

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН

Казахский национальный исследовательский технический университет имени
К.И. Сатпаева

Институт информационных и телекоммуникационных технологий

Кафедра кибербезопасность, обработка и хранение информации

Сағын Әлішер Әсетұлы

Разработка экспертной системы на языке программирования Python

ДИПЛОМНЫЙ ПРОЕКТ

Специальность 5В070300 – «Информационные системы»

Алматы 2019

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН

Казахский национальный исследовательский технический университет имени
К.И. Сатпаева

Институт информационных и телекоммуникационных технологий

Кафедра кибербезопасность, обработка и хранение информации

ДОПУЩЕН К ЗАЩИТЕ

Заведующий кафедрой
КБОиХИ,

канд.техн.наук,
ассистент-профессора.

 Н.А. Сейлова

" 15 " 05 2019 г.

ДИПЛОМНЫЙ ПРОЕКТ

На тему: «Разработка экспертной системы на языке программирования Python»

по специальности 5В070300 – «Информационные системы»

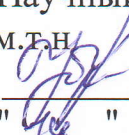
Выполнил

Ә.Ә. Сағын

Рецензент,
заместитель руководителя Национальной
научной лаборатории коллективного
пользования информационных и

Научный руководитель,

М.Т.Н.

 Х. И. Юбузова

" 14 " март 2019 г.

космических технологий НАО КазННТУ
имени К.И.Сатпаева, PhD

К. А. Бостанбеков


Қызметкерлермен
жұмыс бойынша

 март 2019 г.

Алматы 2019

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН

Казахский национальный исследовательский технический университет имени
К.И. Сатпаева

Институт информационных и телекоммуникационных технологий

Кафедра кибербезопасность, обработка и хранение информации

5B070300 – «Информационные системы»

УТВЕРЖДАЮ

Заведующий кафедрой,
КБОиХИ,

канд.техн.наук,
ассистент-профессора.

Н.А. Сейлова
" 15 " 05 2019 г.

ЗАДАНИЕ

на выполнение дипломного проекта

Обучающемуся Сағын Әлішеру Әсетұлы

Тема: Разработка экспертной системы на языке программирования Python.

Утверждена приказом Ректором Университета № 1162-б от 16.10.2018 г.

Срок сдачи законченной работы «16» мая 2019 г.

Исходные данные к дипломному проекту: результат литературного обзора современного состояния по данной теме, сбор теоретического материала.

Краткое содержание дипломного проекта:

- а) анализ и обзор экспертных систем;
- б) модели метода решения для экспертной системы;
- в) информационное и программное обеспечение.

Перечень графического материала (с точным указанием обязательных чертежей): представлены 17 слайдов презентации работы

Рекомендуемая основная литература: из 15 наименований

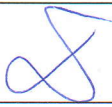
ГРАФИК

подготовки дипломного проекта

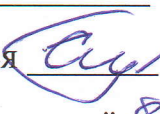
Наименование разделов, перечень разрабатываемых вопросов	Сроки представления научному руководителю и консультантам	Примечание
Анализ и обзор экспертных систем	10.01.2019 – 08.03.2019	
Модели метода решения для экспертной системы	05.02.2019 – 10.03.2019	
Информационное и программное обеспечение	11.03.2019 – 28.04.2019	

Подписи

консультантов и нормоконтролера на законченный дипломный проект с
указанием относящихся к ним разделов работы

Наименование разделов	Научный руководитель, консультанты, И.О.Ф. (уч. степень, звание)	Дата подписания	Подпись
Программное обеспечение	М.Т.Н М.Б. Бауржан	8.05.19г	
Нормоконтролер	Сениор-лектор, PhD О. В. Киселева	8.05.19г	

Научный руководитель  м.т.н Х. И. Юбузова

Задание принял к исполнению обучающийся  О.О. Сағын

Дата " 8 " март 2019 г.

РЕЦЕНЗИЯ

на дипломный проект

Сағын Әлішер Әсетұлы

Специальности 5B070300 Информационные системы

На тему: «Разработка экспертной системы на языке программирования Python»

Выполнено:

- а) графическая часть на 14 листах
б) пояснительная записка на 30 страницах

ЗАМЕЧАНИЯ К РАБОТЕ

В данной дипломной работе представлена экспертная система, которая облегчает пользователю выбор фильма для просмотра, основанная на коэффициенте корреляции Пирсона с оценками кинокритиков. Программное обеспечение реализовано на языке программирования Python с использованием графического фреймворка Qt для пользовательского интерфейса. Дипломный проект состоит из 3 основных разделов. В первом разделе проведен обзор и анализ экспертных систем и постановка задачи. Во втором разделе описана модель представления знаний, а в третьем программная реализация системы.

К данной работе имеются следующие замечания:

- Не проведен подробный анализ имеющихся экспертных систем для решения поставленной задачи.
- Не достаточно развернуто описана программная реализация разработанной системы.

Оценка работы

Тематика дипломной работы является интересной и полезной для любителей кино. Дипломный проект демонстрирует умение и навыки программирования студента на языке Python. Считаю, что дипломный проект Сағын Әлішер на тему: «Разработка экспертной системы на языке программирования Python» заслуживает оценки 95 баллов (Отлично), и присвоения академической степени бакалавра по специальности 5B070300 – Информационные системы.

Рецензент

Заместитель руководителя Национальной
научной лаборатории коллективного пользования

информационных и космических технологий

НАО КазНУ имени К.И. Сатпаева, PhD

К. А. Бостанбеков

Қызметкерлермен
жұмыс бөлімі

2019 г.



ОТЗЫВ

НАУЧНОГО РУКОВОДИТЕЛЯ

на дипломный проект

Сағын Әлішер Әсетұлы

5B070300 Информационные системы

Тема: Разработка экспертной системы на языке программирования Python

Экспертные системы относятся к интеллектуальным информационным системам, предназначенные для массового использования. Такие системы выполняют ряд функций, среди них может быть и функция консультанта, оказывающего помощь в решении той или иной задачи, либо функция эксперта, непосредственно решающего эту задачу. Тема разработки экспертных систем является актуальной и востребованной, т.к. позволяет осуществлять экспертизу, консультацию или выдачу рекомендации в случае невозможности обращения или отсутствия специалиста-эксперта, либо необходимости выполнения экспертизы одновременно в различных местах.

Целью представленного дипломного проекта является: исследование проблем проектирования экспертных систем, разработка экспертной системы по подбору фильма, выполняемого на основе предпочтений других пользователей, просмотревших фильмы, с выдачей рекомендаций.

Для решения поставленной цели дипломник определил алгоритм, выбрал метод решения и модель представления знаний, обосновал выбор средств программной реализации.

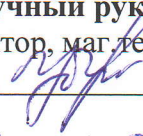
В процессе работы над проектом дипломант проявил творческий подход и показал себя хорошим специалистом, владеющим современными технологиями искусственного интеллекта и экспертных систем.

Дипломник Сағын Ә. Ә. показал хорошее владение навыками самостоятельной работы с технической литературой, самостоятельность в выборе предметной области, инструментария для проектирования экспертных систем.

Считаю, что дипломный проект Сағын Ә. Ә. на тему «Разработка экспертной системы на языке программирования Python» соответствует требованиям учебной программы и может быть допущен к защите.

Научный руководитель

Лектор, маг. техн. наук

 Юбузова Х.И.

«14»  2019 г.

Протокол анализа Отчета подобия

заведующего кафедрой / начальника структурного подразделения

Заведующий кафедрой / начальник структурного подразделения заявляет, что ознакомился(-ась) с Полным отчетом подобия, который был сгенерирован Системой выявления и предотвращения плагиата в отношении работы:

Автор: Сагын Элішер

Название: Разработка экспертной системы на языке программирования Python

Координатор: Халича Юбузова

Коэффициент подобия 1:9,2

Коэффициент подобия 2:0

Тревога:2

После анализа отчета подобия заведующий кафедрой / начальник структурного подразделения констатирует следующее:

- ☒ обнаруженные в работе заимствования являются добросовестными и не обладают признаками плагиата. В связи с чем, работа признается самостоятельной и допускается к защите;
- ☐ обнаруженные в работе заимствования не обладают признаками плагиата, но их чрезмерное количество вызывает сомнения в отношении ценности работы по существу и отсутствием самостоятельности ее автора. В связи с чем, работа должна быть вновь отредактирована с целью ограничения заимствований;
- ☐ обнаруженные в работе заимствования являются недобросовестными и обладают признаками плагиата, или в ней содержатся преднамеренные искажения текста, указывающие на попытки сокрытия недобросовестных заимствований. В связи с чем, работа не допускается к защите.

Обоснование:

Заимствования добросовестные

Дата

Подпись заведующего кафедрой /

начальника структурного подразделения

Халича Юбузова

Окончательное решение в отношении допуска к защите, включая обоснование:

Допускается к защите

Дата 15.05.19

Подпись заведующего кафедрой /

начальника структурного подразделения

КБДХ

Протокол анализа Отчета подобия Научным руководителем

Заявляю, что я ознакомился(-ась) с Полным отчетом подобия, который был сгенерирован Системой выявления и предотвращения плагиата в отношении работы:

Автор: Сагын Әлішер

Название: Разработка экспертной системы на языке программирования Python

Координатор: Халича Юбузова

Коэффициент подобия 1: 9,2

Коэффициент подобия 2: 0

Тревога: 2

После анализа Отчета подобия констатирую следующее:

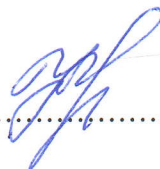
- ☒ обнаруженные в работе заимствования являются добросовестными и не обладают признаками плагиата. В связи с чем, признаю работу самостоятельной и допускаю ее к защите;
- ☐ обнаруженные в работе заимствования не обладают признаками плагиата, но их чрезмерное количество вызывает сомнения в отношении ценности работы по существу и отсутствием самостоятельности ее автора. В связи с чем, работа должна быть вновь отредактирована с целью ограничения заимствований;
- ☐ обнаруженные в работе заимствования являются недобросовестными и обладают признаками плагиата, или в ней содержатся преднамеренные искажения текста, указывающие на попытки сокрытия недобросовестных заимствований. В связи с чем, не допускаю работу к защите.

Обоснование:

обнаруженные замещения внешнего
исследования и не подлежат приложению
приказа

8.05.19

Дата



Подпись Научного руководителя

Краткий отчет



Университет:	Satbayev University
Название:	Разработка экспертной системы на языке программирования Python
Автор:	Сагын Әлішер
Координатор:	Халича Юбузова
Дата отчета:	2019-05-04 15:15:26
Коэффициент подоби́я № 1:	9,2%
Коэффициент подоби́я № 2:	0,0%
Длина фразы для коэффициента подоби́я № 2:	25
Количество слов:	4 059
Число знаков:	30 143
Адреса пропущенные при проверке:	
Количество завершенных проверок:	28



К вашему сведению, некоторые слова в этом документе содержат буквы из других алфавитов. Возможно - это попытка скрыть позаимствованный текст. Документ был проверен путем замещения этих букв латинским эквивалентом. Пожалуйста, уделите особое внимание этим частям отчета. Они выделены соответственно.

Количество выделенных слов 2

>>

Самые длинные фрагменты, определенные, как подобные

№	Название, имя автора или адрес гиперссылки (Название базы данных)	Автор	Количество одинаковых слов
1	URL_ https://www.bibliofond.ru/view.aspx?id=784873		21
2	URL_ https://knowledge.allbest.ru/programming/3c0a65635a2bc69b5d53a88521206c27_0.html		18
3	URL_ http://www.aiuxian.com/article/p-1681480.html		16
4	URL_ https://knowledge.allbest.ru/programming/3c0a65635a2bc69b5d53a88521206c27_0.html		13
5	URL_ https://github.com/arthur-e/Programming-Collective-Intelligence/blob/master/chapter2/recommendations.py		13
6	URL_ https://knowledge.allbest.ru/programming/3c0a65635a2bc69b5d53a88521206c27_0.html		12
7	URL_ https://www.bibliofond.ru/view.aspx?id=784873		12

8	URL_ https://www.bibliofond.ru/view.aspx?id=784873	12
9	URL_ https://knowledge.allbest.ru/programming/3c0a65635a2bc69b5d53a88521206c27_0.html	11
10	URL_ http://www.aiuxian.com/article/p-1681480.html	11

>>

Документы, в которых найдено подобные фрагменты: из RefBooks



Не обнаружено каких-либо

заимствований

>>

Документы, содержащие подобные фрагменты: Из домашней базы данных

Не обнаружено каких-либо заимствований

>>

Документы, содержащие подобные фрагменты: Из внешних баз данных

Документы, выделенные жирным шрифтом, содержат фрагменты потенциального плагиата, то есть превышающие лимит в длине коэффициента подобия № 2

№	Название (Название базы данных)	Автор	Количество одинаковых слов (количество фрагментов)
1	tnpu/Diplomni/Diplomni_2014/14m055/Вишинский Т. l/diplom_vyshynska_2014/diplom_Vyshynska_2014.docx <i>V. Hnatyuk Ternopil National Pedagogic University (TNPU) ()</i>	tnpu	16 (3)

>>

Документы, содержащие подобные фрагменты: Из интернета

Документы, выделенные жирным шрифтом, содержат фрагменты потенциального плагиата, то есть превышающие лимит в длине коэффициента подобия № 2

№	Источник гиперссылки	Количество одинаковых слов (количество фрагментов)
1	URL_ http://www.aiuxian.com/article/p-1681480.html	128 (16)
2	URL_ https://knowledge.allbest.ru/programming/3c0a65635a2bc69b5d53a88521206c27_0.html	82 (8)
3	URL_ https://www.bibliofond.ru/view.aspx?id=784873	77 (7)
4	URL_ https://github.com/arthur-e/Programming-Collective-Intelligence/blob/master/chapter2/recommendations.py	34 (4)
5	URL_ https://github.com/python-pillow/Pillow/issues/2779	18 (2)
6	URL_ http://dataaspirant.com/2015/05/25/collaborative-filtering-recommendation-engine-implementation-in-python/	12 (2)
7	URL_ http://ur-consul.ru/Bibli/CHyelovyechyeskiyi-faktor-v-programmirovanii.html	5 (1)

АҢДАТПА

Менің дипломдық жобамның идеясы фильмді басқа пайдаланушылар мен сыншылардың талғамымен сәйкестендіріп, сәйкестендіру арқылы таңдау. Жобаның тұжырымдамасы фильмдерді іздеу уақытын жеңілдету және қысқарту болып табылады, бұл қазіргі кездегі қиын міндеттердің бірі. Қиындығы бұл - ХХІ ғасырда түрлі жанрдағы 500 мыңға жуық фильм түсірілгені. Қолдану интерфейсі түрлі жастағы пайдаланушылар үшін қарапайым және ыңғайлы.

АННОТАЦИЯ

Данная разработка моего дипломного проекта – это экспертная система для выбора фильма методом отсеивания просмотренных и подбор по схожести вкусов с другими пользователями и критиками. Концепция проекта - это упрощение и укорачивание времени поиска кинофильмов, что является относительно сложной задачей в наше время. Сложность представляется в том, что в 21 веке снято около 500 тысяч фильмов разных жанров. Интерфейс приложения является простым и максимально удобным для пользователей разных возрастов.

THE SUMMARY

The development of my graduation project is an expert system for selecting a film by screening out viewed and matching according to tastes with other users and critics. The concept of the project is to simplify and shorten the time for searching for movies, which is a relatively difficult task in our time. The difficulty is that in the 21st century about 500 thousand films of different genres were shot. The application interface is simple and convenient for users of different ages.

СОДЕРЖАНИЕ

	ВВЕДЕНИЕ	7
1	Анализ и обзор экспертных систем	8
1.1	Постановка цели и задач	8
1.2	Описание предметной области	9
2	Выбор метода решения для экспертной системы	10
2.1	Модель представления знаний	10
2.2	Средства программной реализации	12
2.3	Описание базы знаний	14
3	Информационное и программное обеспечение	17
3.1	Системное программное обеспечение	17
3.2	Инструментальное программное обеспечение	19
3.3	Прикладное программное обеспечение	20
3.4	Программная реализация	21
3.5	Работа и проверка приложения	23
	ЗАКЛЮЧЕНИЕ	26
	СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ	27
	Приложение	28

ВВЕДЕНИЕ

Экспертная система - это интеллектуальная консалтинговая система, основанная на знании целевой области, опыте и знаниях эксперта. Разработка экспертных систем является неотъемлемой частью исследований искусственного интеллекта. Например, создайте экспертную систему, которая поможет решить проблемы с недостатками данных, неясностями и несоответствиями.

База знаний – это специальная база данных, разработанная для оперативных знаний. База знаний содержит структурированную информацию, которая охватывает конкретные области знаний для компьютеров (или отдельных лиц) для конкретных целей.

Как правило, экспертная система включает в себя базу знаний, содержащую накопленный опыт, и механизм логических выводов или правил – набор правил для применения базы знаний к каждой ситуации, которая описывается в программе. Возможности системы могут быть расширены за счет дополнений в базе знаний или в наборе правил. Современные системы могут включать в себя возможности машинного обучения, которые позволяют им улучшать свою производительность на основе опыта, как это делают люди [10].

Концепция экспертных систем была впервые разработана в 1970-х годах Эдвардом Фейгенбаумом, профессором и основателем «Лаборатории систем знаний» в Стэнфордском университете. Фейгенбаум объяснил, что мир перешел от обработки данных к «обработке знаний», переходу, который стал возможен благодаря новым процессорам и компьютерным архитектурам.

Экспертные системы сыграли большую роль во многих отраслях, включая финансовые услуги, телекоммуникации, здравоохранение, обслуживание клиентов, транспорт, видеоигры, производство, авиацию и письменные коммуникации. Две ранние экспертные системы открыли путь для медицинской диагностики в области здравоохранения: Дендрал, который помог химикам идентифицировать органические молекулы, и MYCIN, который помог идентифицировать бактерии, такие как бактериемия и менингит, и рекомендовать антибиотики и дозы [10].

Недавно разработанная экспертная системы ROSS является искусственно-интеллектуальным адвокатом на основе системы когнитивных вычислений IBM Watson. ROSS опирается на самообучающиеся системы, которые используют интеллектуальный анализ данных, распознавание образов, глубокое обучение и обработку естественного языка для имитации работы человеческого мозга.

Экспертные системы и системы искусственного интеллекта развились настолько далеко, что они симулировали дебаты о судьбе человечества перед лицом такого интеллекта, при этом такие авторы, Ник Бостром, профессор философии в Оксфордском университете, размышляя над тем, превзошла ли вычислительная мощность нашу способность контролировать это.

1 Анализ и обзор экспертных систем

Экспертные системы в последнее время становятся популярными и привлекают все больше внимания. Высокое качество работы, достигаемое некоторыми системами в областях, ранее не считавшихся практичными для вычислительных решений, вызвало большой интерес со стороны многих различных дисциплин. Большинство экспертных систем используют подмножество методов из общей области компьютерных исследований, известной как искусственный интеллект. Тем не менее, были разработаны некоторые экспертные системы, которые включают более традиционные методы математического моделирования. Было показано, что сочетание методов искусственного интеллекта и более традиционных математических методов весьма эффективно при разработке нескольких высококачественных компьютерных программных систем. Методы, используемые в экспертных системах, могут быть тем, что необходимо для преодоления разрыва между классическим моделированием оперативных исследований и процессами принятия решений человеком [10].

Можно сказать, что экспертная система в искусственном интеллекте - это компьютерные приложения. Также с помощью этой разработки мы можем решать сложные проблемы. Имеет уровень человеческого интеллекта и опыта. Как правило, он содержит предметно-ориентированные и качественные знания. Коллекция высокоточных и точных знаний является причиной успеха экспертной системы в области искусственного интеллекта.

Что такое данные?

По сути, данные представляют собой совокупность фактов. Также необходимо систематизировать информацию в виде данных и фактов о предметной области. Кроме того, мы можем определить знания как комбинации данных, информации и прошлого опыта.

Возможности экспертной системы:

- консультирование;
- инструктаж и помощь человеку в принятии решения;
- вывод решения;
- диагностирование;
- интерпретация ввода;
- прогнозирование результатов;
- обоснование заключения.

1.1 Постановка цели и задач

Целью данной работы является разработка экспертной системы по подбору фильма, выбор фильма осуществляется за счет предпочтений и уже просмотренных пользователями фильмов, т.е. программа вычисляет, какие фильмы предпочитает пользователь и дает рекомендацию.

Для достижения цели нужно следовать следующему алгоритму:

- создать базу знаний;
- выбрать решение, доказать рациональность решения;
- провести опрос, чтобы убедиться, что в базе достаточно количество фильмов и информации для вывода рекомендации;
- протестировать приложение, внедрить ЭС.

1.2 Описание предметной области

Фильмы - относительно новая форма искусства. Его история очень коротка по сравнению с музыкой, картинами и драмами тысячелетней истории. Однако это не мешает фильму сохранять наиболее распространенную форму искусства на протяжении десятилетий.

На самом деле, хорошие фильмы увлекательны и удивительны, и вы можете думать о том, что происходит в вашей жизни. Как и книга писателя, музыка музыканта, картина художника, фильм режиссера - это способ выразить свои мысли, чувства и желания. Фильмы часто показывают нам идеальную жизнь, жизнь многих людей и жизнь, которую многие люди не могут достичь. Фильмы являются особым мотивом и активно вмешиваются в вашу жизнь, ведя вас к пути к идеалу.

Нет сомнений в том, что фильмы играют важную роль в жизни современных людей. В конце концов, все поколения выросли в фильмах, они сформировались вокруг суровой реальности нашей жизни и имели определенные ценности и идеи.

Фильм формирует взгляд человека на мир, делает его духовно богатым и предоставляет полезную информацию. Под его влиянием фильм также оказывает психологическое воздействие на людей, поскольку может вызывать различные уникальные эмоции. Фильмы и сериалы могут принести улыбки и слезы, счастье и грусть, волнение и душевное спокойствие. Фильмы могут изменить людей, их внутреннее состояние, поведение и личность. Конечно, нельзя отрицать, что фильм негативно влияет на развитие человека.

Взять много фильмов и выбрать один для просмотра - это настоящая проблема.

Чтобы решить эту проблему, будет разработана экспертная система выбора фильмов, которая призвана помочь пользователю быстро выбрать фильм для просмотра в соответствии с заданными условиями и потребностями.

2 Выбор метода решения для экспертной системы

2.1 Модель представления знаний

Модели представления знаний являются одним из важнейших направлений исследований в области искусственного интеллекта. На сегодняшний день разработано достаточное количество моделей. У каждого есть свои сильные и слабые стороны, поэтому вам нужно выбрать собственную модель для каждой конкретной задачи. От этого зависит не только эффективность поставленной задачи, но и возможность ее решения.

В рассматриваемой задаче предметные области представляют собой ряд причинно-следственных связей между внешними факторами и их влиянием на конечный результат.

Для решения поставленной задачи целесообразно использовать метод опроса. Данная модель, основана на конкретных вопросах, позволяет представить знание в виде информации о субъекте и его предпочтениях [6].

Метод опроса - это метод общения с психологическим глаголом, который включает в себя диалог между интервьюером и интервьюируемым (людьми, участвующими в опросе) путем получения ответов на заданные вопросы из темы. Другими словами, опросы - это взаимодействие между интервьюерами и респондентами, а predetermined вопросы являются основным инструментом. Опрос считается одним из наиболее распространенных способов получения информации о субъекте-респонденте. В этом опросе респондентам задавали конкретные вопросы, и ответы предоставили необходимую информацию, основанную на цели исследования. К характеристикам опроса можно отнести их качественные характеристики, обусловленные конкретными деталями задачи, которую они решают.

Для определения сходства двух объектов был выбран R Pearson, поскольку он позволяет определить, насколько пропорциональна изменчивость двух переменных. Определение: коэффициент корреляции Пирсона, также называемый R Пирсона, представляет собой статистический расчет силы взаимосвязи двух переменных. Другими словами, это измерение зависимости двух переменных друг от друга.

Что означает коэффициент корреляции Пирсона?

Коэффициент корреляции произведения Пирсона с моментом указывает степень, в которой изменение одной переменной влияет на другую переменную. Это соотношение измеряется путем расчета наклона линейной регрессии переменных. Значение Person r может принимать только значения в диапазоне от +1 до -1 (оба значения включительно). Если значение r равно нулю, корреляции между переменными нет. Если r больше нуля, существует положительная или прямая корреляция между переменными. Таким образом, уменьшение первой переменной приведет к уменьшению второй переменной. Если r меньше нуля, есть отрицательная или обратная корреляция. Таким образом, уменьшение первой переменной приведет к увеличению второй переменной.

При нанесении на график положительная корреляция будет показывать линию, которая наклонена вниз слева направо, а отрицательная корреляция будет показывать линию, которая наклонена вниз справа налево.

Давайте посмотрим на пример:

Классическим случаем двух переменных, влияющих друг на друга, является спрос и предложение в экономике, когда известны цена продукта и требуемое и поставляемое количество. Значения представлены с использованием простой линейной регрессии.

Pearson R показывает, что спрос и предложение имеют положительную корреляцию. Чем больше потребителей будет нуждаться в товарах, тем больше будет количество поставщиков. Противоположность верна в отношении цены.

По мере того, как цена продукта увеличивается, спрос на этот продукт уменьшается, потому что потребители все меньше хотят покупать продукт по более высоким ценам. С другой стороны, поставщики с большей вероятностью будут производить по более высоким ценам, создавая положительную связь между ценой и предложением. Часто несколько количественных переменных измеряются на каждом члене выборки. Если мы рассмотрим пару таких переменных, часто интересно установить, есть ли отношения между ними, чтобы увидеть, если они взаимосвязаны [9].

Мы можем классифицировать тип корреляции, рассматривая как увеличение одной переменной, в то время как другая переменная:

- положительная корреляция - другая переменная также имеет тенденцию к увеличению;
- отрицательная корреляция - другая переменная имеет тенденцию к снижению;
- нет корреляции - другая переменная не имеет тенденцию увеличиваться или уменьшаться.

Таким образом, отправной точкой любого такого анализа должны быть проектирование и последующие проверки места рассеяния. Примеры отрицательной, «нет» и положительной корреляции на рисунке 2.1.

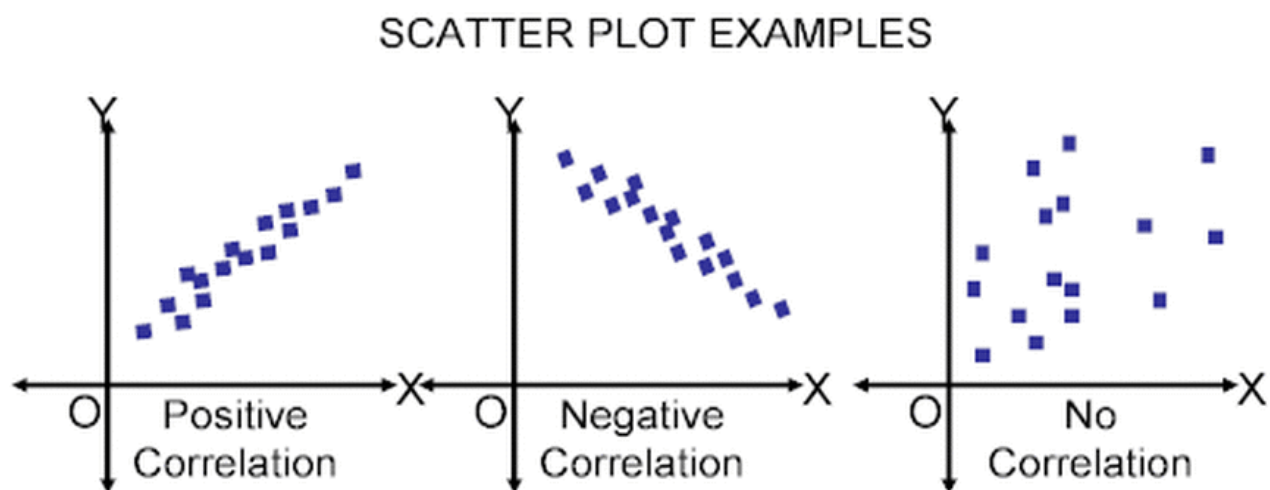


Рисунок 2.1 Виды корреляции

Существует два типа продукционных систем: прямые выводы и обратные выводы. Прямые выводы реализуют стратегию «факт к выводу». В случае обратного вывода можно предположить возможное заключение, которое может быть подтверждено или опровергнуто на основании того, что входит в рабочую память. Некоторые системы имеют двунаправленный выход [5].

Разрабатываемая ЭС основана на прямых выводах, от фактов до выводов. Пользователи выбирают соответствующие факторы, система анализирует их и отображает результаты.

2.2 Средства программной реализации

Реализовать экспертную систему можно с помощью различных программных сред. Каждая функция, которую мы в состоянии выполнить, является результатом команды разработчиков, которые вообразили это, кодировали каждый бит с точностью и заставили это работать. Разработчик пишет код, выполняет его и предоставляет невероятную функциональность программному обеспечению. Это язык, который разработчик использует для инструктажа машины и выполнения ряда действий. Когда у программиста спрашивают, какова важность языка в программировании, не удивляйтесь, если они скажут, что язык - это все, что создает программу и воплощает в жизнь то, что воображается. Есть много языков, которые используются для разных целей. Но есть некоторые языки, которые являются универсальными и имеют очень широкий спектр применения.

Одним из таких языков, который понимают и предпочитают во всем мире для разработки, является Python. Это универсальный интерпретируемый язык на основе ООП высокого уровня, используемый для динамических приложений по всему миру. Python чрезвычайно популярен благодаря своей универсальности и области применения [13]. Python - это язык, охватываемый большим сообществом программистов по многим причинам. Многим предприятиям рекомендуется выбирать Python в качестве языка программирования. Давайте выясним, почему. Это бесплатно, Python - это язык с открытым исходным кодом. Компания Python является одной из крупнейших компаний и до сих пор бесплатная. Использование Python не требует специальной подписки или пользовательской платформы, поэтому любой настольный компьютер и ноутбук совместим с Python. Все инструменты, необходимые для написания кода Python, инструменты, модули и библиотеки полностью бесплатны. Базовая IDE, то есть интегрированные среды разработки, включая PTVS, Pydev с eclipse spyder python, может быть легко загружена бесплатно. Снижение затрат всегда выгодно для бизнеса [12]. Нужно меньше кодирования, Python имеет очень простой синтаксис по своей природе. Та же логика, которая требует 7 строк в языке C++, требует только 3 строки в Python. Наличие меньшего кода требует меньше места, меньше времени и хорошо ценится программистами, так как пересмотр или исправление также занимают меньше времени. Чтобы поддержать себя, язык

имеет много встроенных модулей и библиотек, которые облегчают понимание. Python - широко используемый универсальный язык программирования высокого уровня.

Все виды бизнеса могут себе это позволить. Будучи бесплатной платформой, она может использоваться всеми компаниями малого и среднего уровня. Компании, которые находятся в зачаточном состоянии, могут использовать платформу Python и начать свой бизнес с помощью экономически эффективного программного обеспечения. Способность разрабатывать приложения и программное обеспечение быстро делает его пригодным для стартапов, поскольку они могут выжить в беспощадной конкуренции, используя скорость языка Python. Он подходит не только для малых и средних компаний, но и для таких ведущих компаний, как Google, Spotify, Instagram и Dropbox, а также для разработки на других языках. NASA, Electronic Arts и Disney - одни из ведущих неигровых гигантов, которые переключились на среду Python.

Это один из самых популярных языков. Java и C ++ являются родными объектно-ориентированными языками. Их использование очень широко распространено, а эффективность огромна. Единственная проблема с этими языками в том, что они длинные. Коды громоздки и, следовательно, подкорректировать или переписать - чрезвычайно утомительный процесс. Python, с другой стороны, обладает всеми функциями объектно-ориентированного программирования, такими как Java и C ++, а также работает быстро. Коды короче, а синтаксис прост, поэтому их легко изменить, переработать и оптимизировать. Python для всех, кто занимается разработкой программного обеспечения, каждый программный продукт является результатом серии встроенных программных архитектур, серии строгого кодирования и тестирования. Когда вокруг кода много работы, важно, чтобы язык, на котором написана вся программа, был понятен всем, от разработчика до тестера и все в середине.

Python - это язык, который подходит для всех. Легко понять и проанализировать. Бизнес зависит от программного обеспечения и его функциональности, а программное обеспечение зависит от того, как оно закодировано. Языки программирования помогут вам создать хорошо закодированное программное обеспечение, которое в конечном итоге приведет к хорошей компании. Python - это язык, который делает отличное программное обеспечение. Python обладает всеми функциями, которые делают его настоящим языком, который принимается и ценится во всем мире [14].

Первоначально он был разработан Гвидо ван Россумом в 1991 году и разработан Фондом программного обеспечения Python. Первоначально он был разработан, чтобы подчеркнуть удобочитаемость кода, а его синтаксис позволяет программистам выражать концепции в меньшем количестве строк кода. Python - это язык программирования, который позволяет вам работать быстрее и более эффективно интегрировать системы. Существует две основные версии Python - Python 2 и Python 3. Обе совершенно разные. Итак, давайте рассмотрим некоторые основные различия между Python 2 и Python 3. Python 2 и Python 3

имеют разные (иногда несовместимые) библиотеки. Поскольку Python 3 - это будущее, многие из сегодняшних разработчиков создают библиотеки исключительно для использования с Python 3. Точно так же многие старые библиотеки, созданные для Python 2, не совместимы с форвардами. В Python 3 улучшено целочисленное деление. В Python 2, если вы пишете число без цифр после десятичной точки, оно округляет ваш расчет до ближайшего целого числа. Например, если вы пытаетесь выполнить вычисление 5, деленное на 2, и вводите 5/2, результат будет равен 2 из-за округления. Вы должны были бы написать это как 5.0/2.0, чтобы получить точный ответ 2.5. Однако в Python 3 выражение 5/2 вернет ожидаемый результат 2.5, не беспокоясь о добавлении этих дополнительных нулей. Это один из примеров того, как синтаксис Python 3 может быть более интуитивно понятным, облегчая новичкам изучение программирования на Python. Две версии имеют разные синтаксисы операторов печати. Это только синтаксическое различие - и некоторые могут считать его тривиальным - так что это не влияет на функциональность Python. Тем не менее, это все еще большая и видимая разница, о которой вы должны знать. По сути, в Python 3 оператор print был заменен функцией print (). Например, в Python 2 это печать «привет», но в Python 3 это печать («привет»). Если вы собираетесь изучать программирование на Python впервые, это не должно сильно на вас повлиять. Но если вы начали с Python 2, изменения могут вас несколько раз сбить с толку. Python традиционно не является типизированным языком, но Python v3.5 поддерживает типизацию, что устраняет конфликты разработки при работе с новыми частями кода. В Python 3 текстовые строки по умолчанию являются Unicode. В Python 2 строки хранятся как ASCII по умолчанию - вам нужно добавить «u», если вы хотите хранить строки как Unicode в Python 2.x.

2.3 Описание базы знаний

База знаний - это онлайн-библиотека самообслуживания, содержащая информацию о продукте, услуге, отделе или теме.

Данные в вашей базе знаний могут быть откуда угодно, но обычно поступают от нескольких участников, которые хорошо разбираются в предмете - достаточно, чтобы дать вам все детали. Они варьируются от входов и выходов вашего отдела персонала или юридического отдела до представления нового продукта, оборудования или программного обеспечения. База знаний может включать часто задаваемые вопросы, руководства по устранению неполадок и любые другие подробности, которые вы можете или должны знать. Что касается информационных технологий (ИТ), база знаний - это машиночитаемый ресурс для распространения информации, как правило, в сети или с возможностью размещения в сети. Неотъемлемый компонент систем управления знаниями, база знаний используется для оптимизации сбора, организации и поиска информации для организации или для широкой публики.

Хорошо организованная база знаний может сэкономить корпоративные деньги за счет уменьшения количества времени, затрачиваемого сотрудниками на поиск информации - среди множества возможностей - о налоговых законах или политиках и процедурах компании. Как инструмент управления взаимоотношениями с клиентами (CRM), база знаний может предоставить клиентам легкий доступ к информации, которая в противном случае потребовала бы контакта с персоналом организации, как правило, эта способность должна упростить взаимодействие как для клиента, так и для организации. База знаний может предоставить ценную информацию для клиентов и потенциальных клиентов, это быстрее для ваших клиентов. Вы можете включить знания о каждом отделе, указания и советы по использованию ваших продуктов и услуг, часто задаваемые вопросы и созданный вами контент, который может предоставить всесторонние решения. Доступен ряд программных приложений, которые позволяют пользователям создавать свои собственные базы знаний, либо отдельно (их обычно называют программным обеспечением для управления знаниями), либо в составе другого приложения, такого как пакет CRM [11]. В общем, база знаний - это не статический сбор информации, а динамический ресурс, который сам может обладать способностью к обучению, например, как часть экспертной системы искусственного интеллекта (ИИ). Согласно Консорциуму World Wide Web (W3C), в будущем Интернет может превратиться в обширную и сложную глобальную базу знаний, известную как семантическая сеть.

Метод опроса. Среди различных методов сбора данных для исследовательских целей многие эксперты предпочитают метод обследования из-за его различных преимуществ, преимуществ и преимуществ. Обследования обеспечивают высокий уровень общей способности представлять большинство населения. Обычно из-за огромного количества людей, отвечающих на опрос, собранные данные лучше описывают характеристики населения, участвующего в исследовании.

По сравнению с другими методами сбора данных, обследования могут извлекать данные, близкие к точным признакам большей популяции. При проведении опросов вам нужно только оплатить изготовление анкет. Если вам нужна большая выборка населения в целом, вы можете предоставить денежный или естественный стимул, который может составлять всего 2 доллара на человека. С другой стороны, другие методы сбора данных, такие как фокус-группы и личные интервью, требуют от исследователей платить больше. Опросы могут проводиться для участников различными способами. Анкеты могут быть просто отправлены по электронной почте или факсу, или могут быть отправлены через Интернет. В настоящее время метод онлайн-опроса является наиболее популярным способом сбора данных от целевых участников. В дополнение к удобству сбора данных, исследователи могут собирать данные от людей со всего мира.

На рисунке 2.2 показан пример коллаборативной фильтрации в виде графика, на котором показаны схожесть вкуса двух людей.

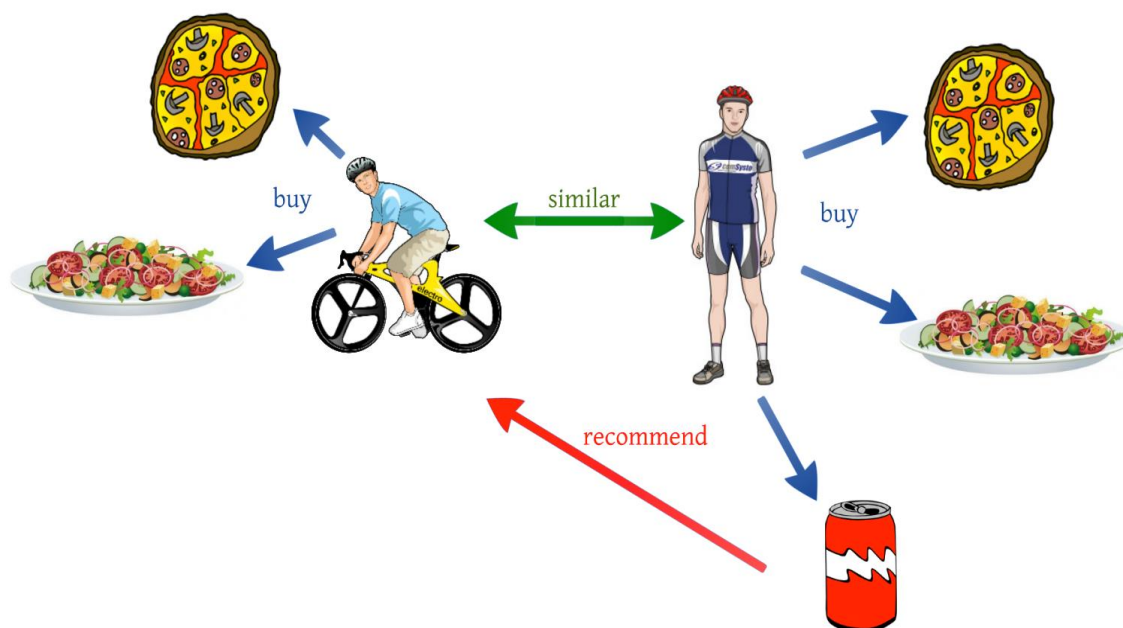


Рисунок 2.2 – Пример коллаборативной фильтрации

Как и многие методы машинного обучения, система рекомендаций делает прогноз на основе исторического поведения пользователей. В частности, это должно предсказать предпочтение пользователя для набора элементов на основе прошлого опыта. Для создания рекомендательной системы наиболее популярны два подхода: контентная фильтрация и коллаборативная фильтрация [1].

Контентно-ориентированный подход требует большого количества информации о собственных функциях элементов, а не использования взаимодействий и отзывов пользователей. Например, это могут быть атрибуты фильма, такие как жанр, год, режиссер, актер и т. д., или текстовое содержимое статей, которые можно извлечь с помощью обработки естественного языка. Коллаборативная фильтрация, с другой стороны, не требует ничего, кроме исторических предпочтений пользователей в отношении набора элементов. Поскольку он основан на исторических данных, основное предположение здесь заключается в том, что пользователи, которые согласились в прошлом, также склонны соглашаться в будущем. С точки зрения пользовательских предпочтений, оно обычно выражается двумя категориями. Явный рейтинг - это рейтинг, который пользователь дает предмету по скользящей шкале, например, 5 звездочек для Титаника. Это самая прямая обратная связь от пользователей, показывающая, насколько им нравится предмет. Неявный рейтинг предлагает косвенные предпочтения пользователей, такие как просмотры страниц, клики, записи о покупках, прослушивание музыкальных композиций или нет и т.д. [3].

3 Информационное и программное обеспечение

3.1 Системное программное обеспечение

Операционная система Microsoft Windows 10. Операционная система - это основное программное обеспечение, которое управляет всем аппаратным и другим программным обеспечением на компьютере. Операционная система, также известная как «ОС», взаимодействует с аппаратным обеспечением компьютера и предоставляет сервисы, которые могут использовать приложения.

Операционная система - это базовое программное обеспечение, установленное на устройстве, которое интегрирует весь контент. Операционная система взаимодействует с оборудованием устройства. Они обрабатывают все: от клавиатур и мышей до беспроводных модулей Wi-Fi, устройств хранения данных и дисплеев. Другими словами, операционная система обрабатывает устройства ввода и вывода. Операционная система связывается с этими устройствами с помощью драйверов устройств, созданных производителями оборудования.

ОС также включает в себя множество общих системных служб, библиотек и программного обеспечения, таких как интерфейсы прикладного программирования (API), которые разработчики могут использовать для создания программ, работающих в операционной системе. ОС расположена между запущенным приложением и оборудованием и использует драйверы оборудования в качестве интерфейса между ними. Например, если ваше приложение хочет что-то напечатать, перенесите эту задачу в операционную систему. Операционная система использует драйвер принтера для отправки инструкций на принтер для отправки правильного сигнала. Приложение печати не должно беспокоить используемый вами принтер или понимать, как оно работает. Детали обрабатываются операционной системой. ОС выполняет многозадачность и распределяет аппаратные ресурсы между несколькими запущенными программами. Если ваш компьютер имеет несколько процессоров или ядер, вы можете запускать несколько процессов параллельно, управляя процессами, которые запускает операционная система, и распределяя их между различными процессорами. Он также управляет внутренней памятью системы и распределяет память между запущенными приложениями.

Windows 10 - это операционная система Microsoft для персональных компьютеров, планшетов, встроенных устройств и интернет-устройств. Любой, использующий Windows 10, может обновить свой старый компьютер непосредственно с Windows 7 или Windows 8 до Windows 10 без необходимости перерисовывать или выполнять громоздкую процедуру очистки и обновления системы. ИТ-специалисты или пользователи могут вручную запустить обновление с помощью Помощника по обновлению Windows или дождаться, пока конфигурация Центра обновления Windows предоставит обновление при запуске. Как и всегда остается возможность откатить версию на более раннюю. Windows 10 имеет встроенные возможности, которые позволяют ИТ-отделам

предприятия использовать программное обеспечение управления мобильными устройствами (MDM) для защиты и управления устройствами под управлением операционных систем. Кроме того, организации могут использовать традиционные программы управления рабочим столом, такие как Microsoft System Center Configuration Manager.

Windows 10 Mobile - это версия операционной системы Microsoft, предназначенная для смартфонов [7].

Особенности Windows 10. Знакомое меню «Пуск», которое Microsoft заменило на Live Tiles в Windows 8, вернулось в Windows 10. Однако пользователи по-прежнему могут получать доступ к Live Tiles и сенсорному центру интерфейса Metro с панели в правой части меню «Пуск».

Microsoft Windows 10 Continuum позволяет пользователям переключаться между интерфейсами сенсорного экрана и клавиатуры на устройствах, которые предлагают оба. Continuum автоматически обнаруживает наличие клавиатуры и ориентирует интерфейс для соответствия.

Интегрированная функция поиска Windows 10 позволяет пользователям выполнять поиск во всех локальных местоположениях, а также в Интернете одновременно.

Microsoft Edge дебютировал с Windows 10 и заменяет Internet Explorer в качестве веб-браузера по умолчанию. Edge включает в себя такие инструменты, как веб-заметки, которые позволяют пользователям разметить веб-сайты, и режим чтения, который позволяет пользователям просматривать определенные веб-сайты без суматохи рекламы. Браузер напрямую интегрируется с Cortana, цифровым помощником Microsoft, который также встроен в Windows 10.

Cortana напрямую интегрируется с поисковой системой Bing и поддерживает как текстовый, так и голосовой ввод. Он отслеживает и анализирует службы определения местоположения, историю общения, электронную почту и текстовые сообщения, персонализацию речи и ввода, службы и приложения, а также историю просмотра и поиска, пытаясь настроить работу ОС в соответствии с потребностями пользователей. ИТ-специалисты могут отключить Cortana и некоторые его функции с помощью параметров групповой политики. Приятная особенность, которая появилась в новой Windows 10 это прокрутка внутри фоновых приложений. Возможность прокручивать неактивные окна при наведении на них, включена в ОС по умолчанию. Теперь, когда вы наведете курсор мыши на приложение, которое не в фокусе, можно прокручивать внутри этого приложения [7].

Безопасность Windows 10. В Microsoft Windows 10 интегрирована поддержка технологий многофакторной аутентификации, таких как смарт-карты и токены. Многие из этих изменений позволят вам улучшить ваше состояние безопасности.

Кроме того, Windows Hello принесла биометрическую аутентификацию в Windows 10, позволяя пользователям входить в систему с помощью технологии сканирования отпечатков пальцев, радужной оболочки глаза или распознавания лиц. Операционная система также включает средства обеспечения безопасности

на основе виртуализации, такие как изолированный пользовательский режим, Защитник устройств Windows Defender и Защитник учетных данных Windows Defender. Эти функции Windows 10 хранят данные, процессы и учетные данные пользователя изолированными в попытке ограничить ущерб от любых атак. Windows 10 также расширила поддержку шифрования BitLocker для защиты данных в движении между пользовательскими устройствами, оборудованием хранения, электронной почтой и облачными сервисами [7].

3.2 Инструментальное программное обеспечение

Для программной реализации был выбран язык программирования Python. Python считался преемником языка ABC в конце 1980-х годов. В Python 2.0, выпущенном в 2000 году, были доступны такие функции, как список циклов ссылок и система сбора мусора. Python 3.0 был выпущен в 2008 году и является базовой версией полностью обратно совместимого языка. Большая часть кода Python 2 не останется прежней в Python 3. Python 2.7 (последняя версия серии 2.x) был продлен до 2020 года. Разработчик языка Гвидо ван Россум взял на себя исключительную ответственность за проект к июлю 2018 года, но в настоящее время является лидером в составе пяти руководящих комитетов.

Многие операционные системы предоставляют интерпретатор Python. Сообщество программистов по всему миру разрабатывает и поддерживает CPython, эталонную реализацию с открытым исходным кодом. Python Software Foundation является некоммерческой организацией, которая управляет ресурсами, разработанными с использованием Python и CPython [15]. CPython является эталонной реализацией языка программирования Python. CPython - это оригинальная реализация Python, который вы, вероятно, будете использовать ежедневно, и который вы можете загрузить официально с python.org. CPython, написанный на C и Python, является наиболее широко используемой реализацией языка. CPython может быть определен как интерпретатор и компилятор, потому что он компилируется в байт-код перед интерпретацией кода Python. Он имеет сторонний интерфейс функций на нескольких языках, включая C, и вы должны явно написать привязку на языке, отличном от Python. Особенностью CPython является то, что он использует глобальную блокировку интерпретатора (GIL) для каждого процесса интерпретатора CPython, что означает, что в пределах одного процесса только один поток может обрабатывать байт-код Python одновременно. Это не означает, что нет смысла в многопоточности, наиболее распространенный многопоточный сценарий - это когда потоки в основном ожидают завершения внешних процессов. Cython - это компилятор, который позволяет писать расширения C для Python, обычно с целью сделать его более эффективным. В отличие от предыдущих примеров, это не другая реализация: он использует CPython для запуска кода Python. Его можно считать надмножеством Python, так как он содержит все свои функциональные возможности и добавляет дополнительные возможности C поверх него.

3.3 Прикладное программное обеспечение

Для своего приложения я использую следующие прикладные программы: JetBrains PyCharm Community Edition, PyQt5. PyCharm - это специализированная среда разработки Python и Django, предоставляющая широкий спектр необходимых инструментов для разработчиков Python, тесно связанных друг с другом для создания удобной среды для продуктивной разработки Python и веб-разработки. PyCharm доступен в трех редакциях: Professional, Community и Education (Edu). Редакции Community и Edu являются проектами с открытым исходным кодом, они бесплатны, но имеют меньше возможностей. PyCharm Edu предоставляет курсы и помогает вам изучать программирование на Python. Professional издание является коммерческим. Набор инструментов и функций.

PyCharm разработан программистами для программистов и предоставляет все инструменты, необходимые для эффективной разработки Python. PyCharm обеспечивает интеллектуальное завершение кода, проверку кода, быстрое исправление ошибок и быстрое исправление, а также автоматический рефакторинг кода и расширенную навигацию. PyCharm предоставляет отличную поддержку фреймворка для последних фреймворков веб-разработки, таких как Django, Flask, Google App Engine, Pyramid и web2py. PyCharm интегрируется с ноутбуками IPython, имеет интерактивную консоль Python, поддерживает Anaconda и несколько научных пакетов, включая matplotlib и NumPy. В дополнение к Python PyCharm также поддерживает JavaScript, CoffeeScript, TypeScript, Cython, SQL, HTML / CSS, языки шаблонов, Angular JS, Node.js и другие. Большое количество готовых к использованию инструментов из коробки. Встроенный отладчик и выполнение теста. Встроенный терминал Python Profiler интегрируется с базовыми VCS и встроенными инструментами базы данных.

PyQt - это набор привязок Python v2 и v3 для инфраструктуры приложений Qt компании Qt. Он работает на всех платформах, поддерживаемых Qt, включая Windows, OS X, Linux, iOS и Android. PyQt5 поддерживает Qt v5. PyQt4 поддерживает Qt v4 и будет собираться против Qt v5. Привязки реализованы в виде набора модулей Python и содержат более 1000 классов. PyQt4 и Qt v4 больше не поддерживаются, и новые выпуски не будут выпускаться. PyQt5 и Qt v5 настоятельно рекомендуются для всех новых разработок. PyQt имеет двойную лицензию на всех поддерживаемых платформах в соответствии с GNU GPL v3 и коммерческой лицензией Riverbank. В отличие от Qt, PyQt недоступен в рамках LGPL. PyQt не включает в себя копию Qt. Вы должны получить правильно лицензированную копию Qt самостоятельно. Однако предусмотрены двоичные диски версии GPR PyQt5, которые включают в себя копию версии QPL LGPL.

Почему PyQt?

PyQt объединяет кроссплатформенную среду приложений Qt C++ и кроссплатформенный интерпретируемый язык Python. Qt - это больше, чем инструментальный GUI. Он включает в себя абстракции сетевых сокетов, потоков, Unicode, регулярных выражений, баз данных SQL, SVG, OpenGL, XML,

полнофункциональный веб-браузер, справочную систему, мультимедийную среду, а также богатую коллекцию графических виджетов. Классы Qt используют механизм сигнал/слот для связи между объектами, который является безопасным по типу, но слабо связанным, что облегчает создание программных компонентов многократного использования. Qt также включает Qt Designer, дизайнер графического пользовательского интерфейса. PyQt может генерировать код Python из Qt Designer. Также возможно добавить новые элементы управления GUI, написанные на Python, в Qt Designer.

Python - это простой, но мощный объектно-ориентированный язык. Его простота облегчает изучение, но его мощь означает, что можно создавать большие и сложные приложения. Его интерпретируемая природа означает, что программисты на Python очень продуктивны, потому что нет цикла разработки/компиляции/компоновки/запуска. Большая часть возможностей Python заключается в его всеобъемлющем наборе модулей расширения, обеспечивающих широкий спектр функций, включая HTTP-серверы, анализаторы XML, доступ к базе данных, инструменты сжатия данных и, конечно же, графический пользовательский интерфейс. Модули расширения обычно реализуются на Python, C или C++. Используя такие инструменты, как SIP, довольно просто создать модуль расширения, который инкапсулирует существующую библиотеку C или C++. При таком использовании Python может стать «клеем» для создания новых приложений из установленных библиотек [15].

3.4 Программная реализация

Алгоритм решения:

- создать базу данных состоящую из данных критиков и их просмотренных фильмов, т.е. к каждому критику был присвоен их собственный список фильмов и оценки на эти фильмы;
- написать программный код, который вычисляет схожесть каждого критика с пользователем на основе коэффициента корреляции Пирсона. Также сохранять данные о пользователе в базе для дальнейшего поиска соответствий среди других пользователей;
- выявить наилучшую схожесть между критиками и пользователем и дать рекомендацию на основе данных, которые пользователь дал при прохождении опроса и поставленных оценок на конкретные фильмы;
- получить рекомендации для заданного человека, пользуясь взвешенным средним оценок, данных всеми остальными пользователями;
- протестировать приложение, выводить на экран конкретный фильм, который будет рекомендован пользователю на просмотр и соответствующую оценку на этот фильм.

База данных хранилась в файле critics.py. В этой базе хранились данные о критиках вместе с просмотренными ими фильмами. Также при работе с

приложением, при окончании опроса и получения результата выведенный системой, данные о пользователе также сохраняются в этом файле. Сам программный код сперва ищет список фильмов, оцененных обоими людьми, т.е. и критиком, и пользователем, если же нет ни одной общей оценки, то дальше уже исключает этого критика. В противном случае, при получения списка фильма оцененных обоими, вычисляется сумма всех предпочтений. И в итоге вычисляется коэффициент корреляции Пирсона, схожесть между этими двумя субъектами. Ниже приведен коды реализации расчета коэффициента корреляции Пирсона и нахождения суммы коэффициента подобия.

```
# Вычислить коэффициент Пирсона
num = p_sum - (sum1 * sum2 / len(si))
den = sqrt((sum1_sq - pow(sum1, 2) / len(si)) * (sum2_sq - pow(sum2, 2) / len(si)))
if den == 0: return 0

# Нахождения суммы коэффициента подобия.
for other in prefs:
    # сравнивать человека с самим собой же 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
        sim_sums.setdefault(item, 0)
        sim_sums[item] += sim
```

Далее после нахождения схожестей между пользователем и критиком, выводится список наилучших соответствий для пользователя. В конечном итоге пользуясь взвешенным средним оценок, данных всеми остальными пользователями получаем рекомендацию фильма для данного пользователя.

Используемые технические средства. При разработке ПО использовался персональный компьютер со следующими характеристиками:

- процессор Intel® Core™ i5-5200 CPU @ 2.20GHz (4CPUs), ~2.2GHz;
- ОЗУ 4096MB RAM;
- видеокарта NVIDIA GeForce 920M;
- видеокарта Intel® HD Graphics 5500;
- жесткий диск HDD 500GB Toshiba, SATA 6Gbit/s.

Вызов и загрузка. Программа не требует установки, для работы достаточно скопировать папку FinalVer на жесткий диск компьютера. Для начала работы нужно запустить файл main.exe. В операционных системах семейства Windows по умолчанию это осуществляется двойным щелчком левой кнопки

мыши по соответствующей иконки. После запуска main.exe появится окно авторизации, в котором Вы можете зарегистрироваться или же создать нового пользователя. В операционных системах семейства Windows по умолчанию это осуществляется двойным щелчком левой кнопки мыши по соответствующей иконки. Программа работает в интерактивном режиме. Вам необходимо прочитать информацию, отображаемую на экране, выбрать действие или вариант ответа и нажать соответствующую кнопку в форме программы. Нажатие кнопки осуществляется одинарным щелчком по ней левой кнопкой мыши.

3.5 Работа и проверка приложения

Рассмотрим работу приложения «FinalVer» на контрольном примере. При запуске приложения откроется начальное окно, представленное на рисунке 3.2. Для дальнейшего использования приложения нужно зарегистрироваться, либо войти с имеющегося аккаунта.

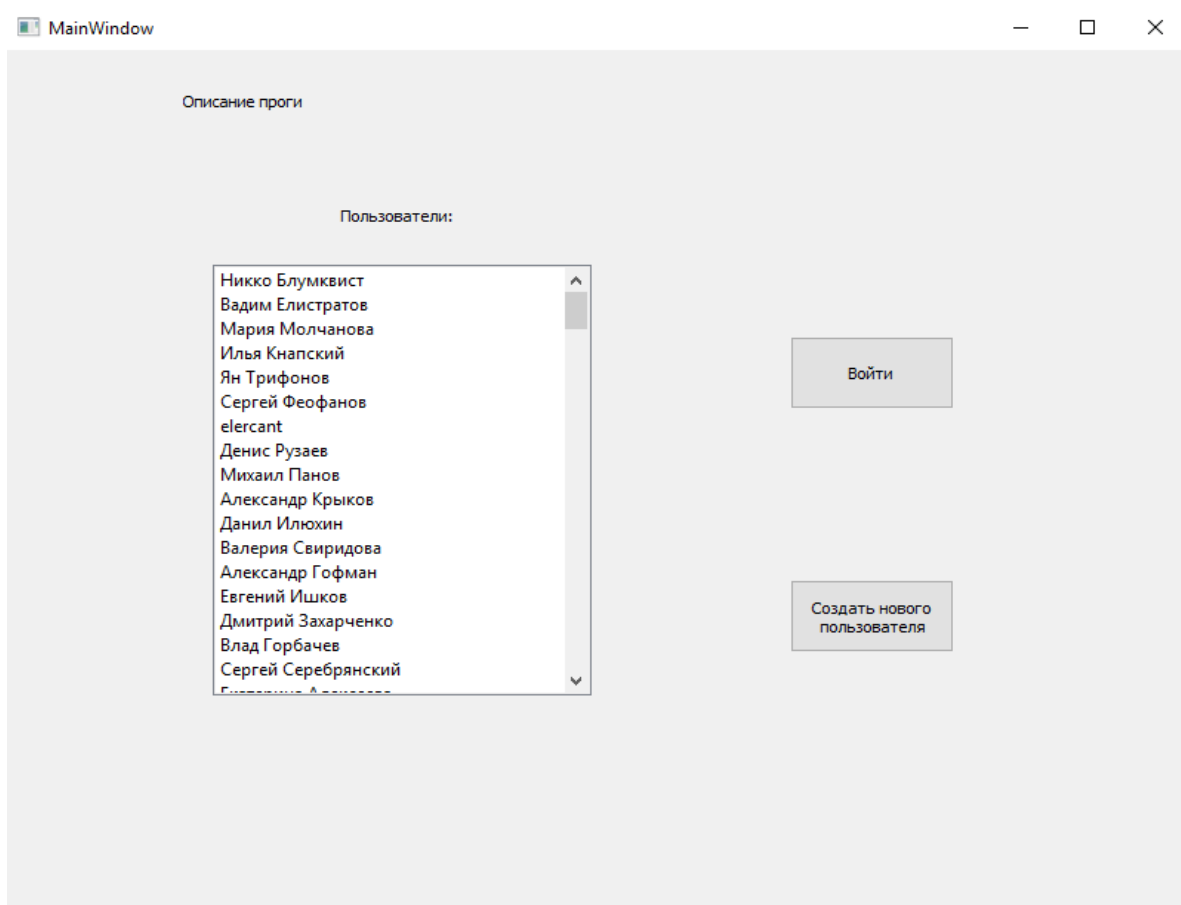


Рисунок 3.2 – Начальная страница

При создании нового пользователя, он сразу же сохраняется, при следующем входе вы просто выберете себя из списка пользователей и войдете в свой аккаунт.

Дальше, как показано на рисунке 3.3 у вас выйдет окно с кнопками «Начать» и «Настройки». После того как вы начали проходить опрос выйдет окно, показанное на рисунке 3.4, в котором представлены сам фильм, окно в котором вы можете оценить данный фильм, либо же вы пропускаете этот фильм, т.к. не смотрели его. И сразу же по окончании опроса, выйдет окно с результатом, т.е. с рекомендуемым Вам фильмом и примерной прогнозируемой оценкой, которую Вы можете выставить этому фильму. Также окно выводит все фильмы, которые вы оценивали в процессе прохождения опроса, Вы можете их сохранить либо удалить, как показано на рисунке 3.5. В настройках Вы можете изменить количество фильмов, участвующих в опросе, и просмотреть весь Ваш список фильмов и оценку на них. Настройки показаны на рисунке 3.6.

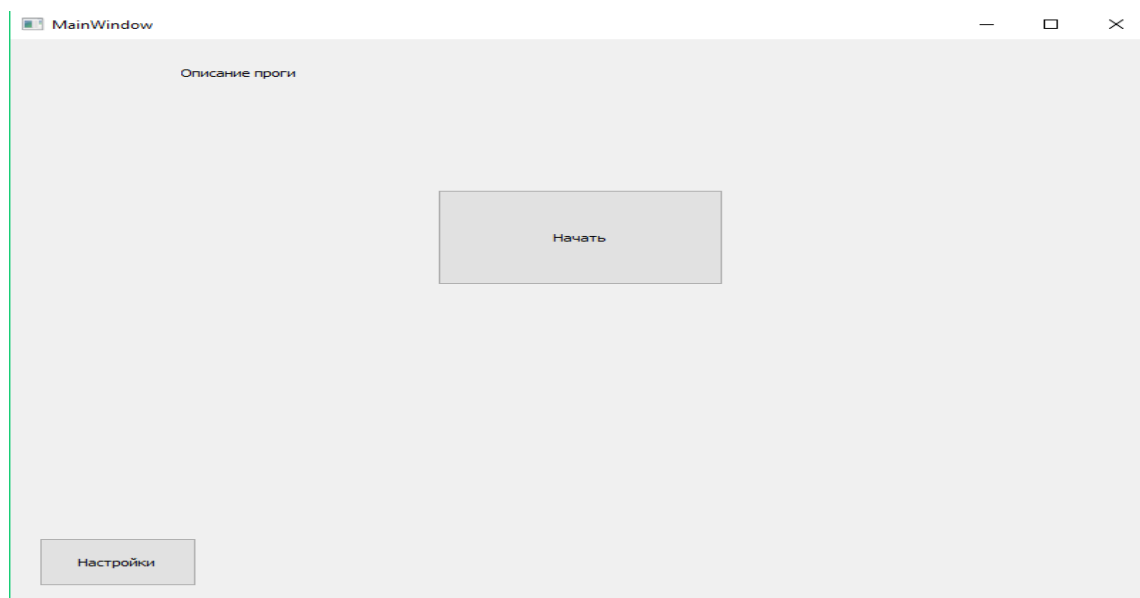


Рисунок 3.3 – Начало приложения

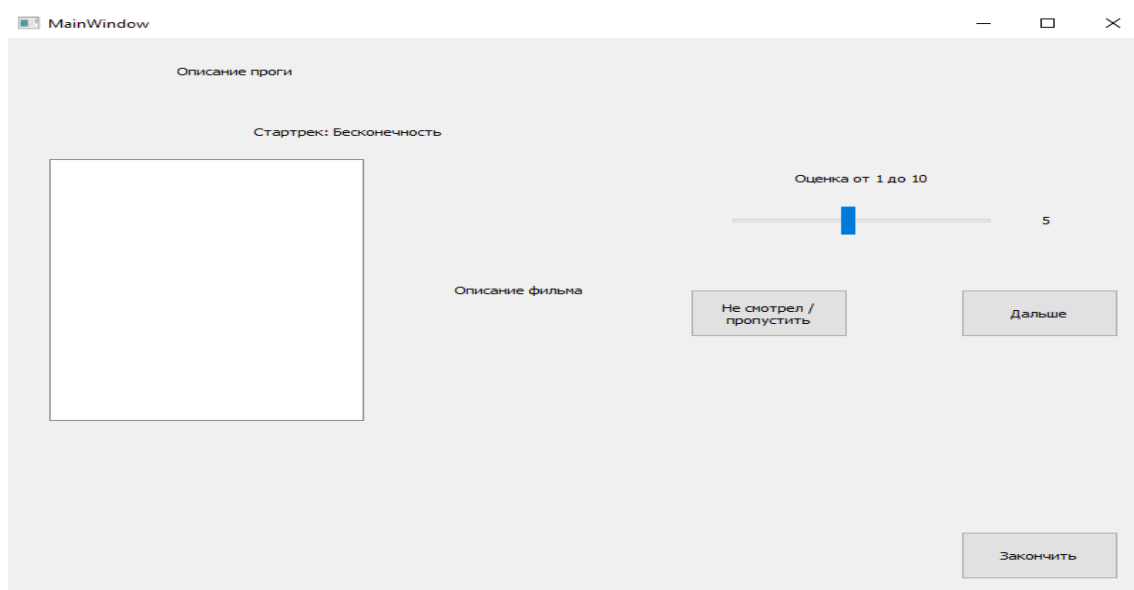


Рисунок 3.4 – Окно опроса

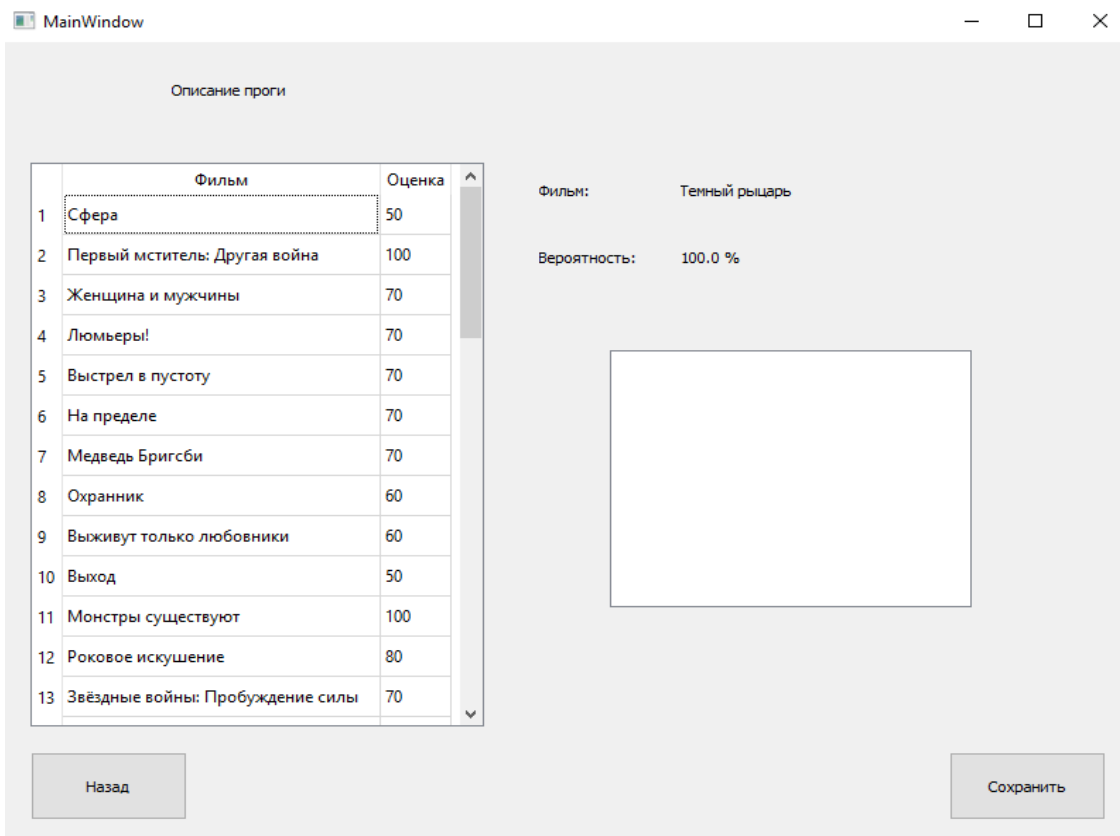


Рисунок 3.5 – Конечный результат с рекомендацией фильма

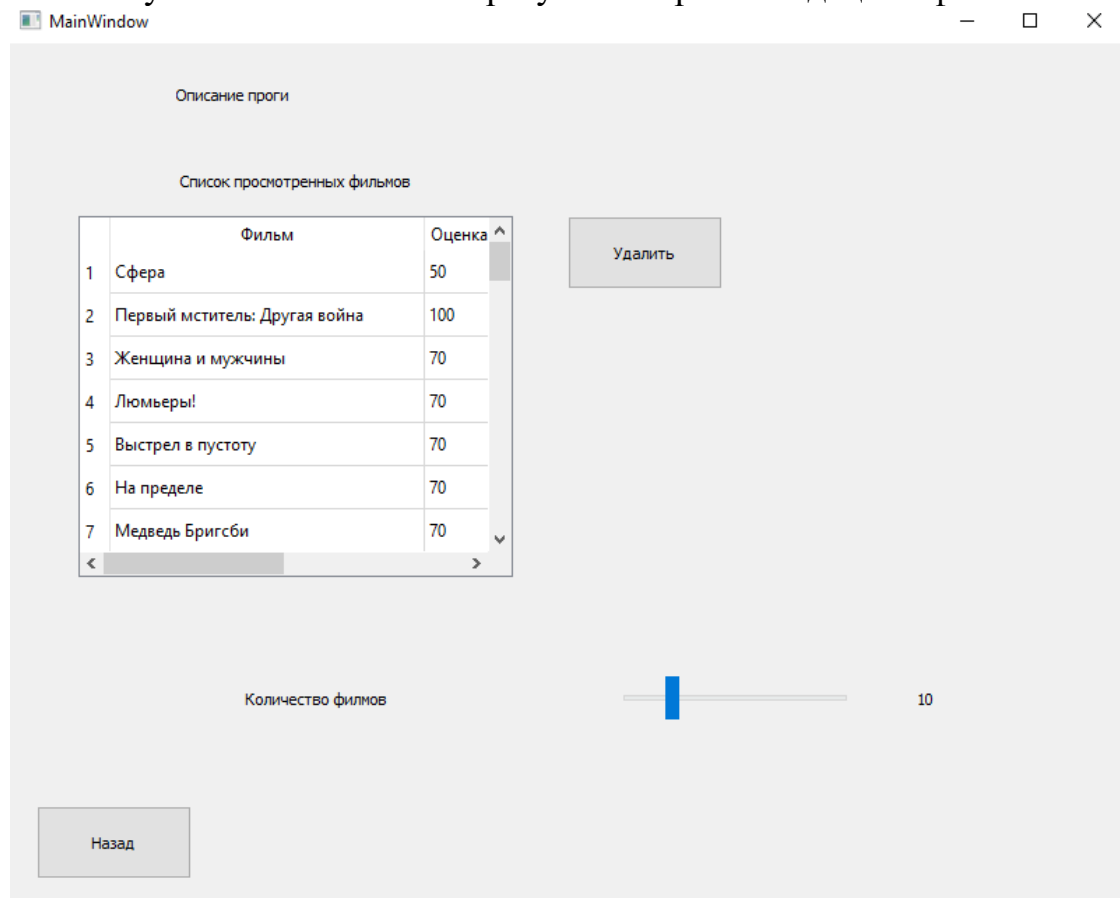


Рисунок 3.6 – Настройки

ЗАКЛЮЧЕНИЕ

В этой работе была создана экспертная система по подбору фильма. Это помогает пользователям найти правильный фильм на основе индивидуальных ответов на вопросы.

В ходе работы были:

- изучена предметная область;
- выбран метод решения задачи;
- выбран язык программирования;
- реализована программа;
- создано руководство пользователя.

Итак, можно сделать вывод, что экспертные системы или вообще рекомендации преследуют нас повсюду, и в обычном общении, и на разных источниках, сайтах, видеохостингах и т.д.

В заключении отметим, что коэффициент корреляции Пирсона является наиболее точным для определения подобия двух объектов.

Из всего сделанного мною, хочу сказать, что работа с экспертной системой очень сложная вещь, много формул, но очень интересная штука.

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

- 1 Алиев, Р.А. Производственные системы с искусственным интеллектом / Р.А. Алиев, Н.М. Абдикеев, М.М. Шахназаров. - М.: Радио и связь, 2016. - 264с.
- 2 Ануфриев Самоучитель Matlab 5.3/6.x / Ануфриев, Игорь. - М.: СПб: БХВ, 2016. - 736 с.
- 3 Васильев, В.И. Распознающие системы / В.И. Васильев. - К.: Наукова думка; Издание 2-е, перераб. и доп., 2016. - 425 с.
- 4 Джордж Основы кибернетики / Джордж, Фрэнк. - М.: Радио и связь, 2016. - 272 с.
- 5 Информационные системы - миф и действительность. - М.: Знание, 2016. – 568 с.
- 6 Методы и средства преобразования информации. - М.: Рига: Зинатне, 2016. - 136 с.
- 7 Мичи, Д. Компьютер - творец / Д. Мичи, Р. Джонстон. - М.: Мир, 2017. - 255 с.
- 8 Мкртчян, С.О. Нейроны и нейронные сети. /С.О. Мкртчян. - М.: Энергия, 2017.-232с.
- 9 Нариньяни, А.С. Моделирование языковой деятельности в интеллектуальных системах / ред. А.Е. Кибрик, А.С. Нариньяни. - М.: Наука. Главная редакция физико-математической литературы, 2012. - 280 с.
- 10 Нейлор, К. Как построить свою экспертную систему. /К. Нейлор. - М.: Энергоатомиздат, 2013. - 286 с.
- 11 Нильсон, Н. Принципы искусственного интеллекта. /Н. Нильсон. - М.: Радио и связь, 2014. - 373 с.
- 12 Доусон М. Програмуємо на Python. – СПб.: Питер, 2014. – 416 с.
- 13 Лутц М. Изучаем Python, 4-е издание. – Пер. с англ. – СПб.: Символ-Плюс, 2011. – 1280 с.
- 14 Лутц М. Программирование на Python, том I, 4-е издание. – Пер. с англ. – СПб.: Символ-Плюс, 2011. – 992 с.
- 15 Лутц М. Программирование на Python, том II, 4-е издание. – Пер. с англ. – СПб.: Символ-Плюс, 2011. – 992 с.

Приложение

Код программы

```
import sys
from PyQt5 import QtWidgets
import design
import random
import pickle
critics1 = {}
films = {}
hist = {}
userH = {}
history = []
rating = []
cUser = ""

from math import sqrt

def sim_pearson(prefs, person1, person2):
    """
    Коэффициент корреляции Пирсона
    :param prefs: набор данных
    :param person1: человек 1
    :param person2: человек 2
    :return: коэффициент корреляции Пирсона
    """

    # 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, то вернуть 0
    if len(si) == 0:
        return 0

    # Вычислить сумму всех предпочтений
    sum1 = sum([prefs[person1][item] for item in si])
    sum2 = sum([prefs[person2][item] for item in si])

    # Вычислить сумму квадратов
    sum1_sq = sum([pow(prefs[person1][item], 2) for item in si])
```

Продолжение приложения

```
sum2_sq = sum([pow(prefs[person2][item], 2) for item in si])

# Вычислить сумму произведений
p_sum = sum([prefs[person1][item]*prefs[person2][item] for item in si])

# Вычислить коэффициент Пирсона
num = p_sum - (sum1 * sum2 / len(si))
den = sqrt((sum1_sq - pow(sum1, 2) / len(si)) * (sum2_sq - pow(sum2, 2) / len(si)))

if den == 0: return 0

# import numpy

# print ("numpy", numpy.corrcoef([prefs[person1][item] for item in si],
[prefs[person2][item] for item in si])[0, 1])
# print ("custom", num/den)

return num/den

def top_matches(prefs, person, n=5, similarity=sim_pearson):
    """
    Возвращает список наилучших соответствий для человека из словаря prefs.
    :param prefs: набор данных
    :param person: человек
    :param n: количество наиболее похожих людей
    :param similarity: функция для вычисления коэффициента подобия
    :return: список наилучших соответствий для человека
    """

    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]

def get_recommendations(prefs, person, similarity=sim_pearson):
    """
```

Продолжение приложения

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

```
:prefs: набор данных
:person: человек
:similarity: функция для вычисления коэффициента подобия
:return: рекомендации
"""
```

```
totals = {}
sim_sums = {}
for other in prefs:
    # сравнивать человека с самим собой же 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
            sim_sums.setdefault(item, 0)
            sim_sums[item] += sim

# Создать нормализованный список
rankings = [(total/sim_sums[item], item)
             for item, total in totals.items()]

# Вернуть отсортированный список
rankings.sort()
rankings.reverse()
return rankings
```

```
class StartApp(QtWidgets.QMainWindow, design.Ui_MainWindow):
    current = 0
    sampleS = 10
    def __init__(self):
        super().__init__()
        self.setupUi(self)
        self.userList.addItem(critics1)
```

Продолжение приложения

```
self.loginButton.clicked.connect(self.login)
self.createUserBtn.clicked.connect(self.createUserPage)
self.createNewUserButton.clicked.connect(self.createNewUser)
self.settBtn.clicked.connect(self.settings)
self.filmListHist.setColumnWidth(0, 225)
self.filmListHist.setColumnWidth(1, 50)
self.removeFilmBtn.clicked.connect(self.delete)
self.sampleSize.valueChanged.connect(self.sliderChangeSet)
self.backButton.clicked.connect(self.backToStart)
self.startButton.clicked.connect(self.start)
self.finishBtn.clicked.connect(self.finish)
self.nextBtn.clicked.connect(self.nextFilm)
self.skipBtn.clicked.connect(self.skip)
self.rateSlider.valueChanged.connect(self.sliderChange)
self.filmList.setColumnWidth(0, 225)
self.filmList.setColumnWidth(1, 50)
self.quitButton.clicked.connect(self.backToStart)
self.exportBtn.clicked.connect(self.save)
```

```
def login(self):
    global cUser
    cUser = next(iter(self.userList.selectedItems())).text()
    print(cUser)
    global hist
    hist = critics1[cUser]
    print(hist)
    self.stackedWidget.setCurrentIndex(2)
```

```
def createUserPage(self):
    self.stackedWidget.setCurrentIndex(1)
```

```
def createNewUser(self):
    global cUser
    cUser = self.userName.text()
    print(cUser)
    self.stackedWidget.setCurrentIndex(2)
```

```
def settings(self):
    p = 0
    for f in hist:
        if self.filmListHist.rowCount() <= p:
            self.filmListHist.insertRow(p)
```

Продолжение приложения

```
self.filmListHist.setItem(p, 0, QtWidgets.QTableWidgetItem(f))
self.filmListHist.setItem(p, 1, QtWidgets.QTableWidgetItem(str(hist[f])))
p += 1
self.stackedWidget.setCurrentIndex(4)

def delete(self):
    i = self.filmListHist.currentRow()
    s = next(iter(self.filmListHist.selectedItems())).text()
    hist.pop(s, None)
    self.filmListHist.removeRow(i)

def sliderChange(self):
    val = self.rateSlider.value()
    self.filmRateNum.setText(str(val))

def sliderChangeSet(self):
    val = self.sampleSize.value()
    self.sampleS = val
    self.sampleSizeNum.setText(str(val))

def start(self):
    self.current = 0
    generateListFilms()
    self.saveQ.setText("")
    self.filmTitle.setText(next(iter(films)))
    self.stackedWidget.update()
    self.stackedWidget.setCurrentIndex(3)

def nextFilm(self):
    addToHistory(self.filmTitle.text(), self.rateSlider.value() * 10)
    self.current += 1
    global films
    if self.current >= self.sampleS: self.finish()
    it = 0
    for film in films:
        it += 1
        if it <= self.current: continue
        self.filmTitle.setText(film)
        break

def skip(self):
    self.current += 1
```

Продолжение приложения

```
it = 0
if self.current >= self.sampleS: self.finish()
global films
for film in films:
    it += 1
    if it <= self.current: continue
    self.filmTitle.setText(film)
    break

def finish(self):
    print(hist)
    userH[cUser] = hist
    critics1.update(userH)
    rec = get_recommendations(critics1, cUser)
    print(rec)
    if len(rec) > 0:
        print(rec[0][1])
        self.recLabel.setText(rec[0][1])
        self.probLabel.setText(str(rec[0][0]) + ' %')
    p = 0
    for f in hist:
        if self.filmList.rowCount() <= p:
            self.filmList.insertRow(p)
        self.filmList.setItem(p, 0, QtWidgets.QTableWidgetItem(f))
        self.filmList.setItem(p, 1, QtWidgets.QTableWidgetItem(str(hist[f])))
        p += 1

    self.stackedWidget.setCurrentIndex(5)

def backToStart(self):
    self.stackedWidget.setCurrentIndex(2)

def save(self):
    if self.saveQ.text() != "Saved!":
        f = open("data.bin", mode="wb")
        pickle.dump(critics1, f)
        f.close()
        self.saveQ.setText("Saved!")

def getsampleS(self):
    return self.sampleS
def generateListFilms():
```

Продолжение приложения

```
global films
films = {}
lc = len(critics1)
x = window.getsampleS()
for i in range(2*x):
    rl = random.randint(1, lc)
    it1 = 0
    for person in critics1:
        it1 += 1
        if it1 < rl: continue
        l = len(critics1[person])
        print(person)
        r = random.randint(1, l)
        it = 0
        for film in critics1[person]:
            it += 1
            if it < r: continue
            films.add(film)
            break
        break
    print(films)
    films.remove("")

def addToHistory(film, rate):
    hist[film] = rate
    history.append(film)
    rating.append(rate)

def main():
    global window
    app = QtWidgets.QApplication(sys.argv)
    window = StartApp()
    #window.stackedWidget.setCurrentIndex(4)
    window.show()
    sys.exit(app.exec_())

if __name__ == '__main__':
    f = open("data.bin", mode="rb")
    critics1 = pickle.load(f)
    f.close()
    main()
```