АКТУАЛИЗИРОВАНО
Решением Ученого совета ИЦТЭ КГЭУ
Протокол №7 от 24.03.2026

К Г Э У

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования
«КАЗАНСКИЙ ГОСУДАРСТВЕННЫЙ ЭНЕРГЕТИЧЕСКИЙ УНИВЕРСИТЕТ»
(ФГБОУ ВО «КГЭУ»)

КГЭУ
Подписан: ФГБОУ ВО «КГЭУ»,
КАЗАНСКИЙ ГОСУДАРСТВЕННЫЙ ЭНЕРГЕТИЧЕСИЙ УНИВЕРСИТЕТ
Владелец: Зайнуллин Радик Рустэмович,
Директор института цифровых технологий и экономики,
Сертификат: 28C40A82922670A2466A21B6019CE75F
Действителен с 13.03.2026 по 13.03.2027

УТВЕРЖДАЮ
Директор Института цифровых технологий и экономики
_____ Э.И. Беляев
«_____»_____ 2023 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Б1.В.13 Рекомендательные системы

Направление подготовки 09.03.03 Прикладная информатика
(*Код и наименование направления подготовки*)

Направленность(и) (профиль(и)) Прикладной искусственный интеллект
(*Наименование направленности (профиля) образовательной программы*)

Квалификация Бакалавр
(*Бакалавр / Магистр*)

г. Казань, 2023

Программу разработали:

Наименование кафедры	Должность, уч.степень, уч.звание	ФИО разработчика
УрФУ БК «АБДиМВ»	Доцент, к.э.н., доцент	Медведев М.А.
УрФУ БК «АБДиМВ»	Ассистент	Балунгу Д.
КГЭУ ИТИС	Ассистент	Афанасьев А.Л.

Согласование	Наименование подразделения	Дата	№ протокола	Подпись
Одобрена	кафедра ИТИС	27.11.2023	11	_____ И.о. зав.каф.,к.ф.-м.н., доц. Соловьев С. А.
Согласована	Учебно-методический совет института ИЦТЭ	27.11.2023	3	_____ Директор, к.т.н., доц. Беляев Э.И.
Одобрена	Ученый совет института ИЦТЭ	28.11.2023	3	_____ Директор, к.т.н., доц. Беляев Э.И.

1. Цель, задачи и планируемые результаты обучения по дисциплине

(*Цель и задачи освоения дисциплины, соответствующие цели ОП*)

Целью преподавания дисциплины «Рекомендательные системы» является ознакомление с принципами работы рекомендательных систем и рассмотрение вопросов, связанных с особенностями проектирования, использования подобных систем.

Задачами дисциплины являются: изучение основных концепций и принципов рекомендательных систем; исследование различных алгоритмов, используемых в рекомендательных системах; анализ и сравнение различных метрик для оценки эффективности рекомендательных систем; исследование проблем и вызовов, связанных с построением и эксплуатацией рекомендательных систем; практическое применение полученных знаний через разработку собственной рекомендательной системы или проведение экспериментов с существующими системами.

Компетенции и индикаторы, формируемые у обучающихся:

Код и наименование компетенции	**Индикаторы достижения компетенции**
2	3
ПК-2. Способен разрабатывать и тестировать программные компоненты решения задач в системах искусственного интеллекта	ПК-2.1. Разрабатывает приложения систем искусственного интеллекта
	ПК-2.2. Проводит тестирование систем искусственного интеллекта
ПК-8. Способен создавать и внедрять одну или несколько сквозных цифровых субтехнологий искусственного интеллекта	ПК-8.1. Участвует в реализации проектов в области сквозной цифровой субтехнологии «Рекомендательные системы и системы поддержки принятия решений»
	ПК-8.2. Участвует в реализации проектов в области сквозной цифровой субтехнологии «Компьютерное зрение»

2. Место дисциплины в структуре ОП

Предшествующие дисциплины (модули), практики, НИР, др.: Учебная практика (научно-исследовательская работа (получение первичных навыков научно-исследовательской работы)), Теория и практика программной инженерии, Виртуализация и облачные технологии

Последующие дисциплины и практики: Производственная практика (технологическая), государственная итоговая аттестация, Подготовка к процедуре защиты и защита выпускной квалификационной работы

3. Структура и содержание дисциплины

3.1. Структура дисциплины

Для очной формы обучения

Вид учебной работы	Всего ЗЕ	Всего часов	Семестр 7	Семестр 8
ОБЩАЯ ТРУДОЕМКОСТЬ ДИСЦИПЛИНЫ	7	252	106	146
КОНТАКТНАЯ РАБОТА*	2			
АУДИТОРНАЯ РАБОТА	2	72	34	38
Лекции	1	36	18	18
Практические (семинарские) занятия	-	-	-	-
Лабораторные работы	1	36	16	20
САМОСТОЯТЕЛЬНАЯ РАБОТА ОБУЧАЮЩЕГОСЯ	4	144	72	72
Проработка учебного материала	3	108	54	54
Курсовой проект	-	-	-	-
Курсовая работа	-	-	-	-
Подготовка к промежуточной аттестации	1	36		36
Промежуточная аттестация:			З	Э

3.2. Содержание дисциплины, структурированное по разделам и видам занятий

Разделы дисциплины	Всего часов	Распределение трудоемкости по видам учебной работы				Формы и вид контроля	Индексы индикаторов формируемых компетенций
		лекции	лаб. раб.	пр. зан.	сам. раб.		
7 семестр							
Тема 1 Введение в рекомендательные системы	**34**	6	4		24		
Тема 2 Коллаборативная (совместная) фильтрация	**36**	6	6		24	ТК1	ПК-2.1, ПК-2.2, ПК-8.1, ПК-8.2
Тема 3 Фильтрация на основе контента	**36**	6	6		24	ТК2	ПК-2.1, ПК-2.2, ПК-8.1, ПК-8.2
Итого за 7 семестр	**106**	**18**	**16**		**72**	ОМ	
8 семестр							
Тема 4 Гибридные рекомендательные	**36**	6	6		24	ТК3	ПК-2.1, ПК-2.2, ПК-8.1, ПК-8.2

системы							
Тема 5 Контекстно-зависимые рекомендательные системы	**36**	6	6		24	ТК4	ПК-2.1, ПК-2.2, ПК-8.1, ПК-8.2
Тема 6 Оценка рекомендательных систем	**38**	6	8		24	ТК5	ПК-2.1, ПК-2.2, ПК-8.1, ПК-8.2
Экзамен	**36**				36		
Итого за 8 семестр	**146**	**18**	**20**		**108**		
ИТОГО	**252**	**36**	**36**		**144**	ОМ	

3.3. Содержание дисциплины

Тема 1 Введение в рекомендательные системы

Важность и применение рекомендательных систем. Типы рекомендательных систем. Данные о взаимодействии пользователя с элементом и их актуальность

Тема 2 Коллаборативная (совместная) фильтрация

Подходы, основанные на памяти (основанные на пользователях и элементах). Подходы, основанные на моделях (матричная факторизация, модели скрытых факторов)

Тема 3 Фильтрация на основе контента

Извлечение и представление признаков. Показатели сходства для рекомендаций на основе контента

Тема 4 Гибридные рекомендательные системы

Сочетание совместной фильтрации и фильтрации на основе контента

Тема 5 Контекстно-зависимые рекомендательные системы

Включение контекстуальной информации (время, местоположение и т.д.) в рекомендации

Тема 6 Оценка рекомендательных систем

Показатели оценки (RMSE, точность, отзыв и т.д.) - Автономные и онлайн-методы оценки

3.4. Тематический план практических занятий

Данный вид работы не предусмотрен учебным планом.

3.5. Тематический план лабораторных работ

1. Коллаборативная фильтрация на основе сходства попользователям (user-based) и продуктам (item-based).
2. Матричная факторизация.
3. Гибридная система рекомендаций с использованием Python.
4. Контекстно-зависимые рекомендации корзины и персонализированные рекомендации.
5. Разработка рекомендательной системы на Python

3.6. Курсовой проект /курсовая работа

Данный вид работы не предусмотрен учебным планом.

4. Оценивание результатов обучения

Оценивание результатов обучения по дисциплине осуществляется в рамках текущего контроля и промежуточной аттестации, проводимых по балльно-рейтинговой системе (БРС).

Шкала оценки результатов обучения по дисциплине:

Код компе-тенции	Код индикатора компетенции	Заплани-рованные результаты обучения по дисциплине	Уровень сформированности индикатора компетенции			
			Высокий	Средний	Ниже среднего	Низкий
			от 85 до 100	от 70 до 84	от 55 до 69	от 0 до 54
			Шкала оценивания			
			отлично	хорошо	удовлет-ворительно	неудов-летвори-тельно
			зачтено			не зачтено
ПК-2	ПК-2.1	знать:.				
		основы теории рекомендательных систем и принципы систем поддержки принятия решений	Глубокое понимание основных моделей и подходов в рекомендательных системах.	Базовое понимание методов и принципов в поддержки и принятия решений.	Некоторое знание основных моделей и подходов в рекомендательных системах.	Осведомленность о существовании рекомендательных систем, но отсутствие знаний о их основах.
		уметь:				
		анализировать требования и потребности пользователей, прототипировать системы, а также интегрировать их в существующую инфраструктуру	Глубокое умение анализировать требования и потребности пользователей, проводить детальный анализ и выявлять ключевые факторы.	Навыки прототипирования систем, способность создавать основные прототипы, удовлетворяющие основным требованиям пользователей.	Некоторое умение анализировать требования и потребности пользователей, способность провести поверхностный анализ и выявить некоторые факторы.	Осведомленность о необходимости интеграции системы в существующую инфраструктуру, но отсутствие практического опыта и навыков

						интеграции.
		владеть:.				
		программированием, способностью разрабатывать и тестировать модели рекомендательных систем, а также знать методы анализа данных и машинного обучения	Глубокие навыки программирования, умение разрабатывать программный код высокого качества.	Навыки разработки и тестирования моделей рекомендательных систем, способность создавать базовые модели и проводить их тестирование.	Основное знание методов анализа данных и машинного обучения, способность применять их базовые принципы для работы с моделями рекомендательных систем.	Осведомленность о разработке и тестировании моделей рекомендательных систем, но отсутствие практического опыта и навыков работы с ними.
	ПК-2.2	знать:				
		основы алгоритмов обработки изображений и компьютерного зрения.	Глубокое знание основных алгоритмов обработки изображений, включая фильтрацию, сегментацию и детекцию объектов.	Базовое понимание основ компьютерного зрения, включая основные методы распознавания образов и классификации.	Некоторое понимание основ компьютерного зрения, включая базовые методы распознавания образов и классификации.	Осведомленность о существовании алгоритмов обработки изображений, но отсутствие знаний об их основах и принципах работы.
		уметь:				
		применять методы распознавания образов, классификации и сегментации изображений.	Глубокие навыки применения методов распознавания образов, классификации и сегментации	Способность использовать и настраивать существующие модели и алгоритмы для распознав	Основные навыки применения методов распознавания образов, классификации и сегментации	Осведомленность о применении готовых моделей и алгоритмов, но отсутствие опыта

			изображений.	ания образов, классификации и сегментации изображений.	изображений.	и навыков настройки и использования их для нужных задач.
		владеть:				
		программированием на языках, таких как Python или C++, и иметь навыки работы с библиотеками для компьютерного зрения, например, OpenCV.	Глубокие навыки программирования на языках Python или C++, включая умение разрабатывать сложные программы и эффективный код.	Умение использовать библиотеку OpenCV для основных задач компьютерного зрения.	Основные навыки программирования на языках Python или C++, способность создавать простые программы.	Осведомленность о библиотеке OpenCV, но отсутствие практического опыта и навыков ее использования для задач компьютерного зрения.
			зачтено			не зачтено
ПК-8	ПК-8.1	знать:.				
		Основы теории рекомендательных систем, включая различные модели и подходы.	основы теории рекомендательных систем, принципы работы и алгоритмы систем поддержки принятия решений, основы алгоритмов обработки изображений и компьютерного	основы теории рекомендательных систем, принципы работы и алгоритмы систем поддержки принятия решений, основы алгоритмов обработки изображений и компьютерного	основы теории рекомендательных систем, основы алгоритмов обработки изображений и компьютерного зрения, базовое знание методов анализа данных и машинного	осведомленность о теории рекомендательных систем, алгоритмах обработки изображений и компьютерного зрения, но отсутствие практического опыта и пониман

			зрения, методы анализа данных и машинного обучения.	зрения, базовые методы анализа данных и машинного обучения.	обучения.	ия.
		уметь:				
		Проектировать и реализовывать модели и алгоритмы для рекомендательных систем и систем поддержки принятия решений.	Анализировать требования и потребности пользователей и проводить детальный анализ. - Прототипировать системы, учитывая требования пользователей.	Создавать прототипы систем, которые удовлетворяют основным требованиям пользователей. - Осуществлять базовую интеграцию систем в существующую инфраструктуру и проводить тестирование.	Проводить базовый анализ требований и потребностей пользователей. - Создавать базовые прототипы систем, удовлетворяющие основным требованиям пользователей.	Осведомленность о необходимости анализировать требования и потребности пользователей, но отсутствие практических навыков выполнения анализа. - Осведомленность о прототипировании систем, но ограниченные навыки создания прототипов.
		владеть:.				
		Знаниями и опытом работы с методами анализа данных и машинного обучения.	Навыками программирования на языках Python или C++ для разработки высокока	Навыками программирования на языках Python или C++ для разработки функцион	Основными навыками программирования на языках Python или C++ для разработк	Осведомленность о программировании на языках Python или C++, но отсутств

			чественного программного кода. - Владением методами и подходами разработки моделей и алгоритмов для рекомендательных систем и систем поддержки и принятия решений.	ального программного кода. - Работой с базовыми моделями и алгоритмами для рекомендательных систем и систем поддержки и принятия решений.	и программного кода. - Работой с базовыми моделями и алгоритмами для рекомендательных систем и систем поддержки и принятия решений.	ие практического опыта и навыков разработки программного кода.
	ПК-8.2	знать:				
		Основы алгоритмов обработки изображений, включая методы фильтрации, сегментации и детекции объектов.	основы алгоритмов обработки изображений и компьютерного зрения, методы распознавания образов, классификации и сегментации изображений, основы машинного обучения и методов анализа данных.	основы алгоритмов обработки изображений и компьютерного зрения, базовые методы распознавания образов, классификации и сегментации изображений, базовые знания машинного обучения и методов анализа данных.	основы алгоритмов обработки изображений и компьютерного зрения, основные методы распознавания образов, классификации и сегментации изображений, основы машинного обучения и методов анализа данных.	осведомленность о алгоритмах обработки изображений и компьютерного зрения, методах распознавания образов, классификации и сегментации изображений, но отсутствие практического опыта и глубокого понимания.

		уметь:				
		Разрабатывать и реализовывать модели и алгоритмы для компьютерного зрения.	Применять методы обработки изображений, компьютерного зрения и машинного обучения для решения сложных задач в области компьютерного зрения.	Применять базовые методы обработки изображений, компьютерного зрения и машинного обучения для решения задач в области компьютерного зрения	Применять основные методы обработки изображений, компьютерного зрения и машинного обучения для решения простых задач в области компьютерного зрения.	Осведомленность о методах обработки изображений, компьютерного зрения и машинного обучения, но отсутствие практического опыта и навыков их применения.
		владеть:				
		Знаниями и опытом работы с библиотеками для компьютерного зрения, такими как OpenCV, TensorFlow или PyTorch.	Навыками программирования на Python или R для обработки и анализа данных. Знаниями в области фич-инжиниринга и предобработки данных.	Знанием методов оценки и выбора моделей машинного обучения. - Умением визуализировать данные и результаты моделей машинного обучения.	Основными навыками программирования на Python или R для работы с данными и простыми алгоритмами машинного обучения. - Базовым понимание алгоритмов машинного обучения и их использованием для простых задач.	Осведомленность о программировании на Python или R, но отсутствие практического опыта и навыков работы с данными и алгоритмами машинного обучения. - Осведомленность о базовых алгоритмах

						машинного обучения, но ограниченные навыки и опыт их применения.

Оценочные материалы для проведения текущего контроля и промежуточной аттестации приведены в Приложении к рабочей программе дисциплины.

Полный комплект заданий и материалов, необходимых для оценивания результатов обучения по дисциплине, хранится на кафедре разработчика.

5. Учебно-методическое и информационное обеспечение дисциплины

5.1. Учебно-методическое обеспечение

5.1.1. Основная литература

1. Isinkaye F.O., Folajimi Y.O., Ojokoh B.A.: «Recommendation systems: Principles, methods and evaluation», 2015 г. - [Электронный ресурс]: https://www.sciencedirect.com/science/article/pii/S1110866515000341

2. Гастон В. «Introduction to Recommender Systems in 2018», 2018 г. - [Электронный ресурс]: www.tryolabs.com/blog/introduction-torecommender-systems

3. Гончаров М. «Data mining: рекомендательные системы. Collaboration Filtering», 2012 г.

4. А. Константинов «Рекомендательные системы: тематический обзор» // Национальный Открытый Университет «ИНТУИТ», 2012- [Электронный ресурс]: www.hse.ru/data/2012/12/20/1303750691/Block-3.pdf

5.2. Информационное обеспечение

5.2.1. Электронные и интернет-ресурсы

1) Государственная публичная научно-техническая библиотека http://www.gpntb.ru

2) Список библиотек, доступных в Интернет и входящих в проект «Либнет» http//www.valley.ru/-nicr/listrum.htm

3) Российская национальная библиотека http//www.rsl.ru

4) Свободная энциклопедия Википедия https://ru.wikipedia.org/

5) Портал национального общества имитационного моделирования http://simulation.su/

6) Портал российской ассоциации искусственного интеллекта https://raai.org/

№ п/п	Наименование электронных и интернет-ресурсов	Ссылка
1	*Электронно-библиотечная система «Лань»*	https://e.lanbook.com/
2	*Электронно-библиотечная система «ibooks.ru»*	https://ibooks.ru/
3	*Электронно-библиотечная система «book.ru»*	https://www.book.ru/
4	Единое окно доступа к образовательным ресурсам	http://window.edu.ru
5	Портал "Открытое образование"	http://npoed.ru
6	Аналитическая платформа Loginom быстрый старт	https://loginom.ru/platform/quick-start

5.2.2. Профессиональные базы данных / Информационно-справочные системы

№ п/п	Наименование профессиональных баз данных	Адрес	Режим доступа
1	Официальный интернет-портал правовой информации	http://pravo.gov.ru	http://pravo.gov.ru
2	Справочная правовая система «Консультант Плюс»	http://consultant.ru	http://consultant.ru
3	Справочно-правовая система по законодательству РФ	http://garant.ru	http://garant.ru

5.2.3. Лицензионное и свободно распространяемое программное обеспечение дисциплины

№ п/п	Наименование программного обеспечения	Описание	Реквизиты подтверждающих документов
1	Браузер Chrome	Система поиска информации в сети интернет	Свободная лицензия Неискл. право. Бессрочно
2	Браузер Firefox	Система поиска информации в сети интернет	Свободная лицензия Неискл. право. Бессрочно
3	OpenOffice	Пакет офисных приложений	Свободная лицензия Неискл. право. Бессрочно
4	LMS Moodle	ПО для эффективного онлайн-взаимодействия преподавателя и студента	Свободная лицензия Неискл. право. Бессрочно

6. Материально-техническое обеспечение дисциплины

№ п/п	Вид учебной работы	Наименование специальных помещений и помещений для СРС	Оснащенность специальных помещений и помещений для СРС
1	Лекционные занятия	Учебная аудитория для проведения занятий лекционного типа	доска аудиторная, акустическая система, проектор, усилитель-микшер для систем громкой связи, экран, микрофон, миникомпьютер, монитор

2	Лабораторные работы	Учебная лаборатория	доска аудиторная, персональный компьютер (25 шт.)
3	Самостоятельная работа обучающегося	Компьютерный класс с выходом в Интернет В-600а	моноблок (30 шт.), система виденаблюдения (6 видеокамер), проектор, экран
		Читальный зал библиотеки	специализированная мебель, компьютерная техника с возможностью выхода в Интернет и обеспечением доступа в ЭИОС, мультимедийный проектор, экран, программное обеспечение

7. Особенности организации образовательной деятельности для лиц с ограниченными возможностями здоровья и инвалидов

Лица с ограниченными возможностями здоровья (ОВЗ) и инвалиды имеют возможность беспрепятственно перемещаться из одного учебно-лабораторного корпуса в другой, подняться на все этажи учебно-лабораторных корпусов, заниматься в учебных и иных помещениях с учетом особенностей психофизического развития и состояния здоровья.

Для обучения лиц с ОВЗ и инвалидов, имеющих нарушения опорно-двигательного аппарата, обеспечены условия беспрепятственного доступа во все учебные помещения. Информация о специальных условиях, созданных для обучающихся с ОВЗ и инвалидов, размещена на сайте университета www//kgeu.ru. Имеется возможность оказания технической помощи ассистентом, а также услуг сурдопереводчиков и тифлосурдопереводчиков.

Для адаптации к восприятию лицами с ОВЗ и инвалидами с нарушенным слухом справочного, учебного материала по дисциплине обеспечиваются следующие условия:

- для лучшей ориентации в аудитории, применяются сигналы оповещения о начале и конце занятия (слово «звонок» пишется на доске);
- внимание слабослышащего обучающегося привлекается педагогом жестом (на плечо кладется рука, осуществляется нерезкое похлопывание);
- разговаривая с обучающимся, педагогический работник смотрит на него, говорит ясно, короткими предложениями, обеспечивая возможность чтения по губам.

Компенсация затруднений речевого и интеллектуального развития слабослышащих обучающихся проводится путем:

- использования схем, диаграмм, рисунков, компьютерных презентаций с гиперссылками, комментирующими отдельные компоненты изображения;
- регулярного применения упражнений на графическое выделение существенных признаков предметов и явлений;
- обеспечения возможности для обучающегося получить адресную

консультацию по электронной почте по мере необходимости.

Для адаптации к восприятию лицами с ОВЗ и инвалидами с нарушениями зрения справочного, учебного, просветительского материала, предусмотренного образовательной программой по выбранному направлению подготовки, обеспечиваются следующие условия:

- ведется адаптация официального сайта в сети Интернет с учетом особых потребностей инвалидов по зрению, обеспечивается наличие крупношрифтовой справочной информации о расписании учебных занятий;
- педагогический работник, его собеседник (при необходимости), присутствующие на занятии, представляются обучающимся, при этом каждый раз называется тот, к кому педагогический работник обращается;
- действия, жесты, перемещения педагогического работника коротко и ясно комментируются;
- печатная информация предоставляется крупным шрифтом (от 18 пунктов), тотально озвучивается;
- обеспечивается необходимый уровень освещенности помещений;
- предоставляется возможность использовать компьютеры во время занятий и право записи объяснений на диктофон (по желанию обучающихся).

Форма проведения текущей и промежуточной аттестации для обучающихся с ОВЗ и инвалидов определяется педагогическим работником в соответствии с учебным планом. При необходимости обучающемуся с ОВЗ, инвалиду с учетом их индивидуальных психофизических особенностей дается возможность пройти промежуточную аттестацию устно, письменно на бумаге, письменно на компьютере, в форме тестирования и т.п., либо предоставляется дополнительное время для подготовки ответа.

8. Методические рекомендации для преподавателей по организации воспитательной работы с обучающимися.

Методическое обеспечение процесса воспитания обучающихся выступает одним из определяющих факторов высокого качества образования. Преподаватель вуза, демонстрируя высокий профессионализм, эрудицию, четкую гражданскую позицию, самодисциплину, творческий подход в решении профессиональных задач, в ходе образовательного процесса способствует формированию гармоничной личности.

При реализации дисциплины преподаватель может использовать следующие методы воспитательной работы:

- методы формирования сознания личности (беседа, диспут, внушение, инструктаж, контроль, объяснение, пример, самоконтроль, рассказ, совет, убеждение и др.);
- методы организации деятельности и формирования опыта поведения (задание, общественное мнение, педагогическое требование, поручение, приучение, создание воспитывающих ситуаций, тренинг, упражнение, и др.);
- методы мотивации деятельности и поведения (одобрение, поощрение социальной активности, порицание, создание ситуаций успеха, создание ситуаций для эмоционально-нравственных переживаний, соревнование и др.)

При реализации дисциплины преподаватель должен учитывать следующие направления воспитательной деятельности:

Гражданское и патриотическое воспитание:

- формирование у обучающихся целостного мировоззрения, российской идентичности, уважения к своей семье, обществу, государству, принятым в семье и обществе духовно-нравственным и социокультурным ценностям, к национальному, культурному и историческому наследию, формирование стремления к его сохранению и развитию;
- формирование у обучающихся активной гражданской позиции, основанной на традиционных культурных, духовных и нравственных ценностях российского общества, для повышения способности ответственно реализовывать свои конституционные права и обязанности;
- развитие правовой и политической культуры обучающихся, расширение конструктивного участия в принятии решений, затрагивающих их права и интересы, в том числе в различных формах самоорганизации, самоуправления, общественно-значимой деятельности;
- формирование мотивов, нравственных и смысловых установок личности, позволяющих противостоять экстремизму, ксенофобии, дискриминации по социальным, религиозным, расовым, национальным признакам, межэтнической и межконфессиональной нетерпимости, другим негативным социальным явлениям.

Духовно-нравственное воспитание:

- воспитание чувства достоинства, чести и честности, совестливости, уважения к родителям, учителям, людям старшего поколения;
- формирование принципов коллективизма и солидарности, духа милосердия и сострадания, привычки заботиться о людях, находящихся в трудной жизненной ситуации;
- формирование солидарности и чувства социальной ответственности по отношению к людям с ограниченными возможностями здоровья, преодоление психологических барьеров по отношению к людям с ограниченными возможностями;
- формирование эмоционально насыщенного и духовно возвышенного отношения к миру, способности и умения передавать другим свой эстетический опыт.

Культурно-просветительское воспитание:

- формирование эстетической картины мира;

- формирование уважения к культурным ценностям родного города, края, страны;
- повышение познавательной активности обучающихся.

Научно-образовательное воспитание:

- формирование у обучающихся научного мировоззрения;
- формирование умения получать знания;
- формирование навыков анализа и синтеза информации, в том числе в профессиональной области.

Вносимые изменения и утверждения на новый учебный год

№ п/п	№ раздела внесения изменений	Дата внесения изменений	Содержание изменений	«Согласовано» Зав. каф. реализующей дисциплину	«Согласовано» председатель УМК института (факультета), в состав которого входит выпускающая
1	2	3	4	5	6
1					
2					
3					

Приложение к рабочей программе дисциплины

К Г Э У

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение высшего образования
«КАЗАНСКИЙ ГОСУДАРСТВЕННЫЙ ЭНЕРГЕТИЧЕСКИЙ УНИВЕРСИТЕТ»
(ФГБОУ ВО «КГЭУ»)

ОЦЕНОЧНЫЕ МАТЕРИАЛЫ
по дисциплине

Б1.В.13 Рекомендательные системы

(Наименование дисциплины в соответствии с учебным планом)

г. Казань, 2023

Оценочные материалы по дисциплине, предназначенны для оценивания результатов обучения на соответствие индикаторам достижения компетенций.

Оценивание результатов обучения по дисциплине осуществляется в рамках текущего контроля (ТК) и промежуточной аттестации, проводимых по балльно-рейтинговой системе (БРС).

1.Технологическая карта

Наименование раздела	Формы и вид контроля	Рейтинговые показатели							
		I текущий контроль	Дополнительные баллы к ТК1	II текущий контроль	Дополнительные баллы к ТК2	III текущий контроль	Дополнительные баллы к ТК3	Итого	Промежуточная аттестация
7 семестр									
Раздел 1. « Введение в рекомендательные системы**»**	**ТК1**	**15**						**15**	**15**
Защита лабораторной работы 1		15							
Раздел 2 Коллаборативная (совместная) фильтрация	**ТК2**			**20**				**20**	**20**
Тест или письменный опрос				5					
Защита лабораторной работы 2				15					
Раздел 3 Фильтрация на основе контента	**ТК3**					**20**		**20**	**20**
Тест (раздел 3)						5			
Защита лабораторной работы 3						15			
Промежуточная аттестация (зачет)									**0-45**
Тест									**0-45**
8 семестр									
Раздел 4 Гибридные рекомендательные системы	**ТК1**	**15**						**15**	**15**
Тест (раздел 4)		5							
Защита лабораторной работы 4		10							
Раздел 5. « Контекстно-зависимые рекомендательные системы **»**	**ТК2**			**15**				**15**	**15**
Тест (раздел 5)				5					
Защита лабораторной работы 5				10					
Раздел 6 «Оценка рекомендательных систем**»**	**ТК3**					**20**		**25**	**25**
Тест (раздел 6)						5			
Защита лабораторной работы 6						20			
Промежуточная аттестация (экзамен)									0-45
В форме теста									0-45

2. Оценочные материалы текущего контроля и промежуточной аттестации

Шкала оценки результатов обучения по дисциплине:

Код компе-тенции	Код индикатора компетенции	Заплани-рованные результаты обучения по дисциплине	Уровень сформированности индикатора компетенции			
			Высокий	Средний	Ниже среднего	Низкий
			от 85 до 100	от 70 до 84	от 55 до 69	от 0 до 54
			Шкала оценивания			
			отлично	хорошо	удовлет-ворительно	неудов-летвори-тельно
			зачтено			не зачтено
ПК-2	ПК-2.1	знать:.				
		основы теории рекомендательных систем и принципы систем поддержки принятия решений	Глубокое понимание основных моделей и подходов в рекомендательных системах.	Базовое понимание методов и принципов в поддержки и принятия решений.	Некоторое знание основных моделей и подходов в рекомендательных системах.	Осведомленность о существовании рекомендательных систем, но отсутствие знаний о их основах.
		уметь:				
		анализировать требования и потребности пользователей, прототипировать системы, а также интегрировать их в существующую инфраструктуру	Глубокое умение анализировать требования и потребности пользователей, проводить детальный анализ и выявлять ключевые факторы.	Навыки прототипирования систем, способность создавать основные прототипы, удовлетворяющие основным требованиям пользователей.	Некоторое умение анализировать требования и потребности пользователей, способность провести поверхностный анализ и выявить некоторые факторы.	Осведомленность о необходимости интеграции системы в существующую инфраструктуру, но отсутствие практического опыта и навыков интеграции.
		владеть:.				
		программированием,	Глубокие навыки	Навыки разработк	Основное знание	Осведом ленность

		способностью разрабатывать и тестировать модели рекомендательных систем, а также знать методы анализа данных и машинного обучения	программирования, умение разрабатывать программный код высокого качества.	и и тестирования моделей рекомендательных систем, способность создавать базовые модели и проводить их тестирование.	методов анализа данных и машинного обучения, способность применять их базовые принципы для работы с моделями рекомендательных систем.	о разработке и тестировании моделей рекомендательных систем, но отсутствие практического опыта и навыков работы с ними.
	ПК-2.2	знать:				
		основы алгоритмов обработки изображений и компьютерного зрения.	Глубокое знание основных алгоритмов обработки изображений, включая фильтрацию, сегментацию и детекцию объектов.	Базовое понимание основ компьютерного зрения, включая основные методы распознавания образов и классификации.	Некоторое понимание основ компьютерного зрения, включая базовые методы распознавания образов и классификации.	Осведомленность о существовании алгоритмов обработки изображений, но отсутствие знаний об их основах и принципах работы.
		уметь:				
		применять методы распознавания образов, классификации и сегментации изображений.	Глубокие навыки применения методов распознавания образов, классификации и сегментации изображений.	Способность использовать и настраивать существующие модели и алгоритмы для распознавания образов, классификации и сегментац	Основные навыки применения методов распознавания образов, классификации и сегментации изображений.	Осведомленность о применении готовых моделей и алгоритмов, но отсутствие опыта и навыков настройки и и использо

				ии изображений.		вания их для нужных задач.
		владеть:				
		программированием на языках, таких как Python или C++, и иметь навыки работы с библиотеками для компьютерного зрения, например, OpenCV.	Глубокие навыки программирования на языках Python или C++, включая умение разрабатывать сложные программы и эффективный код.	Умение использовать библиотеку OpenCV для основных задач компьютерного зрения.	Основные навыки программирования на языках Python или C++, способность создавать простые программы.	Осведомленность о библиотеке OpenCV, но отсутствие практического опыта и навыков ее использования для задач компьютерного зрения.
			зачтено			не зачтено
ПК-8	ПК-8.1	знать:.				
		Основы теории рекомендательных систем, включая различные модели и подходы.	основы теории рекомендательных систем, принципы работы и алгоритмы систем поддержки и принятия решений, основы алгоритмов обработки изображений и компьютерного зрения, методы анализа данных и машинног	основы теории рекомендательных систем, принципы работы и алгоритмы систем поддержки и принятия решений, основы алгоритмов обработки изображений и компьютерного зрения, базовые методы анализа данных и	основы теории рекомендательных систем, основы алгоритмов обработки изображений и компьютерного зрения, базовое знание методов анализа данных и машинного обучения.	осведомленность о теории рекомендательных систем, алгоритмах обработки изображений и компьютерного зрения, но отсутствие практического опыта и пониания.

			о обучения.	машинного обучения.		
		уметь:				
		Проектировать и реализовывать модели и алгоритмы для рекомендательных систем и систем поддержки принятия решений.	Анализировать требования и потребности пользователей и проводить детальный анализ. - Прототипировать системы, учитывая требования пользователей.	Создавать прототипы систем, которые удовлетворяют основным требованиям пользователей. - Осуществлять базовую интеграцию систем в существующую инфраструктуру и проводить тестирование.	Проводить базовый анализ требований и потребностей пользователей. - Создавать базовые прототипы систем, удовлетворяющие основным требованиям пользователей.	Осведомленность о необходимости анализировать требования и потребности пользователей, но отсутствие практических навыков выполнения анализа. - Осведомленность о прототипировании систем, но ограниченные навыки создания прототипов.
		владеть:.				
		Знаниями и опытом работы с методами анализа данных и машинного обучения.	Навыками программирования на языках Python или C++ для разработки высококачественного программного кода. -	Навыками программирования на языках Python или C++ для разработки функционального программного кода. - Работой с	Основными навыками программирования на языках Python или C++ для разработки программного кода. - Работой с	Осведомленность о программировании на языках Python или C++, но отсутствие практического опыта и навыков

			Владением методами и подходами разработки моделей и алгоритмов для рекомендательных систем и систем поддержки и принятия решений.	базовыми моделями и алгоритмами для рекомендательных систем и систем поддержки и принятия решений.	базовыми моделями и алгоритмами для рекомендательных систем и систем поддержки и принятия решений.	разработки программного кода.
	ПК-8.2	знать:				
		Основы алгоритмов обработки изображений, включая методы фильтрации, сегментации и детекции объектов.	основы алгоритмов обработки изображений и компьютерного зрения, методы распознавания образов, классификации и сегментации изображений, основы машинного обучения и методов анализа данных.	основы алгоритмов обработки изображений и компьютерного зрения, базовые методы распознавания образов, классификации и сегментации изображений, базовые знания машинного обучения и методов анализа данных.	основы алгоритмов обработки изображений и компьютерного зрения, основные методы распознавания образов, классификации и сегментации изображений, основы машинного обучения и методов анализа данных.	осведомленность о алгоритмах обработки изображений и компьютерного зрения, методах распознавания образов, классификации и сегментации изображений, но отсутствие практического опыта и глубокого пониания.
		уметь:				
		Разрабатывать и реализовывать модели и	Применять методы обработки и	Применять базовые методы обработк	Применять основные методы	Осведомленность о методах

		алгоритмы для компьютерного зрения.	изображений, компьютерного зрения и машинного обучения для решения сложных задач в области компьютерного зрения.	и изображений, компьютерного зрения и машинного обучения для решения задач в области компьютерного зрения	обработки изображений, компьютерного зрения и машинного обучения для решения простых задач в области компьютерного зрения.	обработки изображений, компьютерного зрения и машинного обучения, но отсутствие практического опыта и навыков их применения.
		владеть:				
		Знаниями и опытом работы с библиотеками для компьютерного зрения, такими как OpenCV, TensorFlow или PyTorch.	Навыками программирования на Python или R для обработки и анализа данных. Знаниями в области фич-инжиниринга и предобработки данных.	Знанием методов оценки и выбора моделей машинного обучения. - Умением визуализировать данные и результаты моделей машинного обучения.	Основными навыками программирования на Python или R для работы с данными и простыми алгоритмами машинного обучения. - Базовым понимание алгоритмов машинного обучения и их использованием для простых задач.	Осведомленность о программировании на Python или R, но отсутствие практического опыта и навыков работы с данными и алгоритмами машинного обучения. - Осведомленность о базовых алгоритмах машинного обучения, но ограниче

						нные навыки и опыт их применения.

+/Оценка **«отлично»** выставляется за выполнение *расчетных работ в семестре; тестовых заданий; глубокое понимание технологических методов расчета норм расхода материалов, полные и содержательные ответы на вопросы билета (теоретическое и практическое задание)*;

Оценка **«хорошо»** выставляется за выполнение *расчетных работ в семестре; тестовых заданий; понимание технологических методов расчета норм расхода материалов, ответы на вопросы билета (теоретическое или практическое задание);*

Оценка **«удовлетворительно»** выставляется за выполнение *расчетных работ в семестре и тестовых заданий;*

Оценка **«неудовлетворительно»** выставляется за слабое и неполное выполнение *расчетных работ в семестре и тестовых заданий*.

3. Перечень оценочных средств

Краткая характеристика оценочных средств, используемых при текущем контроле успеваемости и промежуточной аттестации обучающегося по дисциплине:

Наименование оценочного средства	Краткая характеристика оценочного средства	Описание оценочного средства
Тест (Тест)	Система стандартизированных заданий, позволяющая автоматизировать процедуру измерения уровня знаний и умений обучающегося	Комплект тестовых заданий
Отчет по лабораторной работе (ОЛР)	Выполнение лабораторной работы заканчивается представлением отчета. Результатом выполнения лабораторной работы может быть файл с выполненными заданиями, прикрепленный в электронную среду MOODL или задания, выполненные на лабораторной работе и представленные на проверку преподавателю.	Перечень заданий и вопросов для защиты лабораторной работы, перечень требований к отчету
Контрольная работа (КнтР)	Средство проверки умений применять полученные знания для решения задач определенного типа по теме или разделу	Комплект контрольных заданий по вариантам

4. Перечень контрольных заданий или иные материалы, необходимые для оценки знаний, умений и навыков, характеризующих этапы формирования компетенций в процессе освоения дисциплины

Для текущего контроля ТК1:

Проверяемая компетенция: ПК-2. Способен разрабатывать и тестировать программные компоненты решения задач в системах искусственного интеллекта

ПК-2.1. Разрабатывает приложения систем искусственного интеллекта

ПК-2.2. Проводит тестирование систем искусственного интеллекта

Тест

Вопрос	Варианты ответа
Что такое коллаборативная фильтрация?	Метод обработки и анализа данных, основанный на совместной работе нескольких компьютеров.
	Техника ранжирования объектов на основе сравнения предпочтений пользователей.
	Способ сжатия больших объемов данных для более эффективного хранения и передачи.
Какие методы используются в коллаборативной фильтрации	Content-based и collaborative filtering.
	K-means и Decision Trees.
	Regression и Clustering.
Какую информацию использует коллаборативная фильтрация?	Информацию о содержании объектов.
	Исторические данные о предпочтениях пользователей.
	Характеристики и свойства объектов.
	подходят все варианты
Какие проблемы могут возникнуть при использовании коллаборативной фильтрации?	Холодный старт и проблема рекомендаций популярных объектов.
	Низкая скорость обработки больших объемов данных.
	Необходимость создания сложных математических моделей.
Что такое коллаборативная фильтрация?	Метод анализа данных, основанный на сравнении и предсказании предпочтений пользователей на основе их схожести с другими пользователями.
	Техника шифрования данных совместно несколькими участниками для обеспечения безопасного обмена информацией.
	Метод оптимизации инфраструктуры вычислительных систем совместной работы.
Какие методы используются в коллаборативной фильтрации?	Latent Semantic Analysis и Singular Value Decomposition.
	Decision Trees и Deep Learning.
	K-means и Support Vector Machines.
Какую информацию использует коллаборативная фильтрация?	Исторические данные о действиях пользователей, такие как покупки, оценки или просмотры.
	Метаданные объектов, такие как описание, категории или ключевые слова.
	Глубину воды, силу ветра и другие метеорологические параметры.
Какие проблемы могут возникнуть при использовании коллаборативной фильтрации?	Проблема с т.н. "холодным стартом" новых пользователей или объектов.
	ложность вычислений и требования к высокопроизводительным вычислительным системам.
	Недостаточная экспертиза в области математического моделирования.
Что такое тестирование	Процесс проверки исходного кода программы на наличие

программного обеспечения?	ошибок и недочетов.
	Методика исследования пользовательского опыта для оценки удовлетворенности клиентов программным продуктом.
	Совокупность процедур и деятельностей, направленных на обнаружение ошибок и дефектов в программном обеспечении.
Какие основные виды тестирования существуют?	Функциональное, нагрузочное, совместимостное, уязвимостей.
	Графическое, аудио, видео, текстовое.
	Платформенное, экстремальное, адаптивное, мобильное.

Лабораторная работа 1. Коллаборативная фильтрация на основе сходства по пользователям (user-based) и продуктам (item-based)

```
Словарь кинокритиков и выставленных ими оценок для небольшого набора
данных о фильмах
critics={'Lisa Rose': {'Lady in the Water': 2.5, 'Snakes on a Plane': 3.5,
 'Just My Luck': 3.0, 'Superman Returns': 3.5, 'You, Me and Dupree': 2.5,
 'The Night Listener': 3.0},
'Gene Seymour': {'Lady in the Water': 3.0, 'Snakes on a Plane': 3.5,
 'Just My Luck': 1.5, 'Superman Returns': 5.0, 'The Night Listener': 3.0,
 'You, Me and Dupree': 3.5},
'Michael Phillips': {'Lady in the Water': 2.5, 'Snakes on a Plane': 3.0,
 'Superman Returns': 3.5, 'The Night Listener': 4.0},
'Claudia Puig': {'Snakes on a Plane': 3.5, 'Just My Luck': 3.0,
 'The Night Listener': 4.5, 'Superman Returns': 4.0,
 'You, Me and Dupree': 2.5},
'Mick LaSalle': {'Lady in the Water': 3.0, 'Snakes on a Plane': 4.0,
 'Just My Luck': 2.0, 'Superman Returns': 3.0, 'The Night Listener': 3.0,
 'You, Me and Dupree': 2.0},
'Jack Matthews': {'Lady in the Water': 3.0, 'Snakes on a Plane': 4.0,
 'The Night Listener': 3.0, 'Superman Returns': 5.0, 'You, Me and Dupree':
3.5},
'Toby': {'Snakes on a Plane':4.5,'You, Me and Dupree':1.0,'Superman
Returns':4.0}}
```

```
import numpy as np
A=np.array(critics)
pd.DataFrame(critics)
```

	Lisa Rose	Gene Seymour	Michael Phillips	Claudia Puig	Mick LaSalle	Jack Matthews	Toby
Lady in the Water	2.5	3.0	2.5	NaN	3.0	3.0	NaN
Snakes on a Plane	3.5	3.5	3.0	3.5	4.0	4.0	4.5
Just My Luck	3.0	1.5	NaN	3.0	2.0	NaN	NaN
Superman	3.5	5.0	3.5	4.0	3.0	5.0	4.0
	2.5	3.5	NaN	2.5	2.0	3.5	1.0

Returns

You, Me and Dupree

The Night Listener

```
critics['Toby']
critics['Lisa Rose']['Lady in the Water']
```

Вычисление расстояния Евклида

```
from math import sqrt
sqrt(pow(5-4,2)+pow(4-1,2))
```

Вычисление сходства

```
1/(1+sqrt(pow(5-4.5,2)+pow(5-5,2)))
from numpy import exp
exp(-0.3*sqrt(pow(5-4.5,2)+pow(5-5,2)))
from math import sqrt
```

Возвращает сходство person1 и person2 на основе расстояния

```
def sim_distance(prefs,person1,person2):
```

Получить список предметов, оцененных обоими

```
  si={}
  for item in prefs[person1]:
     if item in prefs[person2]:
       si[item]=1
```

Если нет ни одной общей оценки, вернуть 0

```
  if len(si)==0: return 0
```

Сложить квадраты разностей/

```
  sum_of_squares=sum([pow(prefs[person1][item]-prefs[person2][item],2)
  for item in prefs[person1] if item in prefs[person2]])
  return 1/(1+sum_of_squares)
from math import sqrt
```

Возвращает сходство person1 и person2 на основе расстояния

```
def sim_kernel(prefs,person1,person2,alpha=0.3):
```

Получить список предметов, оцененных обоими

```
  si={}
  for item in prefs[person1]:
    if item in prefs[person2]:
      si[item]=1
```

Если нет ни одной общей оценки, вернуть 0

```
  if len(si)==0: return 0
```

Сложить квадраты разностей

```
  sum_of_squares=sum([pow(prefs[person1][item]-prefs[person2][item],2)
  for item in prefs[person1] if item in prefs[person2]])
  return exp(-alpha*sum_of_squares)


sim_kernel(critics, 'Lisa Rose','Toby')
sim_distance(critics, 'Lisa Rose','Toby')
```

Возвращает коэффициент корреляции Пирсона между p1 и p2

```
def sim_pearson(prefs,p1,p2):
```

Получить список предметов, оцененных обоими

```
  si={}
  for item in prefs[p1]:
    if item in prefs[p2]: si[item]=1
```

Если нет ни одной общей оценки, вернуть 0

```
  if len(si)==0: return 0
```

Количество совместно оцененных фильм

```
  n=len(si)
```

Вычислить сумму всех предпочтений

```
  sum1=sum([prefs[p1][it] for it in si])
  sum2=sum([prefs[p2][it] for it in si])
```

Вычислить сумму квадратов

```
  sum1Sq=sum([pow(prefs[p1][it],2) for it in si])
  sum2Sq=sum([pow(prefs[p2][it],2) for it in si])
```

Вычислить сумму произведений

```
  pSum=sum([prefs[p1][it]*prefs[p2][it] for it in si])
```

```
    Вычислить коэффициент Пирсона
    num=pSum-(sum1*sum2/n)
    den=sqrt((sum1Sq-pow(sum1,2)/n)*(sum2Sq-pow(sum2,2)/n))
    if den==0: return 0
    r=num/den
    return r
sim_pearson(critics,'Lisa Rose','Gene Seymour'), sim_distance(critics, 'Lisa Rose','Gene Seymour')
```

Ранжирование критиков

Возвращает список наилучших соответствий для человека из словаря prefs.

Количество результатов в списке и функция подобия – необязательные параметры.

```
def topMatches(prefs,person,n=5,similarity=sim_pearson):
    scores=[(similarity(prefs,person,other),other)
    for other in prefs if other!=person]
  Отсортировать список по убыванию оценок
    scores.sort( )
    scores.reverse( )
    return scores[0:n]

topMatches(critics,'Toby',n=3, similarity=sim_distance)
topMatches(critics,'Toby',n=3, similarity=sim_kernel)
```

Рекомендация фильмов (User-based подход)

Получить рекомендации для заданного человека, пользуясь взвешенным средним оценок, данных всеми остальными пользователями

```
def getRecommendations(prefs,person,similarity=sim_pearson):
    totals={}
    simSums={}
    for other in prefs:
    сравнивать меня с собой  не нужно
        if other==person: continue
        sim=similarity(prefs,person,other)
```

```
  игнорировать нулевые и отрицательные оценки
    if sim<=0: continue
    for item in prefs[other]:
  оценивать только фильмы, которые я еще не смотрел
      if item not in prefs[person] or prefs[person][item]==0:
  Коэффициент подобия * Оценка
          totals.setdefault(item,0)
          totals[item]+=prefs[other][item]*sim
    Сумма коэффициентов подобия
          simSums.setdefault(item,0)
          simSums[item]+=sim
  Создать нормированный список
  rankings=[(total/simSums[item],item) for item,total in totals.items( )]
  Вернуть отсортированный список
  rankings.sort( )
  rankings.reverse( )
  return rankings

getRecommendations(critics,'Toby')
getRecommendations(critics,'Toby', sim_kernel)
```

Сходство предметов

Как заменить

```
{'Lisa Rose': {'Lady in the Water': 2.5, 'Snakes on a Plane': 3.5},
'Gene Seymour': {'Lady in the Water': 3.0, 'Snakes on a Plane': 3.5}}
```

на

```
{'Lady in the Water':{'Lisa Rose':2.5,'Gene Seymour':3.0},
'Snakes on a Plane':{'Lisa Rose':3.5,'Gene Seymour':3.5}}?
```

```
def transformPrefs(prefs):
  result={}
  for person in prefs:
    for item in prefs[person]:
      result.setdefault(item,{})
  Обменять местами человека и предмет
      result[item][person]=prefs[person][item]
  return result


movies=transformPrefs(critics)
movies
topMatches(movies,'Snakes on a Plane',5, sim_pearson)
getRecommendations(movies,'Lady in the Water', sim_distance)
```

Коллаборативная фильтрация по сходству объектов (Item-based collaborative filtering)

```
def calculateSimilarItems(prefs,n=10):
  Создать словарь, содержащий для каждого образца те образцы, которые
  больше всего похожи на него.
  result={}
  Обратить матрицу предпочтений, чтобы строки соответствовали образцам
  itemPrefs=transformPrefs(prefs)
  c=0
  for item in itemPrefs:
  Обновление состояния для больших наборов данных
    c+=1
    if c%100==0: print("%d / %d" % (c,len(itemPrefs)))
  Найти образцы, максимально похожие на данный
    scores=topMatches(itemPrefs,item,n=n,similarity=sim_distance)
    result[item]=scores
  return result
```

```
itemsim=calculateSimilarItems(critics)
itemsim
```

```
def getRecommendedItems(prefs,itemMatch,user):
  userRatings=prefs[user]
  scores={}
  totalSim={}

  Цикл по образцам, оцененным данным пользователем
  for (item,rating) in userRatings.items():

  Цикл по образцам, похожим на данный
    for (similarity,item2) in itemMatch[item]:
  Пропускаем, если пользователь уже оценивал данный образец
      if item2 in userRatings: continue
    Взвешенная суммы оценок, умноженных на коэффициент подобия
      scores.setdefault(item2,0)
      scores[item2]+=similarity*rating
    Сумма всех коэффициентов подобия
      totalSim.setdefault(item2,0)
      totalSim[item2]+=similarity
      if totalSim[item2]==0: totalSim[item2]=0.0000001 # чтобы избежать деления на ноль
  Делим каждую итоговую оценку на взвешенную сумму, чтобы вычислить среднее
  rankings=[(score/totalSim[item],item) for item,score in scores.items( ) ]

  Возвращает список rankings, отсортированный по убыванию
  rankings.sort( )
  rankings.reverse( )
  return rankings

getRecommendedItems(critics,itemsim,'Toby')
```

Рекомендация на данных MovieLens
Источник: http://grouplens.org/datasets/movielens/

```
def loadMovieLens(path='data/'):
Получить названия фильмов
  movies={}
  for line in open(path+'/u.item'):
    (id,title)=line.split('|')[0:2]
    movies[id]=title
Загрузить данные
  prefs={}
  for line in open(path+'/u.data'):
    (user,movieid,rating,ts)=line.split('\t')
    prefs.setdefault(user,{})
    prefs[user][movies[movieid]]=float(rating)
  return prefs
prefs=loadMovieLens( )
prefs['87']
len(prefs['87'])
getRecommendations(prefs,'87')[0:30]
itemsim=calculateSimilarItems(prefs,n=50)
itemsim["What's Eating Gilbert Grape (1993)"]
getRecommendedItems(prefs,itemsim,'87')[0:30]
```

Для текущего контроля ТК2:

Тест

Вопрос	Варианты ответа
Что такое гибридные рекомендательные системы?	a) Комплексный подход к рекомендациям, включающий использование различных методов и источников данных.
	b) Системы, где пользователи вносят свои предпочтения в процессе формирования рекомендаций.
	c) Технология, использующая только контентную информацию для формирования рекомендаций.
Какие методы сочетаются в гибридных рекомендательных системах?	a) Collaborative filtering и content-based filtering.
	b) K-means и decision trees.
	c) Linear regression и clustering.
Какую информацию используют гибридные	a) Исторические данные о действиях пользователей.
	b) Характеристики и свойства объектов.

рекомендательные системы?	c) Информацию о содержании объектов.
Какие преимущества у гибридных рекомендательных систем?	a) Увеличение точности и разнообразия рекомендаций, учет различных типов данных.
	b) Быстрая скорость обработки данных и меньшие требования к ресурсам.
	c) Простота реализации и отсутствие необходимости в данных о пользователях.
Какие методы можно использовать для объединения рекомендаций в гибридной рекомендательной системе?	a) Среднее арифметическое.
	b) Взвешенное суммирование.
	c) Применение бинарных операций, таких как AND или OR.
Какая основная идея кластеризации пользователей в гибридных рекомендательных системах?	a) Группировка пользователей на основе их социальных связей.
	b) Выделение групп пользователей с похожими предпочтениями и интересами.
	c) Ранжирование пользователей на основе их активности и вклада в систему.
Какую информацию использует коллаборативная фильтрация?	Исторические данные о действиях пользователей, такие как покупки, оценки или просмотры.
	Метаданные объектов, такие как описание, категории или ключевые слова.
	Глубину воды, силу ветра и другие метеорологические параметры.
Какие преимущества предлагает гибридная рекомендательная система по сравнению с отдельными методами (collaborative filtering или content-based filtering)?	a) Увеличение точности и разнообразия рекомендаций.
	b) Большая простота и низкая вычислительная сложность.
	c) Более легкая реализация и интеграция в существующую систему.
Что означает cold start problem в контексте гибридных рекомендательных систем?	a) Проблема с нехваткой данных для формирования рекомендаций новым пользователям или объектам.
	b) Проблема с непредсказуемостью предпочтений пользователей в гибридной системе.
	c) Проблема с высокой ошибкой предсказания рейтингов в гибридных системах.
Какие типы данных обычно используются в гибридных рекомендательных системах?	a) Только текстовые данные.
	b) Только числовые данные.
	c) Комбинация разных типов данных, таких как текст, изображения, видео или контентные характеристики.

Лабораторная работа 2. Матричная факторизация

1. Изучить теоретический материал по ссылке: http://activisiongamescience.github.io/2016/01/11/Implicit-Recommender-Systems-Biased-Matrix-Factorization/

2. Сравнить качество рекомендаций и время выполнения ALS (без базовых предикторов - without biases) и SGD (без базовых предикторов) для различных параметров регуляризации (например, 0,001, 0,01,...) и числа факторов (например, 1, 10, 25, 50,...). Для SGD необходимо усовершенствовать код.
3. Исправить код ALS так, чтобы была возможность производить расчеты с базовыми предикторами (biases).
4. Сравнить все 4 подхода по мере MSE, времени на обучение (training phase) и тестирование (inference phase). Результаты, помимо шагов выполнения в ipython, необходимо представить в сводной таблице.

Источник данных: MovieLens 100k. Может быть выбран альтернативный, но не меньший по числу оценок, либо пользователей и предметов (items).

Загружаем оценки и собираем из них матрицу оценок rij

[ ]

```
# загружаем данные.
names = ['user_id', 'item_id', 'rating', 'timestamp'] df
=   pd.read_csv('u.data',   sep='\t',   names=names)
n_users = df.user_id.unique().shape[0]
n_items = df.item_id.unique().shape[0]

# Создаем r_{ui} - нашу матрицу оценок
ratings = np.zeros((n_users, n_items))
for row in df.itertuples(): ratings[row[1]-
1, row[2]-1] = row[3]
```

Функция, которая делит множество оценок на test и train

[ ]

```
# Делим данные на train и test
# Для каждого пользователя 10 случайных оценок отправляем в test def
train_test_split(ratings):
test = np.zeros(ratings.shape) train =
ratings.copy()
for user in np.arange(ratings.shape[0]):
test_ratings = np.random.choice(ratings[user, :].nonzero()[0], size=10,
replace=False) train[user, test_ratings] =
0.
test[user, test_ratings] = ratings[user, test_ratings] #

Проверяем, что  t est и train не пересекаются
```

```
assert(np.all((train * test) == 0)) return
train, test
```

Метрика качества

```
[ ]
from sklearn.metrics import mean_squared_error

def get_mse(pred, actual):
pred = pred[actual.nonzero()].flatten() actual =
actual[actual.nonzero()].flatten() return
mean_squared_error(pred, actual)
```

Класс, который реализует алгоритм ALS и ExplicitMF с градиентным спуском и смещениями

```
[ ]
from numpy.linalg import solve


    class ExplicitMF():
      def __init__(self,
ratings, n_factors=40,
learning='sgd',
item_fact_reg=0.0,
user_fact_reg=0.0,
item_bias_reg=0.0,
user_bias_reg=0.0,
verbose=False):
"""
Train a matrix factorization model to predict empty entries
in a matrix. The terminology assumes a ratings matrix
which is ~ user x item

Params
======
ratings : (ndarray)
User x Item matrix with corresponding ratings

n_factors : (int)
Number of latent factors to use in matrix factorization
model
learning : (str)
Method of optimization. Options include 'sgd' or 'als'.

item_fact_reg : (float)
Regularization term for item latent factors
```

```
user_fact_reg : (float)
Regularization term for user latent factors

item_bias_reg : (float)
Regularization term for item biases

user_bias_reg : (float)
Regularization term for user biases

verbose : (bool)
Whether or not to printout training progress """

self.ratings = ratings
self.n_users, self.n_items = ratings.shape
self.n_factors = n_factors self.item_fact_reg =
item_fact_reg self.user_fact_reg = user_fact_reg
self.item_bias_reg = item_bias_reg
self.user_bias_reg = user_bias_reg self.learning =
learning
if self.learning == 'sgd':
self.sample_row, self.sample_col = self.ratings.nonzero() self.n_samples =
len(self.sample_row)
self._v = verbose

def als_step(self,
latent_vectors, fixed_vecs,
ratings,
_lambda, type='user'):
"""
One of the two ALS steps. Solve for the latent vectors
specified by type.
"""
        if type == 'user':
            # Precompute
YTY = fixed_vecs.T.dot(fixed_vecs)  lambdaI =
np.eye(YTY.shape[0]) * _lambda

for u in range(latent_vectors.shape[0]):
latent_vectors[u, :] = solve((YTY + lambdaI),
ratings[u, :].dot(fixed_vecs))
        elif type == 'item':
            # Precompute
XTX = fixed_vecs.T.dot(fixed_vecs)  lambdaI =
np.eye(XTX.shape[0]) * _lambda
```

```
for i in range(latent_vectors.shape[0]):
latent_vectors[i, :] = solve((XTX + lambdaI),
ratings[:, i].T.dot(fixed_vecs)) return latent_vectors

def train(self, n_iter=10, learning_rate=0.1):
""" Train model for n_iter iterations from scratch.""" #
initialize latent vectors
self.user_vecs = np.random.normal(scale=1./self.n_factors, size=(self.n_users, self.n_factors)) self.item_vecs =
np.random.normal(scale=1./self.n_factors, size=(self.n_items, self.n_factors))

if self.learning == 'als':
self.partial_train(n_iter,0)
elif self.learning == 'sgd':
self.learning_rate = learning_rate self.user_bias =
np.zeros(self.n_users) self.item_bias =
np.zeros(self.n_items)
self.global_bias = np.mean(self.ratings[np.where(self.ratings != 0)]) self.partial_train(n_iter,0)


def partial_train(self, n_iter, iter_done): """
Train model for n_iter iterations. Can be called
multiple times for further training. """
ctr = 1
while ctr <= n_iter:
if (ctr+iter_done) % 10 == 0 and self._v:
print (f'\tcurrent iteration: {ctr+iter_done}') if
self.learning == 'als':
self.user_vecs = self.als_step(self.user_vecs,
self.item_vecs, self.ratings, self.user_fact_reg,
type='user')
self.item_vecs = self.als_step(self.item_vecs,
self.user_vecs, self.ratings, self.item_fact_reg,
type='item')
elif self.learning == 'sgd':
self.training_indices = np.arange(self.n_samples)
np.random.shuffle(self.training_indices)  self.sgd()
ctr += 1

def sgd(self):
for idx in self.training_indices: u =
self.sample_row[idx]
```

```
i = self.sample_col[idx] prediction =
self.predict(u, i)
e = (self.ratings[u,i] - prediction) # error

# Update biases
self.user_bias[u] += self.learning_rate * (e - self.user_bias_reg * self.user_bias[u]) self.item_bias[i] +=
self.learning_rate * (e - self.item_bias_reg * self.item_bias[i])

#Update latent factors
self.user_vecs[u, :] += self.learning_rate *\
(e * self.item_vecs[i, :] - self.user_fact_reg * self.user_vecs[u,:]) self.item_vecs[i, :] +=
self.learning_rate * \
(e * self.user_vecs[u, :] - self.item_fact_reg * self.item_vecs[i,:]) def predict(self, u, i):
""" Single user and item prediction.""" if
self.learning == 'als':
return self.user_vecs[u, :].dot(self.item_vecs[i, :].T) elif
self.learning == 'sgd':
prediction = self.global_bias + self.user_bias[u] + self.item_bias[i] prediction +=
self.user_vecs[u, :].dot(self.item_vecs[i, :].T)
return prediction

def predict_all(self):
""" Predict ratings for every user and item."""
predictions = np.zeros((self.user_vecs.shape[0],
self.item_vecs.shape[0])) for u in
range(self.user_vecs.shape[0]):
for i in range(self.item_vecs.shape[0]): predictions[u, i]
= self.predict(u, i)

return predictions

def calculate_learning_curve(self, iter_array, test, learning_rate=0.1): """
Keep track of MSE as a function of training iterations.

Params
======
iter_array : (list)
List of numbers of iterations to train for each step of the learning
curve. e.g. [1, 5, 10, 20]
test : (2D ndarray)
Testing dataset (assumed to be user x item). The

function creates two new class attributes:

train_mse : (list)
Training data MSE values for each value of iter_array test_mse :
(list)
Test data MSE values for each value of iter_array
```

```
"""
iter_array.sort()
self.train_mse =[]
self.test_mse = []
iter_diff = 0
for (i, n_iter) in enumerate(iter_array):
if self._v:
print (f'Iteration: {n_iter}') if i == 0:
self.train(n_iter - iter_diff, learning_rate) else:
self.partial_train(n_iter - iter_diff, iter_diff) predictions =

self.predict_all()

self.train_mse += [get_mse(predictions, self.ratings)] self.test_mse +=
[get_mse(predictions, test)]
if self._v:
print (f'MSE train:test: {round(self.train_mse[-1],2)} : {round(self.test_mse[-1],2)}\n') iter_diff = n_iter
```

для построения графиков

[ ]

```
%matplotlib inline
import matplotlib.pyplot as plt
import  seaborn  as  sns sns.set()
  def plot_learning_curve(iter_array, model):
      plt.plot(iter_array, model.train_mse, \
                label='Training', linewidth=3)
plt.plot(iter_array, model.test_mse, \
label='Test', linewidth=3)
plt.xticks(fontsize=16); plt.yticks(fontsize=16);
plt.xlabel('Iterations', fontsize=25);
plt.ylabel('MSE', fontsize=25);
plt.legend(loc='best', fontsize=20);
```

Градиентный спуск с базовыми предикторами (biases)

[ ]

```
train, test = train_test_split(ratings)
MF_SGD = ExplicitMF(train, 40, learning='sgd', verbose=True) iter_array
= [1, 2, 5, 10, 25, 50, 100, 200]
```

[ ]

```
MF_SGD.calculate_learning_curve(iter_array, test, learning_rate=0.001)
```

Iteration: 1
MSE train:test: 1.14 : 1.19

Iteration: 2
MSE train:test: 1.07 : 1.14

Iteration: 5
MSE train:test: 0.98 : 1.06

Iteration: 10
current iteration: 10
MSE train:test: 0.92 : 1.01

Iteration: 25
current iteration: 20
MSE train:test: 0.87 : 0.96

Iteration: 50
current iteration: 30
current iteration: 40
current iteration: 50
MSE train:test: 0.84 : 0.94

Iteration: 100
current iteration: 60
current iteration: 70
current iteration: 80
current iteration: 90
current iteration: 100
MSE train:test: 0.75 : 0.92

Iteration: 200
current iteration: 110
current iteration: 120
current iteration: 130
current iteration: 140
current iteration: 150
current iteration: 160
current iteration: 170
current iteration: 180
current iteration: 190
current iteration: 200
MSE train:test: 0.4 : 0.92

[ ]
plot_learning_curve(iter_array, MF_SGD)

ALS без базовых предикторов

[ ]
MF_ALS = ExplicitMF(train, 40, learning='als', verbose=True) iter_array =
[1, 2, 5, 10, 25, 50, 100, 200]
MF_ALS.calculate_learning_curve(iter_array, test, learning_rate=0.001)
Iteration: 1
MSE train:test: 5.52 : 9.93

Iteration: 2
MSE train:test: 4.21 : 8.66

Iteration: 5
MSE train:test: 3.97 : 8.5

Iteration: 10
current iteration: 10
MSE train:test: 3.93 : 8.48

Iteration: 25
current iteration: 20
MSE train:test: 3.92 : 8.47

Iteration: 50
current iteration: 30
current iteration: 40
current iteration: 50
MSE train:test: 3.92 : 8.48

Iteration: 100
current iteration: 60
current iteration: 70
current iteration: 80
current iteration: 90
current iteration: 100
MSE train:test: 3.92 : 8.48

Iteration: 200
current iteration: 110
current iteration: 120
current iteration: 130
current iteration: 140
current iteration: 150
current iteration: 160
current iteration: 170
current iteration: 180
current iteration: 190
current iteration: 200
MSE train:test: 3.92 : 8.47

Для текущего контроля ТК3:

Тест

Вопрос	Варианты ответа
Что означает понятие контекста в контекстно-зависимых рекомендательных системах?	a) Информация о пользователе, его предпочтениях и интересах.
	b) Информация о времени, месте и ситуации, в которых возникает запрос на рекомендацию.
	c) Информация о связях и взаимодействиях пользователей на платформе.
Какие могут быть источники контекстуальной информации в контекстно-зависимых рекомендательных системах?	a) Геолокация и время суток.
	b) Социальные связи и активность на платформе.
	c) Информация о погоде и трафике.
	d) Все вышеперечисленные источники.
Какие алгоритмы можно использовать для учета контекста в рекомендательной системе?	a) Content-based подходы.
	b) Collaborative filtering методы.
	c) Гибридные алгоритмы.
	d) Все вышеперечисленные алгоритмы
Какая основная идея персонализации рекомендаций в контекстно-зависимых рекомендательных системах?	a) Учитывать контекстуальную информацию для предоставления более точных и релевантных рекомендаций.
	b) Создавать уникальные профили пользователей для каждого контекста.
	c) Анализировать изменения в предпочтениях пользователей в зависимости от контекста.
Что означает понятие точности (precision) в оценке рекомендательных систем?	a) Соотношение релевантных элементов в списке рекомендаций к общему числу рекомендаций.
	b) Соотношение действительно релевантных элементов в списке рекомендаций к числу элементов, которые пользователь оценил.
	c) Соотношение действительно релевантных элементов в списке рекомендаций ко всем релевантным элементам.
Что означает понятие полноты (recall) в оценке рекомендательных систем?	a) Соотношение релевантных элементов в списке рекомендаций к общему числу релевантных элементов.
	b) Соотношение действительно релевантных элементов в списке рекомендаций к числу элементов, которые пользователь оценил.
	c) Соотношение действительно релевантных элементов в списке рекомендаций к общему числу рекомендаций.
Какой показатель наиболее информативен при оценке рекомендательных систем: точность или полнота?	a) Точность.
	b) Полнота.
	c) Оба показателя одинаково важны
Какие метрики оценки качества рекомендательных систем широко используются?	a) Mean Average Precision (MAP).
	b) Mean Reciprocal Rank (MRR).
	c) Normalized Discounted Cumulative Gain (NDCG).
	d) Все вышеперечисленные метрики.
Что означает понятие A/B-тестирования в контексте оценки рекомендательных систем?	a) Разделение пользователей на две случайно составленные группы и сравнение метрик эффективности рекомендаций для каждой группы.
	b) Периодическое изменение алгоритма рекомендаций и

	оценка его влияния на поведение пользователей.
	c) Использование алгоритма Random для генерации рекомендаций и сравнение с другими методами.
Какие методы оценки предсказательной точности рекомендательной системы можно использовать?	a) Holdout метод.
	b) K-fold кросс-валидация.
	c) Leave-one-out (LOO) метод.
	d) Все вышеперечисленные методы.

Лабораторная работа 3. Гибридная система рекомендаций с использованием Python

Приступим к созданию гибридной рекомендательной системы, импортировав необходимые библиотеки Python и набор данных:

```
import pandas as pd
data = pd.read_csv("fashion_products.csv")
print(data.head())
```

Таким образом, эти данные основаны на модных товарах для мужчин, женщин и детей. Наша цель — создать две системы рекомендаций с использованием совместной фильтрации и фильтрации на основе контента, а затем объединить методы рекомендаций для создания системы рекомендаций с использованием гибридного подхода.

Во-первых, давайте импортируем необходимые библиотеки Python, которые мы будем использовать для остальной части задачи:

```
from surprise import Dataset, Reader, SVD
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics.pairwise import linear_kernel
```

В приведенном выше коде я импортировал библиотеку Surprise, которую вы, возможно, раньше не использовали. Библиотека сюрпризов импортирована для использования алгоритма SVD. SVD означает разложение по сингулярным значениям. Проще говоря, это метод матричной факторизации, обычно используемый в алгоритмах совместной фильтрации. Вы можете установить его в своих системах, используя команду, указанную ниже:

- Для терминала или командной строки: pip install scikit-surprise
- Для ноутбука Colab: !pip install scikit-surprise

Фильтрация на основе содержимого

Перейдем к созданию системы рекомендаций с использованием контентной фильтрации:

```
content_df = data[['Product ID', 'Product Name', 'Brand',
                   'Category', 'Color', 'Size']]
content_df['Content'] = content_df.apply(lambda row: '
'.join(row.dropna().astype(str)), axis=1)
```

```
# Use TF-IDF vectorizer to convert content into a matrix of TF-IDF features
tfidf_vectorizer = TfidfVectorizer()
content_matrix = tfidf_vectorizer.fit_transform(content_df['Content'])

content_similarity = linear_kernel(content_matrix, content_matrix)

reader = Reader(rating_scale=(1, 5))
data = Dataset.load_from_df(data[['User ID',
                                  'Product ID',
                                  'Rating']], reader)

def get_content_based_recommendations(product_id, top_n):
    index = content_df[content_df['Product ID'] == product_id].index[0]
    similarity_scores = content_similarity[index]
    similar_indices = similarity_scores.argsort()[::-1][1:top_n + 1]
    recommendations = content_df.loc[similar_indices, 'Product ID'].values
    return recommendations
```

В приведенном выше коде мы реализуем компонент фильтрации на основе контента гибридной рекомендательной системы. Мы начали с выбора соответствующих функций из набора данных, включая идентификатор продукта, название, бренд, категорию, цвет и размер. Затем мы объединили эти функции в один столбец «Контент» для каждого продукта.

Затем мы использовали векторизатор TF-IDF (Term Frequency-Inverse Document Frequency) для преобразования содержимого в матрицу признаков TF-IDF. Эта матрица представляет важность каждого слова в содержании по сравнению со всем корпусом.

Затем мы рассчитали сходство между продуктами на основе их содержания, используя косинусную меру сходства. Эта матрица сходства фиксирует сходство между каждой парой продуктов на основе их содержания.

Чтобы получить рекомендации на основе контента, мы сначала нашли индекс целевого продукта в матрице сходства. Затем мы отсортировали оценки сходства в порядке убывания и выбрали первые N похожих товаров. Наконец, мы вернули идентификаторы рекомендуемых продуктов.

Совместная фильтрация

Теперь давайте перейдем к созданию системы рекомендаций с использованием коллаборативной фильтрации:

```
algo = SVD()
trainset = data.build_full_trainset()
algo.fit(trainset)

def get_collaborative_filtering_recommendations(user_id, top_n):
    testset = trainset.build_anti_testset()
    testset = filter(lambda x: x[0] == user_id, testset)
    predictions = algo.test(testset)
    predictions.sort(key=lambda x: x.est, reverse=True)
```

```
    recommendations = [prediction.iid for prediction in predictions[:top_n]]
    return recommendations
```

В приведенном выше коде мы реализовали компонент совместной фильтрации гибридной рекомендательной системы с использованием алгоритма SVD (Singular Value Decomposition).

Во-первых, мы инициализировали алгоритм SVD и обучили его набору данных. Этот шаг включает в себя декомпозицию матрицы рейтинга пользовательских элементов, чтобы зафиксировать базовые шаблоны и скрытые факторы, определяющие пользовательские предпочтения.

Чтобы сгенерировать рекомендации по совместной фильтрации, мы создали тестовый набор, состоящий из пар «пользователь-элемент», которых не было в обучающем наборе. Мы отфильтровали этот тестовый набор, чтобы включить только элементы, принадлежащие целевому пользователю, указанному user_id.

Затем мы использовали обученную модель SVD для прогнозирования оценок элементов тестового набора. Эти прогнозы представляют собой предполагаемые рейтинги, которые пользователь назначит элементам.

Затем прогнозы сортируются по их оценочным рейтингам в порядке убывания. Мы выбрали первые N элементов с самыми высокими оценками в качестве рекомендаций по совместной фильтрации для пользователя.

Гибридный подход

Объединим методы контентной и коллаборативной фильтрации, чтобы построить рекомендательную систему с использованием гибридного метода:

```
def get_hybrid_recommendations(user_id, product_id, top_n):
    content_based_recommendations =
get_content_based_recommendations(product_id, top_n)
    collaborative_filtering_recommendations =
get_collaborative_filtering_recommendations(user_id, top_n)
    hybrid_recommendations = list(set(content_based_recommendations +
collaborative_filtering_recommendations))
    return hybrid_recommendations[:top_n]
```

В приведенном выше коде мы объединили подходы на основе контента и совместной фильтрации для создания гибридной рекомендательной системы.

Функция get_hybrid_recommendations принимает в качестве входных данных user_id, product_id и желаемое количество рекомендаций top_n.

Во-первых, он вызывает функцию get_content_based_recommendations, чтобы получить список рекомендаций на основе контента для указанного product_id. Эти рекомендации основаны на сходстве характеристик данного продукта с другими продуктами в наборе данных.

Затем он вызывает функцию get_collaborative_filtering_recommendations, чтобы получить список рекомендаций по совместной фильтрации для указанного user_id. Эти рекомендации генерируются путем использования исторических взаимодействий пользователя с элементом и оценки пользовательских предпочтений на основе аналогичного поведения пользователей.

Затем мы объединяем рекомендации по контентной и совместной фильтрации, взяв объединение двух списков. Он гарантирует, что гибридные рекомендации включают рекомендации по контенту и совместной фильтрации на основе пользовательских предпочтений.

Вот как можно использовать нашу гибридную систему рекомендаций, чтобы рекомендовать продукты на основе продукта, который просматривает пользователь:

```
user_id = 6
product_id = 11
top_n = 10
recommendations = get_hybrid_recommendations(user_id, product_id, top_n)

print(f"Hybrid Recommendations for User {user_id} based on Product
{product_id}:")
for i, recommendation in enumerate(recommendations):
    print(f"{i + 1}. Product ID: {recommendation}")
    print(f"{i + 1}. Product ID: {recommendation}")
```

```
Hybrid Recommendations for User 6 based on Product 11:
1. Product ID: 928
1. Product ID: 928
2. Product ID: 131
2. Product ID: 131
3. Product ID: 451
3. Product ID: 451
4. Product ID: 837
4. Product ID: 837
5. Product ID: 875
5. Product ID: 875
6. Product ID: 594
6. Product ID: 594
7. Product ID: 1463
7. Product ID: 1463
8. Product ID: 1688
8. Product ID: 1688
9. Product ID: 601
9. Product ID: 601
10. Product ID: 1566
10. Product ID: 1566
```

Итак, вот как создать гибридную систему рекомендаций с помощью Python. Гибридная система рекомендаций сочетает в себе несколько методов рекомендаций, чтобы предоставить пользователям более точные и разнообразные рекомендации. Он использует сильные стороны различных подходов, таких как совместная фильтрация и фильтрация на основе контента, чтобы преодолеть их ограничения и улучшить процесс рекомендаций.