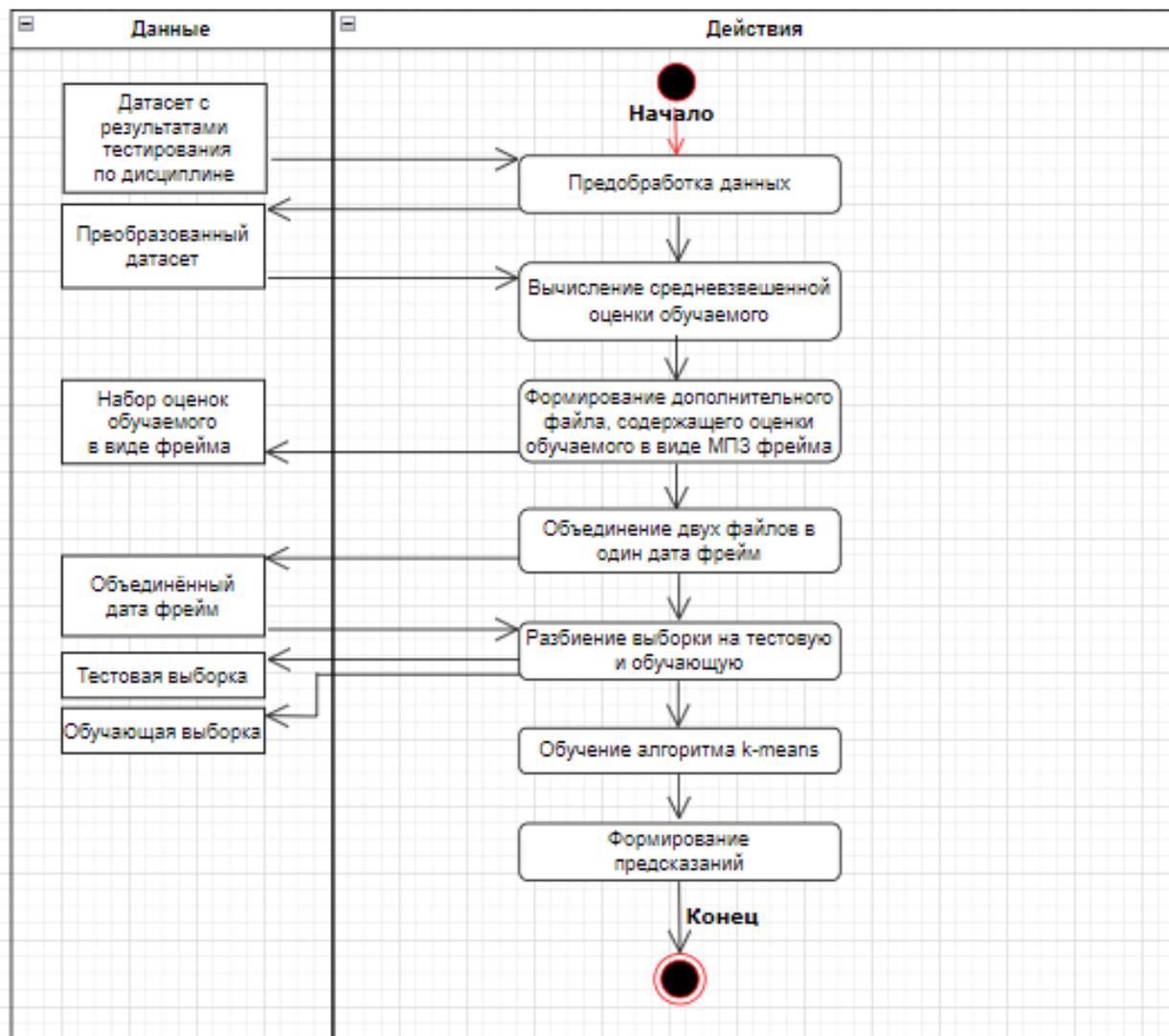


Рисунок 9 – UML диаграмма активности: РС по успеваемости обучаемого

Алгоритмическое обеспечение этой РС включает блок аналитики для реализации первой и второй функции РС по успеваемости обучаемого, а также алгоритм kNN.

Скрипт на Python обрабатывает датасет с результатами тестирования и словарь с темами дисциплины. При этом есть информация о том, какие номера тестовых заданий соответствуют темам дисциплины «Цифровая грамотность» (рисунок 10).

```

#Создаем массив с разделами и темами
#subjects[0][0] = начальный столбец раздела
#subjects[0][1] = конечный столбец раздела
#subjects[0][2] = название раздела
#subjects[0][3] = подтемы, если они есть
list_subjects = [
    [0, 11, 'Компьютерная грамотность',
     [[0, 3, 'Информатика'], [4, 11, 'Алгебра логики']],
    ],
    [12, 34, 'Техническая грамотность',
     [[12, 13, 'Текстовый процессор'], [14, 22, 'Электронные таблицы'], [23, 32, 'Базы данных'], [33, 34, 'Мультимедиа']],
    ],
    [35, 71, 'Программирование',
     [[35, 40, 'Линейные вычисления'], [41, 47, 'Ветвление'], [48, 59, 'Циклы'], [60, 65, 'Массивы'], [66, 71, 'Функции Модули']],
    ],
    [72, 87, 'Кибербезопасность',
     [],
    ],
    [88, 99, 'Интернет грамотность',
     [],
    ]
]
1

```

Рисунок 10 – Фрагмент кода соотнесения номеров вопросов темам дисциплины

На основе датасета на Python создан словарь по каждому обучаемому, в котором в качестве ключей выступают темы тестовых заданий, а значениями – список list, содержащий сумму набранных баллов по каждому вопросу, который затрагивает ту или иную область знаний дисциплины (рисунок 11).

```

#Используя информацию о темах, записываем оценки по данным из list_input_data в list_output_data
i = 0
#Вычисляем оценки каждого студента
for input_student in list_input_data:
    #Для каждого предмета
    for subject in list_subjects:
        #Вычисляем оценку за предмет
        start_question = subject[0]
        end_question = subject[1]+1
        subject_name = subject[2]
        sum_rating = 0.0;
        questions_in_theme = end_question - start_question
        #Суммируем единицы в теме
        for question_number in range(start_question, end_question):
            question_rating = input_student['question ' + str(question_number)]
            if question_rating != '':
                sum_rating += float(question_rating)
        #Оценка за предмет
        subject_rating = sum_rating*100 / questions_in_theme
        #Создаем в выходном словаре с i-м студентом текущий предмет
        list_output_data[i][subject_name] = round(subject_rating, 2)
        #Для каждой подтемы
        for theme in subject[3]:

```

Рисунок 11 – Фрагмент кода вычисления оценки студента

Затем, принимая во внимание вес каждого балла, соответствующего какой-то теме дисциплины, будет вычислена средневзвешенная⁶ оценка по формуле (11):

$$U_{av} = (U_1 \cdot W_1 + O_2 W_2 + K + U_N \cdot W_N) \div (U_1 + W_2 + K + W_N), \quad (11)$$

где

– U_N – это оценка по тестовому заданию,

– W_N – это вес оценки.

–

Создадим функцию (рисунок 12), которая по фамилии студента выводит:

– список тем и разделов, которые нужно повторить (результат от 50 до 74 включительно),

– список тем и разделов, которые нужно изучить (результат меньше 50).

⁶ Средневзвешенное – это среднее с учетом различной ценности (веса, важности) каждого из элементов.

```

def showBadSubjects(family):
    #Ищем нужного студента
    studentRow = 0
    while studentRow < len(list_output_data)-1 and list_output_data[studentRow]['Фамилия'] != family:
        studentRow += 1
    #Если студент не найден, выводим ошибку
    if list_output_data[studentRow]['Фамилия'] != family:
        print('Студент с фамилией ' + family + ' не найден.')
    else:
        #Иначе выводим список для повтора
        themeFound = False
        print("\nСписок тем для повторения 50 - 74:")
        #Выводим разделы для повторения
        for subject in list_subjects:
            subject_name = subject[2]
            subject_rating = list_output_data[studentRow][subject_name]
            #Если оценка по разделу от 50 до 74, выводим раздел
            if subject_rating >= 50 and subject_rating <= 74:
                print("\n\tРаздел: " + subject_name + ": " + str(subject_rating))
                themeFound = True
            #Выводим темы для повторения
            for theme in subject[3]:
                theme_name = theme[2]
                theme_rating = list_output_data[studentRow][theme_name]
                #Если оценка за тему от 50 до 74, выводим тему
                if theme_rating >= 50 and theme_rating <= 74:
                    print("\t\tПредмет: " + theme_name + ": " + str(theme_rating))
                    themeFound = True
        #Если ничего не найдено, выводим сообщение
        if themeFound == False:
            print("Темы не найдены\n")
        #Выводим список тем для изучения
        themeFound = False
        print("\nСписок тем для изучения 0 - 60:")
        #Выводим разделы для изучения
        for subject in list_subjects:
            subject_name = subject[2]
            subject_rating = list_output_data[studentRow][subject_name]

```

Рисунок 12 – Фрагмент функции поиска по фамилии студента результатов тестирования студента

РС может выдать рекомендации следующего вида (рисунок 13, а, б).

```
#Используем функцию для вывода списка тем для изучения и повтора по фамилии Jackson
showBadSubjects('Jackson')
```

Список тем для повторения 50 - 74:
Предмет: Текстовый процессор: 50.0
Предмет: Ветвление: 57.14

Список тем для изучения 0 - 50:

Раздел: Компьютерная грамотность: 25.0
Предмет: Информатика: 25.0
Предмет: Алгебра логики: 25.0

Раздел: Техническая грамотность: 30.43
Предмет: Электронные таблицы: 22.22
Предмет: Базы данных: 40.0
Предмет: Мультимедиа: 0.0

Раздел: Программирование: 32.43
Предмет: Линейные вычисления: 16.67
Предмет: Циклы: 33.33
Предмет: Массивы: 33.33
Предмет: Функции Модули: 16.67

Раздел: Кибербезопасность: 37.5

Раздел: Интернет грамотность: 33.33

a)

```
#Используем функцию для вывода списка тем для изучения и повтора по фамилии Wang
showBadSubjects('Wang')
```

Список тем для повторения 50 - 74:

Раздел: Компьютерная грамотность: 50.0
Предмет: Базы данных: 60.0
Предмет: Функции Модули: 50.0

Раздел: Кибербезопасность: 56.25

Список тем для изучения 0 - 50:

Предмет: Алгебра логики: 37.5

Раздел: Техническая грамотность: 30.43
Предмет: Текстовый процессор: 0.0
Предмет: Электронные таблицы: 11.11
Предмет: Мультимедиа: 0.0

Раздел: Программирование: 37.84
Предмет: Линейные вычисления: 33.33
Предмет: Ветвление: 42.86
Предмет: Циклы: 33.33
Предмет: Массивы: 33.33

Раздел: Интернет грамотность: 33.33

б)

Рисунок 13 – Примеры формирования РС рекомендации

Темы дисциплины, которые на основе имеющихся результатов тестирования, имеют не высокие оценки, будут рекомендованы студентам.

Полученные данные с результатами тестирования подходят для решения задачи классификации на основе такого алгоритма МО с учителем как алгоритм k-ближайших соседей (рисунок 14). Для идентификации группы обучаемых со схожим уровнем знаний по какой-либо теме дисциплины применяется алгоритм k-NN из библиотеки Scikit-Learn. Для того, чтобы воспользоваться этим алгоритмом необходимо подключить и использовать библиотеку Scikit-Learn.

```
neigh = neighbors.NearestNeighbors(n_neighbors=5)
neigh.fit(people)
neighs = neigh.kneighbors(men, return_distance=False)
```

Рисунок – 14 Фрагмент программной реализации алгоритма k-NN

Подводя итоги, необходимо отметить, что данную РС целесообразно использовать в итоговом контроле знаний и компетенций студентов по обозначенной дисциплине. В перспективе разработанная РС может быть интегрирована в информационно-образовательную среду университета, например, система дистанционного обучения (СДО) Moodle, СДО «Прометей» или другие СДО, а так же, непосредственно, в электронный учебно-методический комплекс дисциплины.

2.3 Реализация алгоритма рекомендательной системы по выбору массового онлайн-курса

Для описания последовательности выполняемых действий разрабатываемой РС по выбору MOOK была построена UML диаграмма активности (рисунок 15), раскрывающая детали алгоритмической реализации операций, выполняемых этой РС.

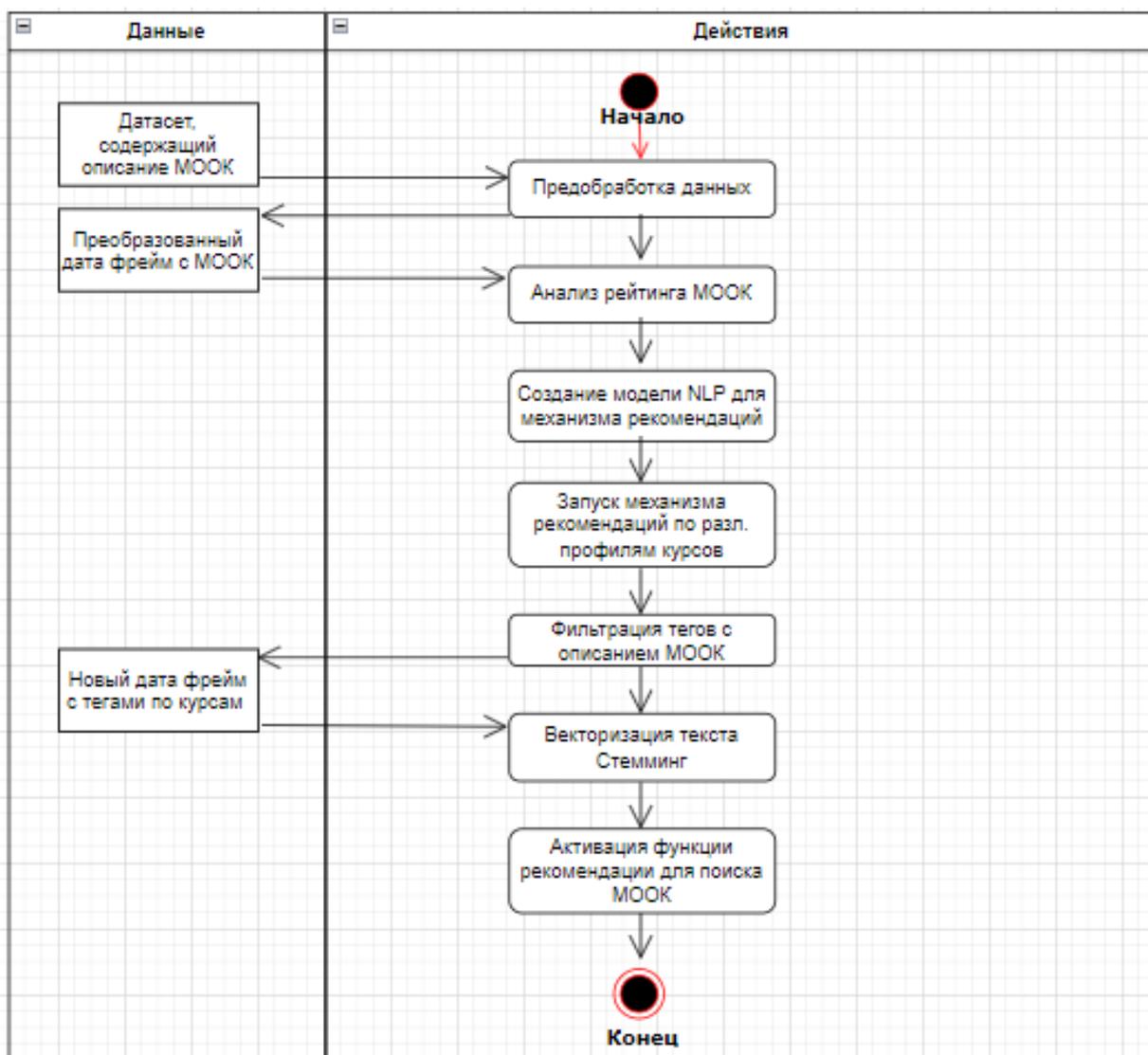


Рисунок 15 – UML диаграмма активности: РС по выбору MOOK

Для предварительного анализа и подготовки данных также был использован Google Colaboratory. Непосредственно работа с датасетами выполнялась с использованием библиотек языка программирования Python: Pandas и NumPy.

Ниже показан фрагмент программного кода на Python для обработки датасета, а также показан фрагмент выборки этого набора данных (рисунок 16).

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

```
data = pd.read_csv("Coursera.csv")
data.head()
```

	Course Name	University	Difficulty Level	Course Rating	Course URL	Course Description	Skills
0	Write A Feature Length Screenplay For Film Or ...	Michigan State University	Beginner	4.8	https://www.coursera.org/learn/write-a-feature...	Write a Full Length Feature Film Script In th...	Drama Comedy peering screenwriting film D...
1	Business Strategy: Business Model Canvas Analy...	Coursera Project Network	Beginner	4.8	https://www.coursera.org/learn/canvas-analysis...	By the end of this guided project, you will be...	Finance business plan persona (user experien...
2	Silicon Thin Film Solar Cells	cole Polytechnique	Advanced	4.1	https://www.coursera.org/learn/silicon-thin-fi...	This course consists of a general presentation...	chemistry physics Solar Energy film lambda...

Рисунок 16 – Фрагмент набора данных Coursera.csv

Структура набора данных coursera.csv содержит 7 признаков (см. рис. 16):

- CourseName – наименование курса,
- University – университет, разработавший этот курс,
- Difficulty Level – уровень сложности,
- Course Rating – рейтинг курса,
- Course URL – URL-адрес курса,
- Course Description – описание курса,
- Skills – умения, которые можно развить/ сформировать в этом курсе.

На данный момент осуществляется обработка данных этого датасета и подбор алгоритма обучения для построения РС.

Построение РС целесообразно описать двумя основными шагами:

Первый. Построение базы пар или групп схожих объектов

Второй. Поиск по этой базе данных, похожих на искомые

В рамках данной работы наша задача состоит в построении и реализации алгоритма фильтрация сходства на основе контента, базирующегося на тегах MOOK, которые пользователи ищут. Используемый набор данных относится к набору данных курсов Coursera, который содержит более 3000 курсов.

На рисунках 17 и 18 представлены фрагменты кода на Python, реализующие предобработку данных (векторизация текста и стемминг) и выбор меры подобия (рисунок 19)

```
from sklearn.feature_extraction.text import CountVectorizer  
  
cv = CountVectorizer(max_features=5000, stop_words='english')  
  
vectors = cv.fit_transform(new_df['tags']).toarray()
```

Рисунок 17 – Векторизация текста

```
#for stemming process  
import nltk  
  
from nltk.stem.porter import PorterStemmer  
ps = PorterStemmer()
```

Рисунок 18 – Стемминг

```
from sklearn.metrics.pairwise import cosine_similarity  
  
similarity = cosine_similarity(vectors)
```

Рисунок 19 – Мера подобия (Similarity Measure)

На рисунках 20 и 21 представлена функция формирования рекомендации перечня MOOK на основе ввода пользователем ключевого

термина необходимой темы дисциплины, по которой требуется дополнительно позаниматься, чтобы повысить уровень освоения дисциплины.

```
def recommend(course):  
    course_index = new_df[new_df['course_name'] == course].index[0]  
    distances = similarity[course_index]  
    course_list = sorted(list(enumerate(distances)),reverse=True, key=lambda x:x[1])[1:7]  
  
    for i in course_list:  
        print(new_df.iloc[i[0]].course_name)
```

```
recommend('Parallel programming')
```

Parallel Programming in Java
Big Data Analysis with Scala and Spark
Functional Programming in Scala Capstone
Functional Program Design in Scala
Crash Course on Python
Functional Programming Principles in Scala

Рисунок 20 – Пример 1 вызова функции формирования рекомендации MOOK

```
def recommend(course):  
    course_index = new_df[new_df['course_name'] == course].index[0]  
    distances = similarity[course_index]  
    course_list = sorted(list(enumerate(distances)),reverse=True, key=lambda x:x[1])[1:7]  
  
    for i in course_list:  
        print(new_df.iloc[i[0]].course_name)
```

```
recommend('Software Security')
```

Web Application Security Testing with Burp Suite
Protecting Cloud Architecture with Alibaba Cloud
Enterprise and Infrastructure Security
Software Design as an Element of the Software Development Lifecycle
Cybersecurity Capstone Project
Information Security Context and Introduction

Рисунок 21 – Пример 2 вызова функции формирования рекомендации MOOK

Для проверки точности поиска списка MOOK в РС [50, 51] используется метрика косинусного подобия по формуле (12):

$$\begin{aligned} \text{cosine similarity} = S_c(A, B) &:= \cos(\theta) = \frac{A \cdot B}{\|A\| \|B\|} = \\ &= \frac{\sum_{i=1}^n A_i B_i}{\sqrt{\sum_{i=1}^n A_i^2} \sqrt{\sum_{i=1}^n B_i^2}} \end{aligned} \quad (12)$$

Для того, чтобы вычислить метрику косинусного подобия, необходимо импортировали функцию подобия косинуса `sklearn` из `sklearn.metrics.pairwise` (см. рис. 19).

Для проверки точности алгоритма поиска MOOK по введенной тематике вместе со списком подходящих MOOK выведем значения меры подобия. Выведенные значение косинусного расстояния достаточны малы (рисунок 22), а значит полученные названия курсов семантически близки по смыслу друг с другом, что позволяет говорить о точности поиска MOOK с помощью РС.

```
def recommend(course):
    course_index = new_df[new_df['course_name'] == course].index[0]
    distances = similarity[course_index]
    course_list = sorted(list(enumerate(distances)), reverse=True, key=lambda x: x[1])[1:7]

    for i in course_list:
        print(new_df.iloc[i[0]].course_name)
        print (distances)
```

```
recommend('Software Security')
```

```
Web Application Security Testing with Burp Suite
[0.09753927 0.03742979 0.01719501 ... 0.14619169 0.02948531 0.17468726]
Protecting Cloud Architecture with Alibaba Cloud
[0.09753927 0.03742979 0.01719501 ... 0.14619169 0.02948531 0.17468726]
Enterprise and Infrastructure Security
[0.09753927 0.03742979 0.01719501 ... 0.14619169 0.02948531 0.17468726]
Software Design as an Element of the Software Development Lifecycle
[0.09753927 0.03742979 0.01719501 ... 0.14619169 0.02948531 0.17468726]
Cybersecurity Capstone Project
[0.09753927 0.03742979 0.01719501 ... 0.14619169 0.02948531 0.17468726]
Information Security Context and Introduction
[0.09753927 0.03742979 0.01719501 ... 0.14619169 0.02948531 0.17468726]
```

Рисунок 22 – Функция рекомендации в рекомендательной системе MOOK

Итак, в случае вызова функций рекомендаций (рисунки 21, 22) – это 6 курсов, которые рекомендуются системой на основе поиска посредством функции рекомендаций.

На данный момент РС по выбору MOOK может запускаться как прототип Web-приложения (рисунок 23), благодаря использованию Streamlit⁷ – это своего рода технология FrontEnd, которая позволяет за считанные минуты специалистам по данным трансформировать сценарии данных в Web-приложения, создавая бесплатно интерактивные приложения для ML.

⁷ Streamlit (<https://streamlit.io/>) — это библиотека на основе языка программирования Python с открытым исходным кодом для разработки и развертывания интерактивных информационных панелей, дашбордов и моделей ML. Streamlit позволяет отображать описательный текст и выходные данные модели, визуализировать данные и производительность модели, а также изменять входные данные модели.

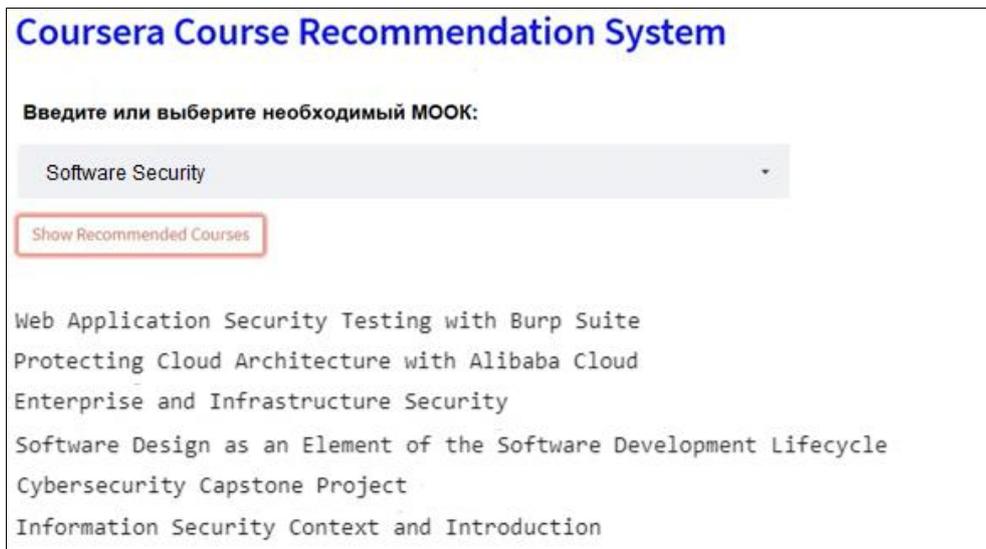


Рисунок 23 – Запуск РС по выбору МООК с помощью технологии FrontEnd (на базе Streamlit)

Ниже представлен программный код (рисунок 24) реализации простейшего Web-приложения (рунок 23), для демонстрации РС по выбору МООК в интерактивном режиме.

```
import os
import pickle
import streamlit as st
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import requests
# загрузка данных
courses_list = pickle.load(open('mook.pkl', 'rb'))
similar = pickle.load(open('similar.pkl', 'rb'))
# функция рекомендации в рекомендательной системе МООК
def recommend(course):
    index = courses_list[courses_list['course_name'] == course].index[0]
    distances = sorted(list(enumerate(similar[index])), reverse=True, key=lambda x: x[1])
    recommended_course_names = []
    for i in distances[1:7]:
        course_name = courses_list.iloc[i[0]].course_name
        recommended_course_names.append(course_name)
    return recommended_course_names

st.markdown("<h2 style='text-align: center; color: blue;'>Coursera Course Recommendation System</h2>", unsafe_allow_html=True)
course_list = courses_list['course_name'].values
selected_course = st.selectbox("Введите или выберите необходимый МООК:", course_list)
if st.button('Show Recommended Courses'):
    recommended_course_names = recommend(selected_course)
    st.text(recommended_course_names[0])
    st.text(recommended_course_names[1])
    st.text(recommended_course_names[2])
    st.text(recommended_course_names[3])
    st.text(recommended_course_names[4])
    st.text(recommended_course_names[5])
```

Рисунок 24 – Код страницы Web (на базе Streamlit)

Предлагается внедрить систему рекомендаций в блок контроля и диагностики электронного учебно-методического комплекса дисциплины, что позволит формировать персональный учебный контент обучаемого. На рисунке 25 представлен обозначенный блок, являющийся неотъемлемым компонентом мультимедийного электронного учебно-методического комплекса дисциплины «Цифровая грамотность»



Рисунок 25 – Фрагмент мультимедийного электронного учебно-методического комплекса дисциплины «Цифровая грамотность»

Итоги по главе 2

Подводя итоги по второй главе, необходимо отметить, что выбран язык программирования и цифровая среда для разработки алгоритмического обеспечения рекомендатора персонального учебного контента. Таким образом, разработан прототипа рекомендатора персонального учебного контента посредством языка программирования высокого уровня Python. Так же создана функциональная модель прототипа рекомендатора персонального

учебного контента, разработаны схемы и алгоритмическое обеспечение РС по успеваемости обучаемого и выбору MOOK.

Описан функционал прототипа рекомендатора персонального учебного контента:

- Формирование перечня тем по каждому обучаемому с процентом освоения.
- Формирование перечня тем дисциплины, которые необходимо изучить.
- Прогнозирование паттерна успешного освоения соответствующих тем/ разделов/ модулей для освоения дисциплины
- Прогнозирование паттерна разделов или модулей или тем дисциплины, которые вызывают затруднения у большинства обучаемых.
- Формирование подборки необходимых MOOK студенту для успешного освоения дисциплины.

Глава 3 Анализ результатов

Основная метрика оценки прототипа рекомендатора с целью формирования персонального контента обучаемого – уровень востребованности и значимости для преподавателя так называемого непосредственного пользователя подобных систем в учебном процессе.

Для выполнения оценки данной метрики в ходе прохождения преддипломной практики на базе федерального государственного бюджетного образовательного учреждения высшего образования «Омский государственный педагогический университет» (далее ОмГПУ) была проведена апробация и демонстрация использования прототипа рекомендатора и затем последующее анкетирование ППС.

Преподавателям ОмГПУ была предложена Анкета 1, позволяющая выяснить, как они сами оценивают потенциал и перспективы использования методов и средств ИИ на примере применения прототипа рекомендатора для формирования персонального учебного контента обучаемого в зависимости от его познавательных потребностей.

Вопросы Анкеты 1 представлены в Приложении 1, а результаты обработки ответов ППС ОмГПУ приведены в таблице 1.

Таблица 1 – Статистические данные по результатам анкетирования ППС

№ вопроса анкеты	Ответы ППС			
		использую	редко использую	не использую
1.		28	2	0
	итого	93%	7%	0%

Продолжение таблицы 1

№ вопроса анкеты	Ответы ППС			
		да	редко	Нет
2.		24	4	2
	итого	80%	13%	7%
		да	возможно	затрудняюсь ответить
3.		22	5	3
	итого	73%	17%	10%
		да	частично	нет
4.		15	9	6
	итого	50%	30%	20%
		да	не знаю	нет
5.		28	2	0
	итого	93%	7%	0%
		да	не знаю	нет
6.		26	2	2
	итого	86%	7%	7%
		да	возможно	нет
7.		30	0	0
	итого	100%	0%	0%
		да	не знаю	нет
8.		26	2	2
	итого	86%	7%	7%
		да	не знаю	нет
9.		26	2	2
	итого	86%	7%	7%
		да	не знаю	нет
10.		30	0	0
	итого	100%	0%	0%
		да	возможно	нет
11.		30	0	0
	итого	100%	0%	0%
		да	не знаю	нет
12.		30	0	0
	итого	100%	0%	0%

На графике (рисунок 26) представлены результаты анкетирования, которые демонстрируют значимость функционала предлагаемого рекомандатора персонального учебного контента, оцененная ревьюерами (т. е. ППС).

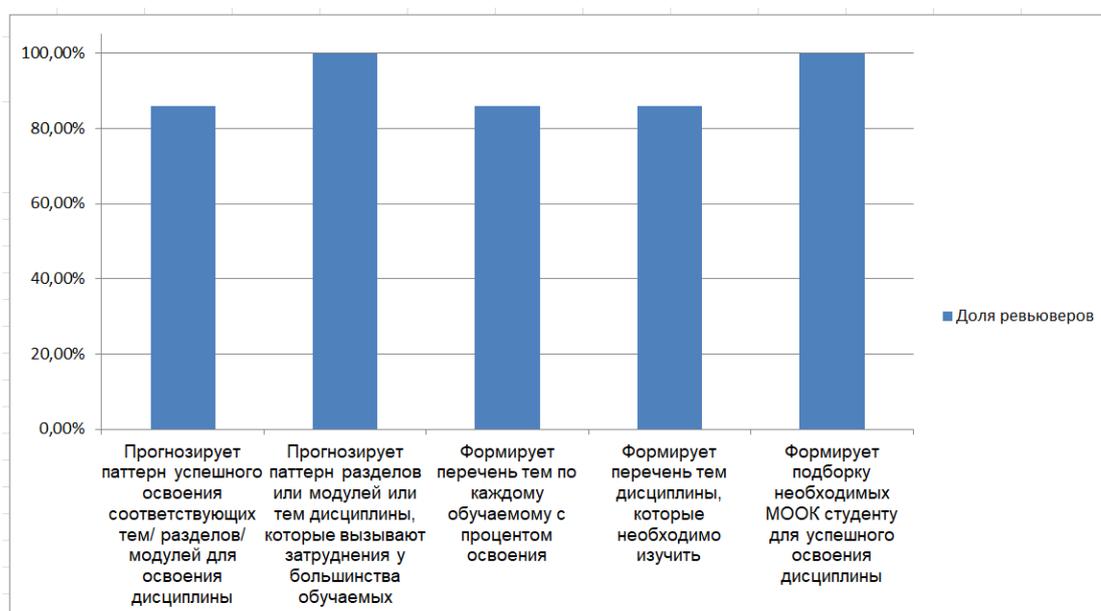


Рисунок 26 –. Значимость функционала рекомендатора персонального учебного контента

Анализ результатов Анкеты 1 позволяет отметить, что 93% опрошенных преподавателей ответили, что используют в своей работе мультимедийный электронный учебно-методический комплекс дисциплины. При этом 80% из опрошенных преподавателей сталкивался в своей работе с необходимостью формирования учебного материала в соответствии с познавательными потребностями студента для успешного усвоения им дисциплины, которую преподает.

Как уже отмечалось ранее, в настоящее время национальной системе образования на всех уровнях, как и во всем мире, особо актуально решение проблемы проектирования персонального образовательного трека обучаемого, в том числе учебного контента с учетом его познавательных потребностей, что является одной из важных составляющих адаптивного обучения. В этой связи, становится понятным, что 73 % преподавателей согласны, что одной из причин, по которой происходит сложность формирования персональной образовательного трека обучаемого, – это недостаточное использование в учебном процессе потенциала методов и средств ИИ.

Интересен тот факт, что только 50% преподавателей удовлетворены своими знаниями и умениями использования средств искусственного интеллекта в сфере образования. Немаловажен тот факт, что 93% опрошенных преподавателей интересуются какие технологии ИИ способствуют эффективному решению проблемы формирования персонального образовательного трека обучаемого в университете для повышения качества обучения.

Большая часть опрошенных (86%) отметили, что рекомендатор вполне может формировать перечень тем по каждому обучаемому с процентом освоения, а также формировать перечень тем дисциплины, которые необходимо изучить – и это будет способствовать лучшему пониманию преподавателем персонального результата освоения дисциплины обучаемым.

Большинство опрошенных преподавателей (87%) считают, что алгоритмы МО смогут спрогнозировать паттерн успешного освоения соответствующих тем/ разделов/ модулей для получения положительной оценки по дисциплине. Все опрошенные преподаватели (100%) считают, что базовый функционал рекомендательных систем на основе содержательного подхода будет способствовать подборке необходимых МООК студенту для успешного освоения соответствующих тем/ разделов/ модулей дисциплины.

Все опрошенные преподаватели (100%) согласны с тем, что алгоритмы МО смогут спрогнозировать паттерн разделов или модулей или тем дисциплины, которые вызывают затруднения у большинства обучаемых. Не маловажен тот факт, что все опрошенные преподаватели (100%) считают целесообразным внедрить рекомендатор как в систему дистанционного обучения Moodle, так и в блок контроля и диагностики мультимедийного электронного учебно-методического комплекса дисциплины для формирования персонального учебного контента обучаемого с целью повышения уровня развития его компетенций по дисциплине.

Итоги по главе 3

Подводя итоги по третьей главе, необходимо отметить, что в качестве основной метрики оценки прототипа рекомендатора с целью формирования персонального контента обучаемого было взято выявление уровня востребованности и значимости для преподавателя так называемого непосредственного пользователя подобных систем в учебном процессе.

Преподавателям была предложена Анкета 1, позволяющая выяснить, как они сами оценивают потенциал и перспективы использования методов и средств ИИ на примере применения прототипа рекомендатора для формирования персонального учебного контента обучаемого в зависимости от его познавательных потребностей

На графике (рисунок 21) представлены результаты анкетирования, которые наглядно показывают значимость и практическую полезность применения рекомендатора в образовательном процессе с целью повышения качества обучения. Таким образом, результаты анкетирования преподавателей полностью обосновывают и подтверждают выдвинутую гипотезу исследования.

Заключение

В заключение необходимо отметить, что в контексте данной научной работы исследованы методы и технологии ИИ, выявлен их потенциал для создания учебного контента по дисциплине с целью формирования персонального образовательного трека обучаемого на основе его итоговых учебных данных по дисциплине. Так же рассмотрены основные виды рекомендательных систем и принципы их построения для систематизации учебного контента обучаемого и формирования персонального учебного контента обучаемого.

В результате проведения исследования в рамках магистерской диссертации разработан прототип рекомендатора персонального учебного контента обучаемого. Так же представлена функциональную модель прототипа разработанного рекомендатора, разработаны схемы и алгоритмическое обеспечение РС по успеваемости обучаемого и выбору МООК.

Описан функционал прототипа рекомендатора персонального учебного контента:

- Формирование перечня тем по каждому обучаемому с процентом освоения.
- Формирование перечня тем дисциплины, которые необходимо изучить.
- Прогнозирование паттерна успешного освоения соответствующих тем/ разделов/ модулей для освоения дисциплины
- Прогнозирование паттерна разделов или модулей или тем дисциплины, которые вызывают затруднения у большинства обучаемых.
- Формирование подборки необходимых МООК студенту для успешного освоения дисциплины.

Предлагается внедрить систему рекомендаций в блок контроля и диагностики электронного учебно-методического комплекса дисциплины, что позволит формировать персональный учебный контент обучаемого.

Область применения: прототип рекомендательной системы ориентирован в большей степени на интеграцию с мультимедийным электронным учебно-методическим комплексом дисциплины, а именно внедрить систему рекомендаций в его блок контроля и диагностики, что позволит формировать персональный учебный контент обучаемого.

Основные характеристики: прототип рекомендательной системы разработан на языке программирования Python; для хранения и анализа данных используется система файлов в формате *.CSV.

Основная метрика оценки прототипа рекомендатора с целью формирования персонального контента обучаемого – уровень востребованности и значимости для преподавателя так называемого непосредственного пользователя подобных систем в учебном процессе. Результаты анкетирования ППС позволяют констатировать значимость и практическую полезность применения рекомендатора в образовательном процессе с целью повышения качества обучения.

В перспективе планируется интеграция разработанного прототипа рекомендательной системы с мультимедийным электронным учебно-методическим комплексом дисциплины.

Список используемой литературы

1. Анатомия рекомендательных систем. Часть первая. – URL: <https://habr.com/ru/company/lanit/blog/420499/> (дата обращения: 30.05.2023).
2. Атанов, Г. А., Пустынникова, И. Н. Обучение и искусственный интеллект, или основы современной дидактики высшей школы. – Донецк: изд-во ДОУ, 2002. – 504 с.
3. Афанасьев, А. Н. Разработка экспериментальной компьютерной программы «Рекомендательная система для САПР КОМПАС-3D» / А. Н. Афанасьев, С. И. Бригаднов // Вестник Ульяновского государственного технического университета. – 2017. – № 4. – С. 33-36.
4. Берман, К. Основы Python для Data Science. – СПб. : Питер, 2017. – 272 с.
5. Бадалходжаев, Т. И. Искусственный интеллект в образовании / Т. И. Бадалходжаев // Информационно-коммуникационные технологии в педагогическом образовании. – 2022. – № 5(80). – С. 7-9.
6. Болотова, Л. С. Системы искусственного интеллекта: модели и технологии основанные на знаниях / Л. С. Болотова. – Москва: Финансы и статистика, 2012. – 664 с.
7. Брюс, П. и др. Практическая статистика для специалистов Data Science. 50+ важнейших понятий с использованием R и Python. – СПб.: БХВ-Петербург. 2021. – 352 с.
8. Ваграменко, Я. А., Яламов, Г. Ю. Анализ направлен на интеллектуализацию современных информационных систем образовательного назначения // Управление образованием: теория и практика. 2016. №4 (24). URL: <https://cyberleninka.ru/article/n/analiz-napravleniy-intellectualizatsii-sovremennyh-informatsionnyh-sistem-uchebnogo-naznacheniya> (дата обращения: 30.05.2023).

9. Валько, Д. В. Рекомендательная система на основе интеллектуального анализа наукометрического профиля исследователя / Д. В. Валько // Программные продукты и системы. – 2018. – № 2. – С. 275-283.
10. Васильева, Т. В. Введение в магистерскую программу : учебное пособие / Т. В. Васильева. – 2 – е изд. – Москва : Ай Пи Ар Медиа, 2022. – 109 с. – ISBN 978 – 5 – 4497 – 1703 – URL: <https://www.iprbookshop.ru/122010.html?replacement=1>
11. Гаврилова, Т. А. Инженерия знаний. Модели и методы / Т. А. Гаврилова, Д. В. Кудрявцев, Д. И. Муромцев. — 6-е изд., стер. — Санкт-Петербург : Лань, 2023. — 324 с.
12. Гамбеева, Ю. Н. Искусственный интеллект как часть концепции современного образования: вызовы и перспективы / Ю. Н. Гамбеева, А. В. Глотова // Известия Волгоградского государственного педагогического университета. – 2021. – № 10(163). – С. 10-16.
13. Глибовец, Н. Н. Создание рекомендационной системы учебного типа с использованием фреймворка / Н. Н. Глибовец, М. О. Сидоренко // Проблемы интеллектуализации компьютера : сб. ст. / Институт кибернетики им. В. М. Глушкова НАН Украины. – Киев, 2012. С. 176 – 181.
14. Гмурман, В. Е. Теория вероятностей и математическая статистика. Учеб. пособие для вузов. Изд. 7-е, стер. – М.: Высш. шк., 1999. – 479 с.
15. Гэддис, Т. Начинаем программировать на Python. – 5-е изд.: Пер. с англ. – СПб.: БХВ-Петербург, 2022. – 880 с.
16. Грабарь, М. И. Применение математической статистики в педагогических исследованиях. Непараметрические методы / М. И. Грабарь, К. А. Краснянская. – М.: Педагогика, 1977. – 136 с.
17. Грас, Д. Data Science. Наука с нуля: Пер. с англ. СПб.: БХВ-Петербург. 2017. 336 с.
18. Гудфеллоу Я., Бенджио И., Курвилль А. Глубокое обучение / пер. с англ. А. А. Слинкина. – 2-е изд., испр. – М.: ДМК Пресс, 2018. – 652 с.:

19. Ендовицкий, Д. А. Университетская наука и образование в контексте искусственного интеллекта / Д. А. Ендовицкий, К. М. Гайдар // Высшее образование в России. – 2021. – Т. 30. – № 6. – С. 121-131. – DOI 10.31992/0869-3617-2021-30-6-121-131.
20. Златопольский, Д. М. Основы программирования на языке Python. – М.: ДМК Пресс, 2017. – 284 с
21. Искусственный интеллект в образовании. – URL: <https://cs.hse.ru/aicenter/education> (дата обращения: 30.05.2023)
22. Искусственный интеллект в образовании: Изменение темпов обучения. Аналитическая записка ИИТО ЮНЕСКО / Стивен Даггэн; ред. С. Ю. Князева; пер. с англ.: А. В. Паршакова. – Москва : Институт ЮНЕСКО по информационным технологиям в образовании, 2020.
23. Калянов, Г. Н. CASE-технологии. Консалтинг в автоматизации бизнес-процессов. – М.: Горячая линия-Телеком, 2002. – 320 с.
24. Карпенко, О. М. Искусственный интеллект в образовании / О. М. Карпенко, М. П. Карпенко // Инновации в образовании. – 2022. – № 9. – С. 4-9.
25. Коляда, М. Г. Искусственный интеллект как движущая сила совершенствования и инновационного развития в образовании и педагогике / М. Г. Коляда, Т. И. Бугаева // Информатика и образование. – 2019. – № 10(309). – С. 21-30. – DOI 10.32517/0234-0453-2018-33-10-21-30.
26. Корнилов, Д. Искусственный интеллект в российских школах поможет ученикам лучше усваивать материал. – URL: https://fbm.ru/novosti/computers/iskusstvennyj-intellekt-v-rossijskih-shkolah-pomozhet-uchenikam-luchshe-usvaivat-material.html?utm_source=yxnews&utm_medium=desktop&utm_referrer=https%3A%2F%2Fyandex.ru%2Fnews%2Fsearch%3Ftext%3D (дата обращения: 30.05.2023)

27. Коэльо Л. П., Ричарт, В. Построение систем машинного обучения на языке Python / перевод с английского А. А. Слинкин. – 2-е изд.– М.: ДМК Пресс, 2016. – 302 с.
28. Лесковец Ю., Раджараман А., Ульман Д. Анализ больших наборов данных. / Пер. с англ. Слинкин А. А. – М.: ДМК Пресс, 2016. – 498 с.:
29. Ляликова, В. Г., Безрядин, М. М. Построение гибридной рекомендательной системы. Вестник ВГУ. Серия: Системный анализ и информационные технологии, 2021 (4), 121-129. <https://doi.org/10.17308/sait.2021.4/3802>
30. Маккинни, У. Python и анализ данных. – Москва : ДМК Пресс, 2023. – 536 с.
31. Марка, Дэвид А., МакГоуэн, К. Методология структурного анализа и проектирования SADT. – М.: Мета Технология, 2009. – 239 с.
32. Моисеева, Н. А. Перспективы и проблемы интеграции технологий искусственного интеллекта в образование / Н. А. Моисеева // Информатизация образования: теория и практика : сборник материалов Международной научно-практической конференции памяти академика РАО М. П. Лапчика, Омск, 18–19 ноября 2022 года. – Омск: Федеральное государственное бюджетное образовательное учреждение высшего образования «Омский государственный педагогический университет», 2022. – С. 64-67.
33. Мухамадиева, К. Б. Машинное обучение в совершенствовании образовательной среды // Образование и проблемы развития общества. 2020. №4 (13). URL: <https://cyberleninka.ru/article/n/mashinnoe-obuchenie-v-sovershenstvovanii-obrazovatelnoy-sredy> (дата обращения: 10.03.2023).
34. Намазбек Улуу, А. Реализация рекомендательной системы на основе метода к-средних / А. Намазбек Улуу, М. В. Коржов, И. В. Хмелева // Вестник Кыргызско-Российского Славянского университета. – 2017. – Т. 17, № 5. – С. 87-89.
35. Новиков, Д. А. Статистические методы в педагогических исследованиях (типовые случаи) [Электронный ресурс] / Д. А. Новиков. – URL:

sch54.narod.ru/netschoolexperiment/results/metodika/novikov/pedstat.pdf

(30.05.2023)

36. Околелов, О. П. Искусственный интеллект в образовании / О. П. Околелов. – Москва-Берлин : ООО "Директмедиа Паблишинг", 2020. – 82 с.

37. Паскова, А. А. Технологии искусственного интеллекта в персонализации электронного обучения // Вестник Майкопского государственного технологического университета. 2019. №3. URL: <https://cyberleninka.ru/article/n/tehnologii-iskusstvennogo-intellekta-v-personalizatsii-elektronnogo-obucheniya> (дата обращения: 30.05.2023)

38. Плас Дж. Вандер Python для сложных задач: наука о данных и машинное обучение. – СПб.: Питер, 2018. – 576 с.

39. Прохоренок, Н. Python 3 и PyQt 6. Разработка приложений / Н. А. Прохоренок, В. А. Дронов. – СПб.: Б:ХВ-Петербург, 2023. – 832 с.

40. Пырнова, О. А., Зарипова, Р. С. Технологии искусственного интеллекта в образовании // Russian Journal of Education and Psychology. 2019. №3. URL: <https://cyberleninka.ru/article/n/tehnologii-iskusstvennogointellekta-v-obrazovanii> (дата обращения: 30.05.2023).

41. Рашка, С. Python и машинное обучение. – М.: Издательство ДМК Пресс, 2017 г. – 418 с.

42. Садыкова, А. Р. Искусственный интеллект как компонент инновационного содержания общего образования: анализ мирового опыта и отечественные перспективы / А. Р. Садыкова, И. В. Левченко // Вестник Российского университета дружбы народов. Серия: Информатизация образования. – 2020. – Т. 17. – № 3. – С. 201-209. – DOI 10.22363/2312-8631-2020-17-3-201-209.

43. Седер, Н. Python. Экспресс-курс. 3-е изд. – СПб.: Питер, 2019. – 480 с.

44. Силен, Д. и др. Основы Data Science и Big Data. Python и наука о данных. – СПб.: Питер, 2017. – 336 с.

45. Скиен, С. Наука о данных. Учебный курс. – М. : Вильямс 2020. – 544 с.
46. Таратухина Ю., Барт Т., Власов В. Машинное обучение модели информационной рекомендательной системы по вопросам индивидуализации образования // Образовательные ресурсы и технологии. 2019. № 2 (27). С. 7-14. DOI: 10.21777/2500-2112-2019-2-7-14
47. Холмс, У. и др. Искусственный интеллект в образовании : Перспективы и проблемы для преподавания и обучения / У. Холмс, М. Бялик, Ч. Фейдел. – Москва : Альпина ПРО, 2022. – 304 с.
48. Шихи, Д. Структуры данных в Python: начальный курс / пер. с англ. А. В. Снастина. – М.: ДМК Пресс, 2022. – 186 с.:
49. Шкодырев, В. П. Вопросы стандартизации и взаимосвязи образования и искусственного интеллекта / В. П. Шкодырев, Н. М. Куприков, Е. А. Башкирова // Известия Тульского государственного университета. Технические науки. – 2022. – № 7. – С. 83-87. – DOI 10.24412/2071-6168-2022-7-83-87.
50. Учебник для начинающих: рекомендательные системы в PythonBeginner Tutorial: Recommender Systems in Python. – URL: <https://www.datacamp.com/tutorial/recommender-systems-python#rdl> (дата обращения: 30.05.2023)
51. Фальк, К. Рекомендательные системы на практике : руководство / К. Фальк ; перевод с английского Д. М. Павлова. – Москва : ДМК Пресс, 2020. – 448 с.
52. Федеральный закон № 472-ФЗ «Об образовании в Российской Федерации». – URL: <http://publication.pravo.gov.ru/Document/View/0001202112300167> (дата обращения: 10.03.2023)
53. Федеральный проект «Цифровая образовательная среда». <https://edu.gov.ru/national-project/projects/cos/>

54. Шкодина, Т. А. Разработка архитектуры рекомендательной системы по выбору онлайн-курсов / Т. А. Шкодина // Прикладная информатика. – 2022. – Т. 17, № 4(100). – С. 87-96. – DOI 10.37791/2687-0649-2022-17-4-87-96.
55. Шполянская, И. Ю., Середкина, Т. А. Технология Semantic Web в организации поддержки онлайн обучения // Системный анализ в проектировании и управлении: сборник научных трудов XXIV Международной научной и учебно-практической конференции: в 3 ч. / под общ. ред. Г. В. Гореловой, А. В. Логиновой. – СПб., 2020. С. 343-350.
56. Burke, R. Hybrid Recommender Systems: Survey and Experiments. *User Model User-Adap Inter* 12, 331–370 (2002). <https://doi.org/10.1023/A:1021240730564>
57. Fernández-García, A. J. et al. (2020). Creating a Recommender System to Support Higher Education Students in the Subject Enrollment Decision, in *IEEE Access*, vol. 8, pp. 189069-189088, 2020, doi: 10.1109/ACCESS.2020.3031572.
58. Holmes, W., et al. (2018). Technology-Enhanced Personalised Learning Untangling the Evidence. Robert Bosch Stiftung
59. Khosravi, H. (2017). Recommendation in Personalised Peer-Learning Environments. – URL: https://www.researchgate.net/publication/321719519_Recommendation_in_Personalised_Peer-Learning_Environments (data access: 30.05.2023)
60. Klinkenberg, S. et al. Computer adaptive practice of maths ability using a new item response model for on the fly ability and difficulty estimation // *Computers & Education*. – 2011. – Vol. 57, no. 2. – P. 1813–1824.
61. Kulkarni, Pradnya & Rai, Sunil & Kale, Rohini. (2020). Recommender System in eLearning: A Survey. 10.1007/978-981-15-0790-8_13.
62. Luckin, R., et al. (2016) *Intelligence Unleashed. An Argument for AI in Education*. Pearson. <https://discovery.ucl.ac.uk/id/eprint/1475756/> (data access: 30.05.2023)

63. Maiti, M., Priyaadharshini, M. Recommender System for Low Achievers in Higher Education," *International Journal of Information and Education Technology* vol. 12, no. 12, pp. 1390-1398, 2022.
64. Park, D. H., Kim, H. K., Choi, I. Y., Kim, J. K. A literature review and classification of recommender systems research // *Expert Systems with Applications*. 2012. Vol. 39. No. 11. P. 10059-10072. DOI: 10.1016/j.eswa.2012.02.038
65. Robin, B. Hybrid recommender systems: Survey and experiments. *User modeling and user adapted interaction*, 12(4):331–370, 2002.
66. Salehi, M. Latent feature based recommender system for learning materials using genetic algorithm // *Journal of Information Systems and Telecommunication*. 2014. Vol. 2. No. 3. P. 137-144. DOI: 10.7508/jist.2014.03.001
67. Salehi, M., Pourzaferani, M., Razavi, S. Hybrid attribute-based recommender system for learning material using genetic algorithm and a multidimensional information model // *Egyptian Informatics Journal*. 2013. Vol. 14. No. 1. P. 67-78. DOI: 10.1016/j.eij.2012.12.001
68. Salehudin, Norazuwa & Kahtan, Hasan & Abdulhak, Mansoor & Al-bashiri, Hael. (2019). A Proposed Course Recommender Model based on Collaborative Filtering for Course Registration. *International Journal of Advanced Computer Science and Applications*. 10. 10.14569/IJACSA.2019.0101122.
69. Sharma, L., Gera, A. A survey of recommendation system: Research challenges // *International Journal of Engineering Trends and Technology (IJETT)*. 2013. Vol. 4. No. 5. P. 1989-1992.
70. Shishehchi, S., Banihashem, S. Y., Zin, N. A. A proposed semantic recommendation system for e-learning: A rule and ontology-based e-learning recommendation system // *Proceedings of the International Symposium on Information Technology (ITSim)*. 2010. Vol. 1. P. 1-5.
71. Smirnov, I. Estimating educational outcomes from students' short texts on social media. *EPJ Data Sci.* 9, 27 (2020). <https://doi.org/10.1140/epjds/s13688-020-00245-8>

72. Su X., Khoshgoftaar, T. A survey of collaborative filtering techniques // Journal of Advances in Artificial Intelligence. 2009. Article 421425. P. 1-19. DOI: 10.1155/2009/421425

73. Tavakoli, Mohammadreza & Hakimov, Sherzod & Ewerth, Ralph & Kismihok, Gabor. (2020). A Recommender System For Open Educational Videos Based On Skill Requirements. 1-5. 10.1109/ICALT49669.2020.00008.

74. Vera, A. A., González, C. M. (2022). Educational Resource Recommender Systems Using Python and Moodle. In: Gervasi, O., Murgante, B., Misra, S., Rocha, A.M.A.C., Garau, C. (eds) Computational Science and Its Applications – ICCSA 2022 Workshops. ICCSA 2022. Lecture Notes in Computer Science, vol 13380. Springer, Cham. https://doi.org/10.1007/978-3-031-10542-5_2

Приложение А

Анкета 1

1. Используете ли Вы в своей работе мультимедийный электронный учебно-методический комплекс дисциплины (ЭУМКД)?

а) всегда использую б) редко использую в) не использую

2. Сталкиваетесь ли Вы в своей работе с необходимостью формирования учебного материала в соответствии с познавательными потребностями студента для успешного усвоения им дисциплины, которую Вы преподаете?

а) да б) редко в) нет

3. Считаете ли Вы, что одной из причин, по которой происходит сложность формирования персональной образовательной траектории студента, – это недостаточное использование в учебном процессе потенциала методов и средств искусственного интеллекта?

а) да б) возможно в) затрудняюсь ответить

4. Удовлетворены ли Вы компетенциями применения технологий искусственного интеллекта в сфере образования?

а) да б) частично в) нет

5. Хотели бы Вы узнать о технологиях искусственного интеллекта, способствующие эффективному решению проблемы формирования персонального образовательного трека обучаемого в университете для повышения качества обучения?

а) да б) не знаю в) нет

Продолжение Приложения А

6. Считаете ли Вы, что алгоритмы машинного обучения смогут спрогнозировать паттерн успешного освоения соответствующих тем/разделов/ модулей для получения положительной оценки по дисциплине?

- а) да б) не знаю в) нет

7. Как Вы считаете, базовый функционал рекомендательных систем на основе содержательного подхода будет способствовать подборке необходимых МООК студенту для успешного освоения соответствующих тем/разделов/ модулей дисциплины?

- а) да б) не знаю в) нет

8. Как Вы считаете то, что рекомендатор может формировать перечень тем по каждому обучаемому с процентом освоения – это будет способствовать лучшему пониманию преподавателем персонального результата освоения дисциплины обучаемым?

- а) да б) не знаю в) нет

9. Как Вы считаете то, что рекомендатор может формировать перечень тем дисциплины, которые необходимо изучить – это будет способствовать пониманию преподавателем какие темы следует обучаемому повторить/дополнительно изучить?

- а) да б) не знаю в) нет

10. Считаете ли Вы, что алгоритмы машинного обучения смогут спрогнозировать паттерн разделов или модулей или тем дисциплины, которые вызывают затруднения у большинства обучаемых?

- а) да б) возможно в) нет

Продолжение Приложения А

11. Считаете ли Вы целесообразным внедрить рекомендатор в СДО Moodle для формирования персонального учебного контента обучаемого с целью повышения уровня развития его компетенций по дисциплине?

- а) да б) не знаю в) нет

12. Считаете ли Вы целесообразным внедрить рекомендатор в блок контроля и диагностики мультимедийного ЭУМКД для формирования персонального учебного контента обучаемого с целью повышения уровня развития его компетенций по дисциплине?

- а) да б) не знаю в) нет