|  |
| --- |
| МИНОБРНАУКИ РОССИИ |
| Федеральное государственное бюджетное образовательное учреждение высшего образования **«МИРЭА − Российский технологический университет»**  **РТУ МИРЭА** |

**Институт информационных технологий (ИИТ)**

**Кафедра вычислительной техники (ВТ)**

**ОТЧЁТ ПО ПРАКТИКЕ**

**Научно-исследовательская работа**

приказ Университета о направлении на практику от «01» сентября 2021 г.

№ 4087-С

|  |  |  |
| --- | --- | --- |
| Отчет представлен к  рассмотрению:  Студент группы ИКБО-04-18 | «\_\_» декабря 2021 | (подпись и расшифровка подписи) |
| Отчет утвержден.  Допущен к защите: |  |  |
| Руководитель практики  от кафедры | «\_\_» декабря 2021 | (подпись и расшифровка подписи) |
|  |  |  |

Москва 2021 г.

|  |
| --- |
| МИНОБРНАУКИ РОССИИ |
| Федеральное государственное бюджетное образовательное учреждение высшего образования **«МИРЭА − Российский технологический университет»**  **РТУ МИРЭА** |

**Институт информационных технологий (ИИТ)**

**Кафедра вычислительной техники (ВТ)**

**ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ НА ПРАКТИКУ**

**Научно-исследовательская работа**

**Студенту 4 курса учебной группы ИКБО-04-18**

**Зыкову Александру Алексеевичу**

**Место и время практики:** Кафедра вычислительной техники института информационных технологий Московского технологического университета - МИРЭА с 1 сентября 2021г. по 21 декабря 2021г.

**Должность на практике:**  студент

**1. ЦЕЛЕВАЯ УСТАНОВКА:** Исследование выбранной предметной области на предмет осуществления её анализа и актуальности, постановки задачи, реализации под неё технического задания и проектирования на основе структурных и объектных методологий.

**2. СОДЕРЖАНИЕ ПРАКТИКИ:**

2.1. Изучить: научную и техническую литературу, электронные информационно-образовательные ресурсы, применяемые для профессиональной деятельности по теме научно-исследовательской работы.

2.2. Практически выполнить: реализацию технического задания и проектной части с использованием диаграмм структурных и объектных методологий, описать процессы и этапы жизненного цикла выбранного программного обеспечения, согласно условиям стандартизации, охарактеризовать актуальность выбранной предметной области, представить подобные программные разработки и провести анализ предметной области.

2.3. Ознакомиться: с литературой по выбранной предметной области и технической литературой по искусственному интеллекту, а также стандартами и ГОСТами реализации программного обеспечения.

**3. ДОПОЛНИТЕЛЬНОЕ ЗАДАНИЕ:** оформить презентационный материал

**4. ОРГАНИЗАЦИОННО-МЕТОДИЧЕСКИЕ УКАЗАНИЯ:** в процессе практики рекомендуется использовать периодические издания и отраслевую литературу годом издания не старше 5 лет от даты начала прохождения практики

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Руководитель практики от кафедры  «06» сентября 2021 г. | | | *Подпись* | | (Сорокин А. Б.) |
| Задание получил | |  |  | |  |
| «06» сентября 2021 г. | |  | *Подпись* | | (Зыков А.А.) |
|  | |  |  | |  |
|  | |  |  | |  |
| **СОГЛАСОВАНО:** | |  |  | |  |
| Заведующий кафедрой: | | | | | |
| «06» сентября 2021 г. | | | *Подпись* | | (Платонова О.В.) |
| **Проведенные инструктажи:** |  | | |  | |
| Охрана труда: |  | | | «06» сентября 2021 г. | |
| Инструктирующий | *Подпись* | | | Сорокин А. Б. доцент кафедры ВТ | |
| Инструктируемый | *Подпись* | | | Зыков А.А. | |
|  |  | | |  | |
| Техника безопасности: |  | | | «06» сентября 2021 г. | |
| Инструктирующий | *Подпись* | | | Сорокин А. Б. доцент кафедры ВТ | |
| Инструктируемый | *Подпись* | | | Зыков А.А. | |
|  |  | | |  | |
| Пожарная безопасность: |  | | | «06» сентября 2021 г. | |
| Инструктирующий | *Подпись* | | | Сорокин А. Б. доцент кафедры ВТ | |
| Инструктируемый | *Подпись* | | | Зыков А.А. | |
|  | | | |  | |
| С правилами внутреннего распорядка ознакомлен: | | | | «06» сентября 2021 г. | |
|  | *Подпись* | | | Зыков А.А. | |

|  |
| --- |
| МИНОБРНАУКИ РОССИИ |
| Федеральное государственное бюджетное образовательное учреждение высшего образования **«МИРЭА − Российский технологический университет»**  **РТУ МИРЭА** |

**СОВМЕСТНЫЙ РАБОЧИЙ ГРАФИК ПРОВЕДЕНИЯ  
ПРАКТИКИ**

студента Зыкова А.А. 4 курса группы ИКБО-04-18 очной формы обучения, обучающегося по направлению подготовки 09.03.04 Программная инженерия, профиль «Системы поддержки принятия решений».

|  |  |  |  |
| --- | --- | --- | --- |
| **Неделя** | **Сроки**  **выполнения** | **Этап** | **Отметка о выполнении** |
| 1 | 06.09.2021 | Подготовительный этап, включающий в себя организационное собрание (Вводная лекция о порядке организации и прохождения научно-исследовательской работы, инструктаж по технике безопасности, получение задания на практику) |  |
| 1-6 | 06.09.2021-11.10.2021 | Исследовательский этап (Формирование текстового описания, актуальности темы исследования и анализа предметной области) |  |
| 7 | 18.10.2021 | Представление руководителю структурированного материала об объекте исследования согласно содержанию выше указанного этапа |  |
| 7-10 | 18.10.2021-08.11.2021 | Исследовательский этап (Формирование текстового описания, постановки задачи и технического задания |  |
| 11 | 15.11.2021 | Представление руководителю структурированного материала об объекте исследования согласно содержанию выше указанного этапа |  |
| 11-14 | 22.11.2021-29.11.2021 | Представление руководителю текстового описания процессов проектирования на основе объектных диаграмм и реализации их в соответствующих инструментальных средах |  |
| 15 | 07.12.2021 | Представление руководителю структурированного материала об объекте исследования согласно содержанию выше указанного этапа |  |
| 16 | 14.12.2021 | Представление руководителю предварительной версии отчета с обеспечением согласованности материала по всем его частям, полученных на предыдущих этапах |  |
| 17 | 21.12.2021 | Подготовка окончательной версии отчета и презентационного материала по практике (Оформление материалов отчета и презентации в полном соответствии с требованиями на оформление письменных учебных работ студентов) |  |

Содержание практики и планируемые результаты согласованы с руководителем практики от профильной организации

Руководитель практики от  
кафедры \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Сорокин А.Б., к.т.н., доцент/

Обучающийся \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Зыков А.А./

**Согласовано:**

Заведующий кафедрой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Платонова О.В., к.т.н., доцент/

Содержание

[Содержание 6](#_Toc91506778)

[1 текстовое описание 8](#_Toc91506779)

[1.1 Описание видов рекомендательных систем 9](#_Toc91506780)

[1.1.1 Рекомендации, основанные на контенте 9](#_Toc91506781)

[1.1.2 Рекомендательные системы, основанные на коллаборативной фильтрации 12](#_Toc91506782)

[1.1.3 Рекомендательная система, основанная на знаниях 15](#_Toc91506783)

[1.1.4 Гибридные рекомендательные системы 16](#_Toc91506784)

[2 Актуальность темы исследования 18](#_Toc91506785)

[3 Постановка задачи и техническое задание 21](#_Toc91506786)

[3.1 Основные задачи 21](#_Toc91506787)

[3.2 Техническое задание 21](#_Toc91506788)

[3.2.1 Введение 21](#_Toc91506789)

[3.2.2 Назначение разработки 21](#_Toc91506790)

[3.2.3 Требования к программе или программному изделию 22](#_Toc91506791)

[3.2.4 Требования к надежности 22](#_Toc91506792)

[3.2.5 Время восстановления после отказа 23](#_Toc91506793)

[3.2.6 Отказы из-за некорректных действий оператора 23](#_Toc91506794)

[3.2.7 Условия эксплуатации 24](#_Toc91506795)

[3.2.8 Требования к численности и квалификации персонала 24](#_Toc91506796)

[3.2.9 Требования к составу и параметрам технических средств 24](#_Toc91506797)

[3.2.10 Требования к информационной и программной совместимости 25](#_Toc91506798)

[3.2.11 Требования к исходным кодам и языкам программирования 25](#_Toc91506799)

[3.2.12 Требования к программным средствам, используемым программой 25](#_Toc91506800)

[3.2.13 Требования к защите информации и программ 26](#_Toc91506801)

[3.2.14 Специальные требования 26](#_Toc91506802)

[3.2.15 Требования к программной документации 26](#_Toc91506803)

[3.2.16 Технико-экономические показатели 26](#_Toc91506804)

[3.2.17 Стадии и этапы разработки 27](#_Toc91506805)

[3.2.18 Содержание работ по этапам 28](#_Toc91506806)

[4 Этап проектирования 30](#_Toc91506807)

[4.1 Реализация общей структуры 32](#_Toc91506808)

[4.2 Реализация рекомендательной системы 33](#_Toc91506809)

[Заключение 35](#_Toc91506810)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 36](#_Toc91506811)

1. текстовое описание

Тема работы: построение различных видов рекомендательных систем фильмов для реализации наилучшего решения.

Нейронные сети являются одной из областей искусственного интеллекта. Основной их целью является прогнозирование и, как следствие, моделирование различных ситуаций в зависимости от исходных данных. Задачи, решаемые типичной нейронной сетью:

* это классификация;
* прогнозирование;
* распознавание.

Нейронные сети часто описываются в виде слоёной структуры, где каждый слой состоит из входных, скрытых или выходных нейронов. Клетки между собой не взаимодействуют в пределах уровня, а связаны только со соседними слоями. Самая простая нейронная сеть прямого распространения имеет два входных нейрона и один выходной.

Замечательным примером биологической нейронной сети является человеческий мозг, представляющий собой многоуровневую биологическую нейронную сеть, которая получает, хранит и обрабатывает информацию от органов чувств.



Рисунок 1 – Схема искусственного нейрона

Как видно на рисунке 1.1, у нейрона есть n входов xi, у каждого из которого есть вес wi, на который умножается число, проходящий по связи. После этого взвешенные сигналы xi⋅wi складываются и передаются далее.

Без дополнительных расчетов передавать взвешенную сумму на выход из нейрона достаточно бессмысленно — необходимо информацию обработать и сформировать полноценный выходные данные. Роль преобразователя взвешенной суммы берет на себя функция активации, которая преобразует взвешенную сумму в число, которое и будет являться выходом нейрона. Функция активации обозначается ϕ(взвешенной суммы).

Самые часто используемые функции активации (далее w – взвешенная сумма):

* Функция единичного скачка. Если взвешенная сумма> пороговое значение, возвращаем 1, иначе 0;
* Сигмоидальная функция, , где показывает степень крутизны функции.
* Гиперболический тангенс, , где показывает степень крутизны функции.
* Исправление линейных единиц,

Нейронные сети могут учиться и развиваться самостоятельно, накапливая опыт на основе допущенных ими ошибок.

Как основные можно выделить следующие типы нейронных сетей:

1. Нейронные сети прямого распространения (feed forward neural networks, FF или FFNN)
2. Сети радиально-базисных функций (radial basis function, RBF)
3. Нейронная сеть Хопфилда (Hopfield network, HN)
4. Цепи Маркова (Markov chains, MC или discrete time Markov Chains, DTMC)
5. Машина Больцмана (Boltzmann machine, BM)
6. Ограниченная машина Больцмана (restricted Boltzmann machine, RBM)
7. Автокодировщик (autoencoder, AE)
8. Разреженный автокодировщик (sparse autoencoder, SAE)
9. Вариационные автокодировщики (variational autoencoder, VAE)
10. Шумоподавляющие автокодировщики (denoising autoencoder, DAE)
11. Сеть типа «deep belief» (deep belief networks, DBN)
12. Свёрточные нейронные сети (convolutional neural networks, CNN) и глубинные свёрточные нейронные сети (deep convolutional neural networks, DCNN)

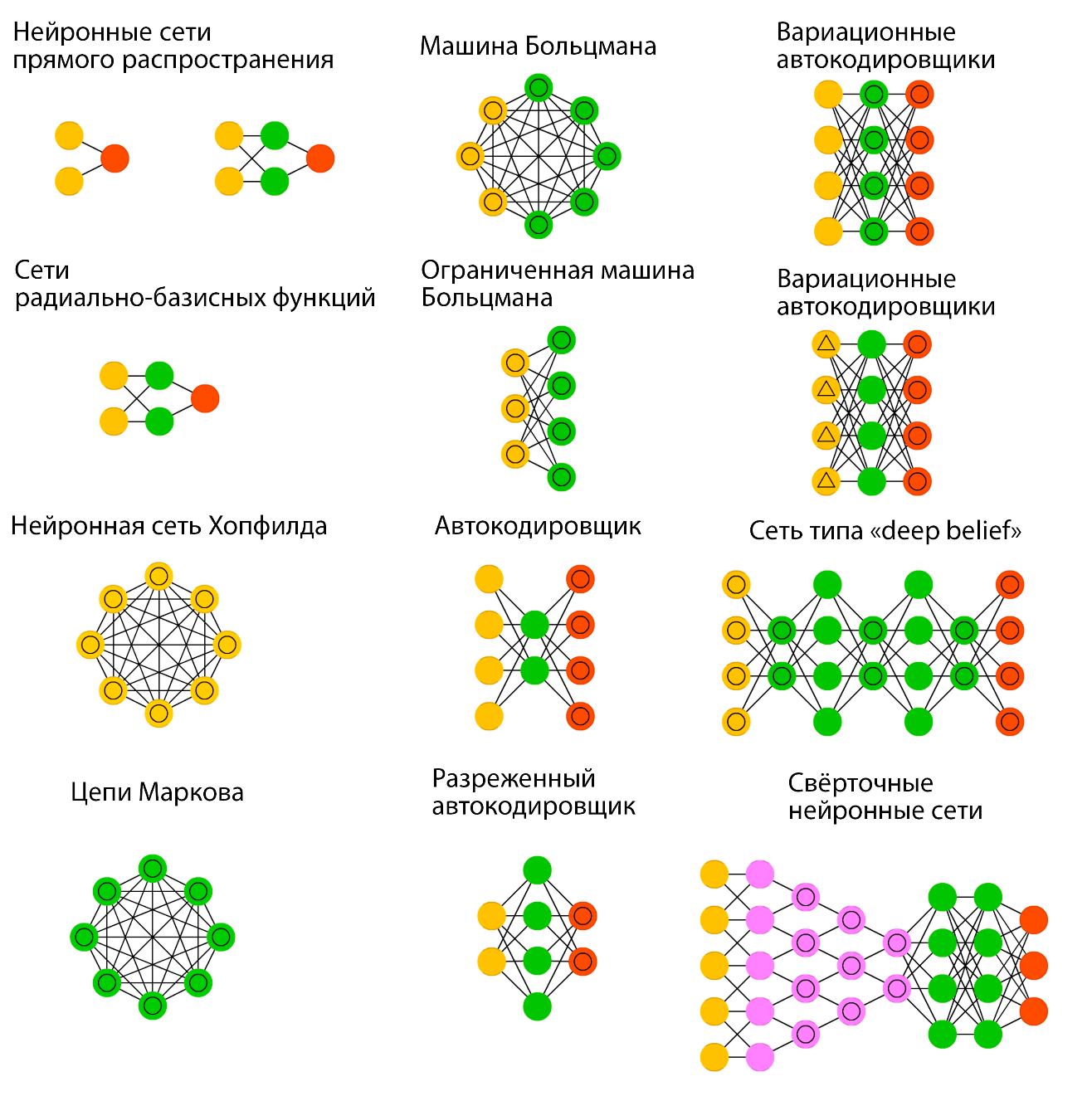


Рисунок 2 – основные виды нейронных сетей

* 1. Описание видов рекомендательных систем
     1. Нейронные сети прямого распространения

**Нейронные сети прямого распространения** (feed forward neural networks, FF или FFNN) и **перцептроны** (perceptrons, P) – это тип нейронных сетей, отличающийся прямой передачей информации (с учетом весов). Для обучения нейронной сети обычно используется метод обратного распространения ошибки, в котором для тренировки вносятся входные данные и ожидаемые выходные. Данный формат тренировки называется «обучение с учителем», помимо этого, существует «обучения без учителя», в качестве особенности которого можно выделить самостоятельное получение нейронной сетью выходных данных. Теоретически сеть может смоделировать структуру скрытых слоев так, чтобы получить необходимые выходные данные, на практике сети используются редко, тем не менее их часто комбинируют с другими типами.

* + 1. Сети радиально-базисных функций

**RBF (**radial basis function) - сети, позволяющие нам перейти к использованию приближений функций. Сети радиальных базисных функций можно представить двумя способами:

В качестве линейной модели, в которой мы вначале произвели извлечение признаков, а признаки стали ядрами радиальных базисных функций.

В виде нейронной сети с одним скрытым слоём и радиальными базисными функциями в качестве функции активации.

* + 1. Нейронная сеть Хопфилда

**Нейронная сеть Хопфилда** (Hopfield network, HN) — это нейронная сеть с симметричной матрицей связей. Во время получения входных данных каждый нейрон является входным, в процессе обучения он становится скрытым, а затем становится выходным. Обучение происходит следующим образом: значения нейронов устанавливаются в соответствии с желаемым шаблоном, вычисляются веса, которые в дальнейшем не будут меняться. После тренировки на одном или нескольких шаблонах, сеть всегда будет сводиться к одному из них. Она стабилизируется в зависимости от общего энергетического состояния всех нейронов. Каждый нейрон может находиться в одном из 2-х состояний, зачастую он может быть «заторможен» (соответствует -1), либо «возбужден» (соответствует 1). Подобные нейронные сети зачастую сравнивают с ассоциативной памятью; как, например, человек, видя половину объекта, может воссоздать вторую часть по памяти, так и эта сеть, получая таблицу, наполовину зашумленную, восстанавливает её до полной.

* + 1. Цепи Маркова

**Цепи Маркова** (Markov chains, MC или discrete time Markov Chains, DTMC) — это предшественники машин Больцмана (BM) и сетей Хопфилда (HN). Общий смысл заключается в следующем: какова вероятность попасть в один из следующих нейронов, если я нахожусь в каком-то конкретном? Каждое следующее состояние зависит только от предыдущего. Хотя на самом деле цепи Маркова не являются нейронной сетью, они весьма похожи. Также цепи Маркова не обязательно полносвязны.

* + 1. Машина Больцмана

**Машина Больцмана** (Boltzmann machine, BM) имеет сходство с сетью Хопфилда, но некоторые её нейроны помечены как входные, а некоторые — как скрытые. Входные нейроны в последствии становятся выходными. Машина Больцмана — это стохастическая сеть. Обучение проходит по методу обратного распространения ошибки или по алгоритму сравнительной расходимости.

* + 1. Ограниченная машина Больцмана

**Ограниченная машина Больцмана** (restricted Boltzmann machine, RBM) имеет аналогичный принцип работы, что и машина Больцмана, следовательно, и основана она на сети Хопфилда. Основополагающей разницей является её ограниченность, проявляющееся в отсутствии связей между нейронами одного типа. Ограниченную машину Больцмана можно обучать методом обратного распространения ошибки, но с небольшим дополнением: вместо прямой передачи данных и обратной передачи ошибки нужно передавать данные сперва в прямом направлении, затем в обратном.

* + 1. Автокодировщик

**Автокодировщик** (autoencoder, AE) чем-то похож на сеть прямого распространения. Основной идеей является автоматическое сжатие не информации. Сама сеть по форме напоминает песочные часы, что обуславливается меньшим количеством скрытых слоев по сравнению с входными и выходными, помимо этого сеть симметрична. Сеть обучается методом обратного распространения ошибки, с учетом того факта, что ошибка изначально равна разнице входом и выходом.

* + 1. Разреженный автокодировщик

**Разреженный автокодировщик** (sparse autoencoder, SAE) — нейронная сеть, имеющая диаметрально противоположный принцип работы, нежели обычный автокодировщик. Вместо оптимизации количества нейронов, сводящей их количество к минимуму, мы, наоборот, увеличиваем их количество. Вместо постепенного сжатия по направлению к центру сеть постепенно становится все больше и разрежение, что и обуславливает название данного типа. Сети подобного типа крайне благоприятно сказываются на скорости и оптимизации работы с большим набором всевозможных мелких свойств данных. Для корректной работы сети, помимо исходных данных передается специальный фильтр разреженности, допускающий возникновение только определенных типов ошибок.

* + 1. Вариационные автокодировщики

**Вариационные автокодировщики** (variational autoencoder, VAE) обладают схожей с обычными автокодировщиками архитектурой, но стремятся к абсолютно иному результату: приближению вероятностного распределения входных данных. Данная идея исходит из машины Больцмана. Тем не менее, сети подобного типа опираются на байесовскую математику, когда речь идёт о вероятностных выводах и независимости, которые интуитивно понятны, но сложны в реализации. Если обобщить, то можно сказать, что эта сеть принимает в расчёт влияния нейронов друг на друга. Если что-то происходит в одном месте, а что-то — в другом, то эти события не обязательно связаны, и это должно учитываться.

* + 1. Шумоподавляющие автокодировщики

**Шумоподавляющие автокодировщики** (denoising autoencoder, DAE) — это автокодировщик, в котором входные данные подаются в зашумленном состоянии. Ошибку мы вычисляем так же, и выходные данные сравниваются с зашумленными. Благодаря этому сеть учится обращать внимание на более широкие свойства, поскольку маленькие могут изменяться вместе с шумом.

* + 1. Сеть типа «deep belief»

**Сеть типа «deep belief»** (deep belief networks, DBN) — это название, которое получил тип архитектуры, в которой сеть состоит из нескольких соединённых ограниченных машин Больцмана или вариационных автокодировщиков. Такие сети обучаются поблочно, причём от каждого блока требуется исключительно умение кодировать предыдущий. Такая техника называется «жадным обучением» и заключается она в выборе локальных оптимальных решений, не гарантирующих оптимальный конечный результат. Также сеть можно обучить отображать данные в виде вероятностной модели. Если использовать обучение без учителя, стабилизированную модель можно использовать для генерации новых данных.

1. Актуальность темы исследования

Проблема поиска подходящего товара или услуги является вечной: ежедневно человек делает выбор в пользу наиболее подходящего, начиная с выбора яблок в продуктовом магазине, заканчивая выбором подходящего врача. С развитием технологий кинематограф не стал исключением и фильмы начали распространять через онлайн-кинотеатры. Например, по итогам 2020 года российская аудитория онлайн-кинотеатров выросла на 52% и составила 8,5 млн человек, согласно исследованию "ТМТ Консалтинг".

В некоторых случаях, крупные хиты могут выходить в обход кинотеатров, сразу в «онлайне», или появляться в интернете через неделю-две после премьеры.

С ростом спроса на просмотр фильмов в интернете растет и предложение, что приводит к появлению новых игроков на этом рынке. На данный момент конкуренция растет, идет борьба за пользователей. Среди мировых «игроков» можно выделить Netflix, Amazon Prime, Disney+ и Apple TV. Более того, спрос оказался на столько высок, что привел к появлению локальных сервисов в России, таких как: Ivi, Кинопоиск, Окко, Амедиатека, Megogo и многие другие.

Таким образом, для огромного количества пользователей сети Интернет, доступ к фильмам и сериалам значительно упростился. Но возникла другая проблема: что посмотреть при огромном выборе контента? Искать подходящий фильм вручную может оказаться непростой задачей, требующей времени.  
По частоте посещений пользователей можно условно разделить на группы:

1. 1-2 раза в месяц
2. каждый день или почти каждый день
3. несколько раз в месяц
4. два-три раза в неделю

В качестве основы для такого разделения пользователей была использована инфографика от Telecom Daily за первое полугодие 2021 года (Рисунок 5).



Рисунок 5 - Частота посещений онлайн-кинотеатров в России по данным TELECOM DAILY

Всех пользователей объединяет одна проблема - трата времени на подбор подходящего фильма для просмотра. Более того, чем больше просматривается контента, тем сложнее становится найти что-то подходящее. Нередко случаются ситуации, когда в качестве рекомендованного к просмотру контента всем предлагаются одни и те же популярные варианты, когда как менее известные фильмы, не смотря на их возможную актуальность для конкретного пользователя, просто теряются среди подобных вариантов.

Релевантные рекомендации сокращают время, необходимое для поиска товаров и услуг, и значительно увеличивают вероятность попадания в поле зрения пользователя других объектов, которые смогут его заинтересовать. В результате повышается лояльность и удовлетворенность пользователей веб-сервисами. Как правило, пользователи также взаимодействуют с большим количеством товаров, что приводит к увеличению потребления контента и росту прибыли. Кроме того, информационные бюллетени, персонализированные рекламные материалы и push-уведомления побуждают пользователей возвращаться, увеличивают частоту посещений постоянными пользователями и уменьшают отток клиентов.

Одна из основных задач рекомендательной систем – увеличить вовлеченность пользователя в использование сервиса. Например, 80% просматриваемого контента на Netflix – это рекомендованные пользователям фильмы или сериалы. Благодаря упрощению пользовательского опыта за счет рекомендательной системы им удалось увеличить процент постоянных пользователей и сэкономить 1 миллиард долларов на привлечении пользователей (на момент 2016 года). Данная рекомендательная система не сможет представлять конкуренции такому гиганту, но при этом не требует больших вложений, улучшая пользовательский опыт, который впоследствии конвертируется в увеличение статистики посещения и просмотров, что напрямую влияет на увеличение прибыли и сокращение трат на привлечение и удержание пользователей.

Сегодня каждой компании просто необходимо наладить процесс сбора данных и уметь грамотно и эффективно использовать их в бизнесе, тем самым оптимизируя и улучшая пользовательский опыт, снижая расходы, увеличивая выручку и средний чек, и повышая рентабельность бизнеса в целом.

1. Постановка задачи и техническое задание
   1. Основные задачи

Основными задачами являются:

1. Анализ различных видов РС
2. Выбор наиболее эффективного подхода к реализации РС
3. Реализация наиболее эффективной РС
4. Реализация веб-интерфейса для взаимодействия с РС
   1. Техническое задание
      1. Введение

В качестве объектов для реализации выступает рекомендательная система фильмов Reccorsys (Recommendation core system) и пользовательский web-интерфейс. Данная рекомендательная система может применяться в приложениях и сервисах, связанных с фильмами (их просмотром или оценкой).

* + 1. Назначение разработки

Итоговая программа используется в качестве помощника конечному пользователю в вопросе выбора фильма для просмотра, основываясь на его предпочтениях, а также предпочтениях других пользователей. Она позволит пользователю упростить проблему выбора фильма для просмотра, тем самым экономя время. Для создания удачного пользовательского опыта реализуется web-интерфейс для взаимодействия с программой.

* + 1. Требования к программе или программному изделию

Выполняемые функции:

* Обработка входных данных
* Получение новых данных в качестве результата работы алгоритма рекомендации
* Вывод данных

Входные данные должны быть представлены в формате csv или json.

Система должна удовлетворять критериям эффективности:

* Максимально допустимое время ответа системы – 10 сек
* Максимальный используемый объем оперативной памяти – 2 гб
  + 1. Требования к надежности

Надежное (устойчивое) функционирование программы должно быть обеспечено рядом организационно-технических мероприятий, а именно:

* Организацией бесперебойного питания технических средств;
* Выполнением рекомендаций Министерства здравоохранения и социального развития РФ, изложенных в приказе от 14 октября 2011 г. № 1175н «Об утверждении межотраслевых типовых норм времени работы по сервисному обслуживанию оборудования телемеханики, сопровождению и доработке программного обеспечения»
* Выполнением требований ГОСТ 51188-98. Защита информации. Испытания программных средств на наличие компьютерных вирусов;
* Необходимым уровнем квалификации сотрудников профильных подразделений.
  + 1. Время восстановления после отказа

Время восстановления после отказа, вызванного сбоем электропитания технических средств (иными внешними факторами), не должно превышать времени, необходимого для урегулирования этих проблем.

Время восстановления после отказа, вызванного неисправностью технических средств, фатальным сбоем (крахом) приложения, не должно превышать времени, требуемого на устранение неисправностей технических средств и переустановки программных средств.

Для сохранности данных производится их резервное копирование каждые 7 дней.

* + 1. Отказы из-за некорректных действий оператора

Предотвратить отказы программы вследствие некорректных действий оператора (пользователя) при взаимодействии с программой. Во избежание возникновения отказов программы по указанной выше причине следует тщательно проверять вводимые данные и ограничить возможности пользователя, во избежание ошибок программы.

* + 1. Условия эксплуатации

Климатические условия эксплуатации, при которых должны обеспечиваться заданные характеристики, должны удовлетворять требованиям, предъявляемым к техническим средствам в части условий их эксплуатации.

* + 1. Требования к численности и квалификации персонала

Минимальное количество персонала, требуемого для работы

программы, должно составлять не менее 2 штатных единиц: системный

программист и конечный пользователь программы – оператор. Системный программист должен иметь техническое образование. В перечень задач, выполняемых системным программистом, должны входить:

* задача поддержания работоспособности технических средств;
* задачи установки (инсталляции) и поддержания работоспособности системных программных средств – операционной системы;
* задача установки (инсталляции) программы.
  + 1. Требования к составу и параметрам технических средств

Для сервера:

В состав технических средств должен входить сервер или персональный компьютер с настроенным окружением для запуска python-файлов и JavaScript-файлов. Минимальные требования к аппаратной части:

* Оперативная память (ОЗУ) — объемом не менее 2 Гб;
* Объём свободного дискового пространства (SSD) — не менее 100 Гб;
* Процессор – Intel Core i5 (или выше);
* Доступ в интернет

Для клиента:

В состав технических средств должен входить персональный компьютер с доступом к интернету и скоростью соединения не менее 10 мб/с.

* + 1. Требования к информационной и программной совместимости

Пользовательский интерфейс должен быть интуитивно понятным. На данных этапах разработки доступа к подстройке системы извне не предусмотрен.

* + 1. Требования к исходным кодам и языкам программирования

Исходные коды рекомендательной системы и серверной части web-приложения должны быть реализованы на языке программирования Python версии 3.0 и выше. Клиентская часть должна быть реализована на языке программирования JavaScript спецификации ES6 (ES2015) и выше. В качестве среды разработки предлагается использовать Visual Studio Code.

* + 1. Требования к программным средствам, используемым программой

Системные программные средства, используемые программой, должны быть представлены операционными системы Windows 7 и выше, MacOS версии 10.9 и выше. Для выполнения математических операций в Python-коде используется библиотека math, для работы с массивами – numpy. Для клиентской части используются библиотеки: react – для расширения возможностей языка и добавления «реактивности», typescript – для типизации, node-sass – для стилизации.

* + 1. Требования к защите информации и программ

В системе должен быть обеспечен надлежащий уровень защиты информации в соответствии с законом о защите персональной информации и программного комплекса в целом от несанкционированного доступа – «Об информации, информационных технологиях и о защите информации» Федеральный закон от 27.07.2006 г. No 149-ФЗ.

* + 1. Специальные требования

Программа должна обеспечивать взаимодействие с пользователем (оператором) посредством графического пользовательского интерфейса.

* + 1. Требования к программной документации

Программная документация для разработчика системы не требуется. Руководство пользователя не требуется.

* + 1. Технико-экономические показатели

Работы по проекту разделены на 5 этапов, суммарная длительность

которых составила 60 дней. Договорная цена разработки составила 300 000

рублей. Плановая прибыль от проекта составляет 20% от текущего уровня.

* + 1. Стадии и этапы разработки

Стадии разработки:

Разработка должна быть проведена в пять стадий:

* разработка технического задания;
* постановка задачи и анализ предметной области;
* проектирование программного обеспечения;
* разработка программного кода;
* тестирование и дальнейшая модификация.

Этапы разработки:

На стадии разработки технического задания должен быть выполнен этап разработки, согласования и утверждения настоящего технического задания.

На стадии постановки задачи и анализа предметной области должны

быть выполнены перечисленные ниже этапы работ:

* определить и вербально описать предметную область как сферу человеческой деятельности;
* предметную область определить, как объект со всем комплексом понятий и знаний о ее функционировании;
* обозначить цель разработки программного обеспечения. В соответствии с целью определить задачи.

На стадии проектирования программного обеспечения должны быть выполнены перечисленные ниже этапы работ:

* определить входные и выходные данные программного обеспечения;
* предоставить проектную реализацию в виде графических диаграмм, основанных на структурном подходе.

На стадии разработки программного кода должен быть выполнен этап написания кода на определенном языке программирования.

На стадии тестирование и дальнейшая модификация должны быть выполнены перечисленные ниже этапы работ:

* оценка разрабатываемого программного обеспечения, чтобы проверить его возможности, способности и соответствие ожидаемым результатам;
* описать дальнейшее видение программной реализации.
  + 1. Содержание работ по этапам

На этапе разработки технического задания должны быть выполнены

перечисленные ниже работы:

* определение и уточнение требований к техническим средствам;
* определение требований к программе;
* определение стадий, этапов и сроков разработки программы и
* документации на неё;
* выбор языков программирования;
* согласование и утверждение технического задания.

На этапе постановка задачи и анализ предметной области должны быть выполнены перечисленные ниже работы:

* описать сценарии использования программного обеспечения;
* определить квалификацию пользователей и условия, в которых они будут работать;
* определить объемы передаваемых данных и модель данных, которая будет использоваться, внесите ясность, в то, какие типы данных будут использоваться;
* описать примеры аналогичных реализаций, уточнить их преимущества и недостатки.

На этапе проектирование программного обеспечения должны быть

выполнены перечисленные ниже работы:

* создать функциональную модели программного обеспечения (IDEF0);
* создать базу данных основанную на диаграмме «сущность-связь» (IDEF1);
* создать модель потоков данных (нотация Гейна-Сэрсона).

На этапе разработки программного кода должна быть выполнена работа по программированию и отладке программы. Должна быть выполнена разработка программных документов в соответствии с требованиями ГОСТ 19.101-77.

На этапе тестирование и дальнейшая модификация должны быть выполнены перечисленные ниже работы:

* разработка, согласование и утверждение программы и методики испытаний;
* корректировка программы и программной документации по результатам испытаний.

1. Этап проектирования

В качестве методологии разработки ПО выбрана каскадная модель с промежуточным контролем (Рисунок 6).



Рисунок 6 - Каскадная модель с промежуточным контролем

Достоинства каскадной модели:

* постоянство системных требований на протяжении всего жизненного цикла разработки;
* каждый этап заканчивается набором проектной документации, которая должна отвечать критерию полноты;
* простота исполнения и применения модели обусловлена понятностью действий;
* легкость планирования работ и сроков реализации программного обеспечения на основании логической связанности последовательности этапов жизненного цикла.

Недостатки каскадной модели:

* большая дистанция жизненного цикла приводит к неадекватности чёткого формирования требований, а иногда даже к невозможности их изменений;
* низкая пластичность в управлении проектом;
* логическая последовательность определяет линейность этапов разработки, возвращение к предыдущим шагам приводит к срыву разработки программного обеспечения;
* промежуточный продукт не может использоваться;
* разработка специальных систем на основе гибкого моделирования невозможна;
* разработка больших информационных систем вызывает сложности интеграции всех программных продуктов;
* модель обусловлена большими рисками при реализации уникальных программных продуктов, в связи с невозможностью реализации гибкого моделирования.
  1. Реализация общей структуры

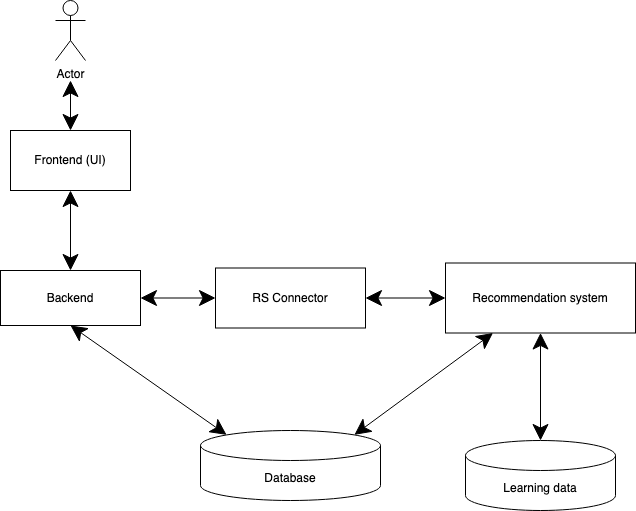


Рисунок 7 - Общая структура приложения

Общая структура приложения включает в себя (Рисунок 7):

* Frontend - web-интерфейс для взаимодействия пользователя с системой, реализуется с использованием языка программирования Javascript (спецификации ES6 и выше), библиотек React и node-sass;
* Backend – серверная часть пользовательского интерфейса, реализуется с использованием языка программирования Python и фреймворка Django;
* RS Connector (Recommendation service connector) – сервис-коннектор между серверной частью пользовательского интерфейса и рекомендательной системой, реализуется с использованием языка программирования Python;
* Recommendation system – рекомендательная система фильмов, реализуется с использованием языка программирования Python;
* Database – база данных, в которой хранятся данные пользователей ресурса, а так же предустановленные данные, необходимые для ускорения работы программы, представлена в формате базы данных SQL;
* Learning data – база данных с обучающей выборкой, необходимая для «холодного старта», представлена в формате набора файлов формата .csv.
  1. Реализация рекомендательной системы



Рисунок 8 - Структура рекомендательной системы

Рекомендательная система состоит из (Рисунок 8):

* RS Interface – интерфейс рекомендательной системы для связи с внешними потребителями (RS Connector);
* RS Controller – контроллер рекомендательной системы, в котором хранится логика для обработки и систематизации данных, полученных в результате работы рекомендательной системы и обработки запросов от интерфейса рекомендательной системы. Также имеет механизмы обращения к базе данных и к обучающей выборке;
* Hybrid RS – реализация гибридной рекомендательной системы, взаимодействует рекомендательными системами, такими как: основанная на контенте (Content-based RS), коллаборативной фильтрации (Collaborative Filtering RS);
* Content-based RS – рекомендательная система, основанная на контенте, принимает запросы от Hybrid RS и возвращает результаты работы рекомендательного алгоритма;
* Collaborative Filtering RS – рекомендательная система, основанная на коллаборативной фильтрации, принимает запросы от Hybrid RS и возвращает результаты работы рекомендательного алгоритма.

Заключение

Были установлены цели и задачи работы, проведено исследование предметной области и собрана пользовательская статистика. Сформировались четкие условия и требования к продукту, проведен архитектурный анализ и построение верхнеуровневой архитектуры. Сформировано техническое задание, определены программные средства реализации ПО.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Thomas Hofmann and Jan Puzicha. Latent class models for collaborative filtering. In IJCAI, volume 99, pages 688–693, 1999
2. Gediminas Adomavicius and Alexander Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions.  
   Knowledge and Data Engineering, IEEE Transactions on, 17(6):734–749, 2005.
3. Akiko Aizawa. An information-theoretic perspective of tf–idf measures // Information Processing & Management, 2003. Vol. 39, Issue 1, pp. 45-65