Федеральное государственное автономное образовательное учреждение высшего образования

НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ «ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

Высшая школа бизнеса

КУРСОВОЙ ПРОЕКТ

Разработка рекомендательной системы для маркетплейса Farfetch

по направлению подготовки: 38.03.05 «Бизнес-информатика»
образовательная программа «Бизнес-информатика»

Проект выполнили:
Алиев Хайрутдин Аллилович, ББИ1901
Зубов Дмитрий Сергеевич, ББИ1901
Курбанов Иван Сергеевич, ББИ1901
Лухнев Игорь Дмитриевич, ББИ1901
Шишков Максим Алексеевич, ББИ1901

Руководитель курсового проекта:
канд. экон. наук, доцент,
зам. рук. деп. бизнес-информатики
Брускин Сергей Наумович

Курсовой проект соответствует / не соответствует требованиям (*нужное подчеркнуть*)

Москва 2022

СОДЕРЖАНИЕ

СОДЕРЖАНИЕ 2
ВВЕДЕНИЕ 4
СОСТАВ КОМАНДЫ 5
ПОДХОД К РАБОТЕ 5
KPI ПРОЕКТА 6
РАЗДЕЛ 1. ОБСЛЕДОВАНИЕ БИЗНЕСА 8
1.1 ЗАДАЧИ И ЦЕЛЬ ИССЛЕДОВАНИЯ 8
1.2 ОПИСАНИЕ ОБЪЕКТА ИССЛЕДОВАНИЯ 8
1.3 ОПИСАНИЕ БИЗНЕС-КЕЙСА 13
1.4 КОНЦЕПЦИЯ РАЗРАБАТЫВАЕМОЙ МОДЕЛИ РЕКОМЕНДАТЕЛЬНОЙ СИСТЕМЫ 15
1.5 ИСТОЧНИКИ ДАННЫХ 17
1.6 ИДЕНТИФИКАЦИЯ СТЕЙКХОЛДЕРОВ 20
1.7 ВЫВОДЫ ПО РАЗДЕЛУ 22
РАЗДЕЛ 2. ФОРМАЛИЗАЦИЯ ФУНКЦИОНАЛЬНЫХ И НЕФУНКЦИОНАЛЬНЫХ ТРЕБОВАНИЙ 23
2.1 НЕФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ К СИСТЕМЕ 23
2.2 ФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ К СИСТЕМЕ 27
2.3 ВЫВОДЫ ПО РАЗДЕЛУ 31
РАЗДЕЛ 3. ВЫБОР ИТ-РЕШЕНИЯ 32
3.1 ГРАНИЦЫ ПРОЕКТА И СТРАТЕГИЯ РАЗВИТИЯ 32
3.2 ОПРЕДЕЛЕНИЕ СОСТАВА КРИТЕРИЕВ ДЛЯ ВЫБОРА ИТ-РЕШЕНИЯ 32
3.3 АНАЛИЗ СУЩЕСТВУЮЩИХ ИТ-РЕШЕНИЙ 34
3.4 ОБОСНОВАНИЕ ВЫБОРА ИТ-РЕШЕНИЯ 36
3.5 ВЫВОДЫ ПО РАЗДЕЛУ 39
РАЗДЕЛ 4. ПРАКТИЧЕСКАЯ РЕАЛИЗАЦИЯ 40
4.1 РЕАЛИЗАЦИЯ АЛГОРИТМОВ РЕКОМЕНДАЦИЙ RTB 40

4.2 РАЗРАБОТКА API 49
4.3 ВЫВОДЫ ПО РАЗДЕЛУ 51
РАЗДЕЛ 5. СТРАТЕГИЯ ВНЕДРЕНИЯ 52
5.1 ИНФРАСТРУКТУРНЫЕ ИЗМЕНЕНИЯ 52
5.2 A|B ТЕСТ 53
ЗАКЛЮЧЕНИЕ 55
ИСПОЛЬЗОВАННЫЕ МАТЕРИАЛЫ 56

ВВЕДЕНИЕ

В современном мире использование ИТ уже не является конкурентным преимуществом, теперь конкуренция возможна только при постоянном и инновационном развитии цифровых услуг и ИТ-решений, предлагаемых компанией. В частности, область маркетплейсов, вне зависимости от характера размещаемых товаров и услуг, крайне требовательна к продвинутым рекомендательным системам и разработке собственного ПО, решающего это задачу.

Компания Farfetch, как представитель маркетплейсов с вполне конкретным видом товаров не может использовать решения, идентичные тем, которые используются на маркетплейсах общего назначения, поскольку желания пользователя могут быть очень специфическими, а предложить смежные товары не получится. В связи с этим, для Farfetch поиск, разработка и исследование новых рекомендательных систем является ключевым драйвером развития платформы.

Цель данной работы заключается в разработке рекомендательной системы товаров для покупателей компании Farfetch, основанной на методах интеллектуального анализа данных, включая машинное и глубинное обучение.

В рамках данной работы для достижения обозначенной цели, команда поставила перед собой следующие задачи:

1. Исследование предметной области, выявление проблем текущей рекомендательной системы;
2. Сбор и анализ функциональный и нефункциональных требований к разрабатываемой рекомендательной системе;
3. Выбор способа реализации разработки рекомендательной системы и разработка архитектуры пилотного решения;
4. Сбор, анализ и отчистка данных для модели.
5. Разработка демонстрационной модели;
6. Тестирование демонстрационной модели;
7. Исследование перспектив развития проекта и возможности внедрения решения в промышленную эксплуатацию.

СОСТАВ КОМАНДЫ

Таблица 1. Состав и роли в команде

ФИО	Роль в команде	Состав работ
Алиев Хайрутдин Алилович	Аналитик данных	Обучение моделей машинного обучения и нейронных сетей, настройка гиперпараметров, интерпретация результатов работы алгоритмов машинного обучения, проведение экспериментов, EDA[1]
Зубов Дмитрий Сергеевич	Архитектор ПО	Анализ требований к решению, формулирование ТЗ для разработки, построение архитектуры готового решения, разработка API, внедрение решения, тестирование готового ПО.
Курбанов Иван Сергеевич	Бизнес-аналитик	Проведение интервью с Заказчиком, описание бизнес-задачи, ожиданий от результатов, идентификация стейкхолдеров, ведение проектной документации, сбор и анализ требований, анализ существующих решений, подготовка презентационных материалов. Выбор метрики и бейзлайна[2].
Лухнев Игорь Дмитриевич	Тимлид, ML-инженер	Формализация и валидация проектной документации, организация работы и распределение ролей, разработка архитектуры решения (выбор алгоритмов машинного обучения, разработка методологии EDA[1] и обработки данных, формулирование гипотез), руководство разработкой решения, консультирование в вопросах алгоритмов машинного обучения, дизайн A|B теста. Выбор метрики и бейзлайна[2].
Шишков Максим Алексеевич	Data Scientist	Проведение EDA[1], предобработка данных, проверка гипотез, подготовка данных для алгоритмов машинного обучения в соответствии с методологией и архитектуры решения, поиск и обогащение данных, разработка процесса интеграции системы с внутренними системами Заказчика.

ПОДХОД К РАБОТЕ

Команда в целом применяет комбинированный подход к реализации проекта: каскадный (ведение документации, исследования, проектирование и тестирование идут последовательно друг за другом) и итерационный (в

[1] Exploratory Data Analysis - анализ данных на предмет наличия закономерностей и связей, составление и проверка гипотез

[2] Модель, задающая минимальное качество

вопросах управления задачами и непосредственной разработкой). Детали подхода к работе описаны в Приложении 1 (Устав проекта). Приложение 2 содержит укрупненный план работ по проекту, и не отражает задачи в рамках итерационного подхода. Распределение задач и работ в рамках итерационного подъода организовано на сервисе YouTrack[3] по методу Kanban, так как его суть заключается в визуализации работы, ограничении объема незавершенной работы и достижении максимальной эффективности (или скорости). Kanban-команды стремятся максимально сократить время, которое уходит на выполнение проекта (или пользовательской истории) от начала до конца. Именно эти характеристики описывают наш подход к разработке в рамках проекта. Участники команды распределены на 4 направления работ (Data Science, Dev (разработка), Бизнес Аналитика и Управление). На каждом из этапов на соответствующего участника проектной команды назначаются задачи, которые проходят цикл очередь-работа-ревью-завершение (рисунок 1).

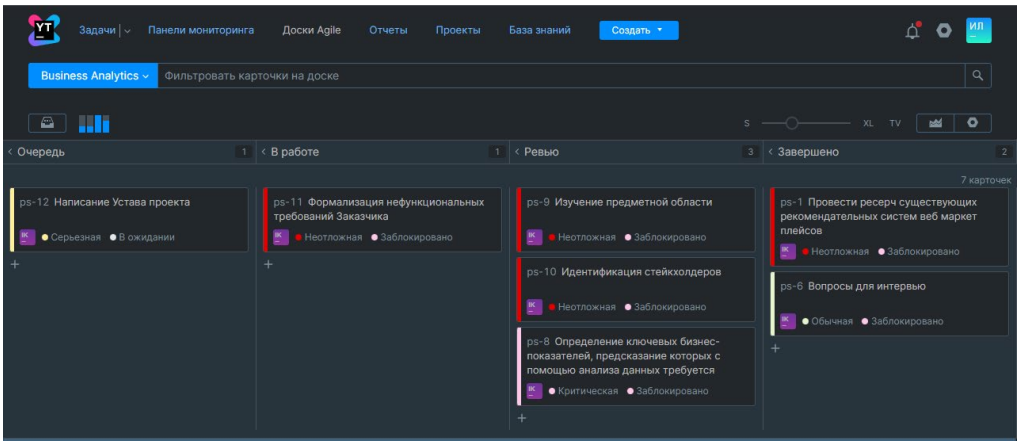

Рисунок 1. Доска Kanban Бизнес-аналитики

KPI ПРОЕКТА

KPI или Key Performance Indicators – это созданная система оценки, которая значительно упрощает процесс определения текущего уровня достижения целей предприятия. Ключевые показатели эффективности или КПЭ – это грамотная оценка нынешнего состояния проекта. Именно поэтому мы вводим следующую систему ключевых показателей эффективности проекта на разных стадиях (реализация, сдача пилотного решения):

1. На стадии реализации проекта необходимо отслеживать следующие показатели:
    a. Среднее отклонение по срокам задач от запланированных не превышает 1 дня для каскадного подхода;
    b. Релиз включает все запланированные на спринт задачи;

[3] Ресурс от JetBrains для управление проектами, в частности разработкой

c. В бэклоге[4] каждого отдела не более 10 задач;
d. В работе у каждого отдела не более 2 задач;
e. В рамках этапа бюджет не превышен более чем на 30%.

2. На стадии сдачи пилотного решения эффективность проекта будет вычисляться по следующим показателям:
    a. Достижение целевого уровня выбранной Заказчиком метрики (подробнее о метрике рассказано в следующих разделах).
    b. Соответствие реализованного функционала требованиям, описанным в настоящем отчете в разделе 2;
    c. Общий бюджет проекта не превысил запланированный более чем на 30%;
    d. Задержка по проекту не превышает 30% от запланированных сроков.

[4] Список задач на стадии «в очереди»

РАЗДЕЛ 1. ОБСЛЕДОВАНИЕ БИЗНЕСА

1.1 ЗАДАЧИ И ЦЕЛЬ ИССЛЕДОВАНИЯ

Объектом данного исследования является компания Farfetch. Предметом исследования являются подходы и технологии для улучшения качества рекомендаций товаров пользователям, направленных на повышение продаж.

Цель данной работы заключается в разработке рекомендательной системы товаров для покупателей компании Farfetch, основанной на методах интеллектуального анализа данных, включая машинное и глубинное обучение.

Рекомендательная система рассматривается как инструмент повышения числа релевантных рекомендаций для покупателей, позволяющий устранить проблемы существующей системы, обеспечить релевантность выдачи товаров, измеренную с помощью метрики pFound (разработана для оценки ранжирования поисковой выдачи Яндекса) на уровне 70%.

Согласно перечисленным во введении задачам выстроен план работы над проектом – с ним можно ознакомиться в Приложении 2.

1.2 ОПИСАНИЕ ОБЪЕКТА ИССЛЕДОВАНИЯ

1.2.1 О КОМПАНИИ

Farfetch — британско-португальская онлайн-платформа для розничной торговли модной одеждой класса люкс, на которой продаются товары из более чем 700 бутиков и брендов со всего мира. Компания была основана в 2007 году португальским предпринимателем Жозе Невисом со штаб-квартирой в Лондоне и основными филиалами в Порту, Браге, Лиссабоне, Нью-Йорке, Лос-Анджелесе, Токио, Шанхае, Гонконге, Сан-Паулу и Дубае.

Компания управляет веб-сайтами на местных языках и мобильными приложениями для международных рынков на английском, испанском, французском, японском, китайском, арабском, немецком, португальском, корейском, итальянском и русском языках. Farfetch имеет офисы в 14 городах и насчитывает более 4500 сотрудников.

Компания изначально задумывалась как user-friendly[5] онлайн маркетплейс брендовой одежды. Нарастив огромную пользовательскую базу за счет клиентоориентированности (быстрая доставка, быстрая и бесплатная обработка возвратов, регулярные скидки), платформа в короткие сроки обрела популярность во всём мире, что дало ей толчок для расширения во многих странах. Farfetch посещает около 10 миллионов посетителей в месяц, а сама

[5] В этом контексте user-friendliness включает в себя соответствие программного продукта ISO 9241-11:2010, а также наличие нецифровых услуг, направленных на улучшение опыта пользования сервисом.

компания отправляет товары покупателям почти в 190 стран. По состоянию на сентябрь 2014 года «годовые продажи товаров через сайт Farfetch превысили 167 миллионов фунтов стерлингов».

Бизнес-модель компании заключается в предоставлении онлайн канала дистрибьюции люксовой модной одежды и аксессуаров для бутиков и брендов-производителей. При этом, для бутиков и брендов применяются разные условия. На рисунке 2 представлено распределение добавленной стоимости между сторонами, где product cost – себестоимость товара, brand margin – наценка бренда, retail margin – наценка бутика (ритейлера), farfetch – наценка Farfetch.

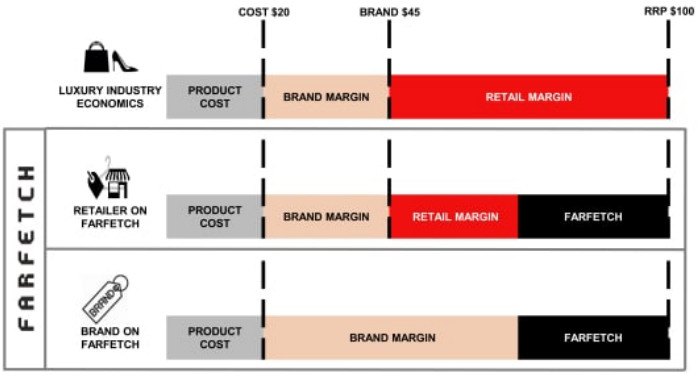

Рисунок 2. Распределение добавленной стоимости товаров

Помимо размещения на сайте предметов из бутика, платформа предлагает логистические услуги для транспортировки таких товаров.

Основной доход компании формируется за счет продаж с платформы, комиссии со сделок и размещения рекламы на сайте и в приложении. Обычно на платформе рекламируются бренды, публикации стилистов и экспертов, содержащие сезонные подборки образов.

Основную долю в структуре издержек компании имеют административные расходы в т. ч. зарплаты сотрудникам, обслуживание офисов и так далее. Помимо этого, компания тратит достаточно большое количество средств на рекламу и копирайт (в данном случае на право размещения бренда на платформе). Также компания ведет политику M&A: были приобретены, например, платформа для продажи кроссовок и материнская компания бренда Off-white[6].

Ключевыми партнерами компании являются: модные бутики по всему миру, бренды роскошной одежды, транспортные компании (DHL, UPS, FedEx) и платежные системы (Visa, MasterCard).

[6] Off-White - бренд одежды премиум класса, основанный в Милане в 2012 году.

Ценностным предложением компании является предоставление пользователям доступа к люксовым товарам со всего мира и предоставление онлайн платформы для бутиков.

Описанная бизнес-модель представлена в виде диаграммы Остервальдера на рисунке 3.

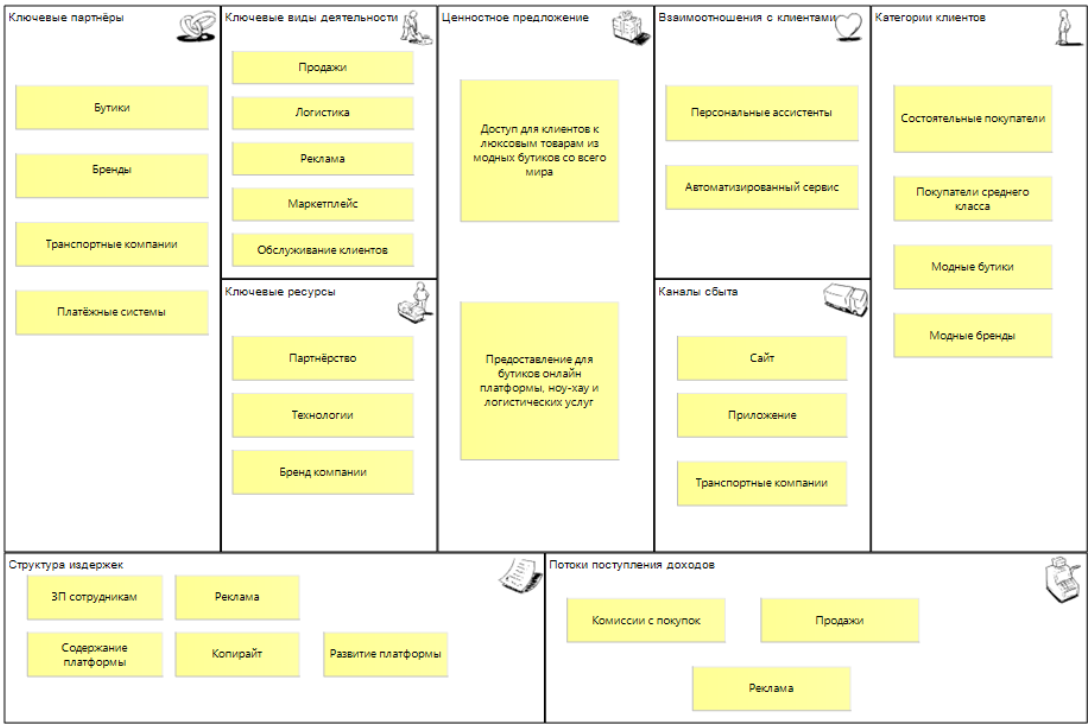

Рисунок 3. Диаграмма Остервальдера для Farfetch

В сентябре 2018 года Farfetch (FTCH.N) провела листинг на Нью-Йоркской фондовой бирже (NYSE), оценивая акции выше предполагаемого целевого диапазона, что привело к оценке более чем в 5,8 миллиарда долларов. IPO привлекло для компании $885 млн после выпуска 33,6 млн новых акций.

В декабре 2018 года Farfetch приобрела онлайн-платформу для кроссовок Stadium goods за 250 миллионов долларов.

В августе 2019 года Farfetch приобрела New Guard Group, материнскую организацию дизайнерского лейбла Off-white, за 675 миллионов долларов США. Сразу после покупки акции Farfetch упали более чем на 40%.

	Year ended December 31,				
	2016	2017	2018	2019	2020
	(in thousands, except per share data and AOV)				
Selected Other Data:					
Consolidated Group:					
GMV (1)	$585,842	$ 909,826	$1,407,698	$2,139,699	$ 3,187,014
Revenue	242,116	385,966	602,384	1,021,037	1,673,922
Adjusted Revenue (1)	193,605	311,784	504,590	893,077	1,460,694
Gross Profit	116,878	204,766	298,450	459,846	770,928
Gross Profit Margin	48.3%	53.1%	49.5%	45.0%	46.1%
Loss After Tax	$ (81,459)	$(112,275)	$ (155,575)	$ (373,688)	$(3,333,071)
Adjusted EBITDA (1)	(53,380)	(58,079)	(95,960)	(121,376)	(47,432)
Adjusted EBITDA Margin (1)	(27.6%)	(18.6%)	(19.0%)	(13.6%)	(3.2%)
Earnings Per Share ("EPS")	$ (0.43)	$ (0.50)	$ (0.59)	$ (1.21)	$ (9.75)
Adjusted EPS (1)	(0.27)	(0.39)	(0.38)	(0.56)	(0.66)
Digital Platform:					
Digital Platform GMV (1)	$573,174	$ 894,392	$1,392,103	$1,947,868	$ 2,759,476
Digital Platform Services Revenue	180,937	296,350	488,995	701,246	1,033,156
Digital Platform Gross Profit	111,762	196,581	291,706	371,913	560,206
Digital Platform Gross Profit Margin	61.8%	66.3%	59.7%	53.0%	54.2%
Digital Platform Order Contribution (1)	$ 63,381	$ 127,379	$ 194,411	$ 220,563	$ 361,419
Digital Platform Order Contribution Margin (1)	35.0%	43.0%	39.8%	31.5%	35.0%
Active Consumers (1)	652	936	1,382	2,068	3,024
Average Order Value - Marketplace (1)	$ 584	$ 620	$ 619	$ 608	$ 568
Average Order Value - Stadium Goods (1)	-	-	-	315	316
Brand Platform:					
Brand Platform GMV (1)	$ -	$ -	$ -	$ 164,210	$ 390,014
Brand Platform Revenue	-	-	-	164,210	390,014
Brand Platform Gross Profit	-	-	-	75,007	190,806
Brand Platform Gross Profit Margin	-	-	-	45.7%	48.9%

(1) See "Non-IFRS and Other Financial and Operating Metrics" below.

Рисунок 4. Консолидированный отчет о прибылях и убытках Farfetch по итогам 2020 года

Финансовая отчетность (например, представлен консолидированный отчет о прибылях и убытках компании за 2020 год на рисунке 4) компании позволяет нам заметить несколько тенденций: компания постоянно наращивает выручку (от года к году выручка компании растёт), новый бизнес компании - платформа для брендов - вырос с 2019 по 2020 год на 138% по выручке. Он также является более маржинальным, чем бизнес, связанный с реализацией товаров бутиков. Однако можно заметить, что один из наиболее значимых для инвестиционной оценки показателей компании, EBITDA[7], несмотря на его рост, остаётся отрицательным. В 2020 году у компании случились аномально высокие расходы, и прибыль до налогообложения более чем в 9 раз уменьшилась по сравнению с 2019 годом — это связано, как следует из годового отчета, с переоценкой обязательств компании по деривативам на её акции, и с фактическими затратами компании не связано.

По итогам 3 кварталов 2021 года компания получила положительную EBITDA, при этом выручка стала расти с меньшей скоростью (28% в 2021 году по сравнению с 49% в 2020) - это замедление привело к значительному снижению стоимости акций компании (на 72%), что

[7] Прибыль предприятия до вычета процентов, налогов, износа и амортизации.

свидетельствует об опасениях акционеров, что компания прекращает этап бурного роста, при этом оставляя инвесторов без дивидендов. В связи с этим ключевые акционеры требуют от компании повысить выручку, привлечь новых пользователей и выйти в другие сегменты рынка.

1.2.2 РЕКОМЕНДАТЕЛЬНАЯ СИСТЕМА

Коммерческий успех в сфере электронной коммерции во многом зависит от того, насколько удобна платформа для покупателя, удаётся ли ей удерживать его внимание, подталкивать к движению по каталогу, делать заказы. Исследования [1], [2] подтверждают, что внедрение рекомендательной системы для пользователей позволяет повысить бизнес-показатели: например, в статье Р. Лоурен, Г. Алмаси, М. Виверос и С. Дури [3] утверждается, что выручка вырастает на 1.8%. При этом, если учитывать не только прямые эффекты, то согласно другому исследованию [2] продажи в рекомендуемой категории выросли на 26%. Аналогичная ситуация с eBay – улучшение рекомендательной системы (с точки зрения повышения качества предсказания релевантности моделью) помогло повысить выручку на 6% [4], а внедрение – повысило выручку на 500% в отдельных разделах сайта [5].

Понимание данного контекста привело руководство Farfetch к принятию решения о внедрении рекомендательной системы в свои продукты. В 2019 году FARFETCH развернула «коробочное решение»[8], считавшееся на тот момент лучшей рекомендательной системой в отрасли. Система более 20 раз дорабатывалась, чтобы обеспечить достаточный уровень быстродействия и удобства для пользователя. Тем не менее работа Системы была неудовлетворительной, так как релевантность рекомендаций была не очень высокой, что объяснялось тем, что модель обучалась на данных других сегментов электронной коммерции, из-за чего не учитывала специфику индустрии – модные тренды, различие жизненных циклов продуктов и наличие аналогов. Релевантность оценивается следующим образом: для каждого рекомендованного товара определяется степень его визуальной схожести с просматриваемым товаром, соответствие по цене, соответствие пользовательскому предпочтению брендов (по шкале от 0 до 1) – модель имела среднюю релевантность в 0.45 (такая оценка релевантности была дана в статье Делджу Ю. [6]).

Следующим этапом развития рекомендаций на платформе была разработка полноценного решения под нужды компании, получившего

[8] Коробочное решение – здесь: не гибкое решение, разработанное сторонним производителем, предоставляющее малый спектр возможных настроек параметров.

название Precog. Данная система имела в своем основании реализацию алгоритмов, построенных на ансамблях решающих деревьев. Кроме того, специально для Precog были написаны API и ETL. Сам алгоритм рекомендаций работает на основе маршрута пользователя по каталогу и дополнительной информации о товарах, предоставленной производителем. То есть данная модель является user-based5, что в свою очередь решает часть проблем с низкой релевантностью рекомендаций, так как теперь алгоритм хорошо учитывает вкусы пользователей, при этом появились новые проблемы:

- новые пользователи получают нерелевантные рекомендации, так как о них у системы нет подходящей истории запросов и покупок;
- система придает наибольшее значение пользовательским характеристикам (сессия, история покупок, страна пользователя, прошлая страница), из-за чего в рекомендации попадают товары только одного бренда даже при наличии более похожих товаров у других брендов.

На данный момент система имеет среднюю релевантность рекомендаций в 0.55 [6].

Для руководства компании очевидно, что был выбран неверный подход, и такая средняя релевантность не удовлетворяет пользователей, что приводит к низкой отдачи от внедрения рекомендательной системы. В связи с этим у компании есть запрос на переработку системы, о чем будет подробнее указано в следующем подразделе.

1.3 ОПИСАНИЕ БИЗНЕС-КЕЙСА

1.3.0 РЕЗЮМЕ ПОДРАЗДЕЛА

В данном подразделе резюмируются результаты исследования предметной области и изучения документов, данных и других открытых ресурсов, которые использовались в данном разделе отчета.

Мы проанализировали финансовую отчетность[9], публичную управленческую отчетность[10], пресс-релизы[10] заседаний Совета Директоров и Общего собрания акционеров компании для того, чтобы понять, какие бизнес-задачи стоят перед компанией. Понимание того, что компании важно внедрить новую рекомендательную систему товаров для покупателей, было получено благодаря изучению научных исследований [1-5]. Проблемы текущей рекомендательной системы в компании были выявлены из управленческой

[9] Отчетности, доступные на сайте отношений с инвесторами Farfetch. URL - https://www.farfetchinvestors.com/financials-and-filings/financial-reports/.

отчетности и описания истории рекомендаций товаров на сайте компании в блоге компании [7, 8]. Оттуда же мы узнаем про ожидания компании.

1.3.1 БИЗНЕС-ЗАДАЧИ

Бизнес-задачи компании продиктованы требованиями акционеров, которые теряют свой капитал из-за понижения стоимости компании на фоне замедления темпов роста выручки и прибыли.

Бизнес-задачи, на решение которых проект окажет непосредственное влияние:

- Рекомендовать покупателям релевантные для них товары;
- Сократить среднее время между поиском товара и оформлением заказа;

Бизнес-задачи, частично закрываемые проектом:

- Удерживать пользователя на сайте (в приложении) как можно дольше до принятия им решения о заказе;
- Предоставлять лучшие в индустрии продукты и сервисы;
- Привлекать новых покупателей.

1.3.2 ОЖИДАНИЯ БИЗНЕСА ОТ ПРОЕКТА

Компания стремительно развивается последние 5 лет, однако акционеры чувствуют, что рост замедляется, а сегмент рынка начинает себя исчерпывать, что выражается в резком падении капитализации компании на протяжении 2021 года. Среда является высококонкурентной, и скорость инноваций играет важную роль в том, кто является лидером на рынке. В связи с этим акционеры хотят видеть инновационное развитие платформы и привлечение новых пользователей. Руководство компании считает, что решить эти задачи поможет развитие рекомендательной системы, учитывая недостатки используемого сейчас решения:

1. обычные подходы к рекомендациям на основе существующих решений, которые, как правило, хорошо работают для музыки, фильмов и даже электронной коммерции, неадекватны в контексте роскошной моды. Каталог Farfetch далек от идеала для совместных рекомендаций от товара к товару, поскольку жизненный цикл продукта либо слишком короток (товары покупаются, как только они выходят в продажу или даже до того, как они появятся), либо слишком долог (некоторые вневременные культовые предметы остаются актуальными на протяжении большого промежутка времени);

2. система не рекомендует нишевые товары, так как не успевает подстроиться под пользователя из-за быстрой смены трендов в моде;
3. рекомендации часто содержат товары просматриваемого бренда, так как в текущей версии системы именно логотип бренда на фото имеет наибольшую значимость в определении похожих товаров;
4. рекомендации не содержат товары, являющиеся аналогом просматриваемого - если бы эта проблема была устранена, можно было бы ожидать рост продаж, так как пользователи могли бы быстрее находить брендовую одежду по приемлемой для себя цене.

Все описанные выше проблемы влияют на то, что решение имеет среднюю релевантность рекомендаций на уровне 55%.

Компания ожидает получить в результате реализации проекта систему, которая могла бы решить указанные проблемы и добиться релевантности на уровне 70% (то есть 7 из 10 товаров в рекомендациях должны быть релевантными пользователю) в срок не более 4 календарных месяцев и затратами не более 1 000 000 (одного миллиона) рублей.

Методы подсчета релевантности модели подробнее описаны в разделе 2.2.3.

1.4 КОНЦЕПЦИЯ РАЗРАБАТЫВАЕМОЙ МОДЕЛИ РЕКОМЕНДАТЕЛЬНОЙ СИСТЕМЫ

Изучив предметную область, мы подготовили концепцию модели рекомендательной системы, в соответствии с которой будем производить работу с данными и разработку модели на основе методов машинного и глубинного обучения.

Для устранения проблем, которые имеет система, используемая компанией сейчас необходимо изменить подход к работе модели – в данной предметной области важнее, чтобы рекомендация была релевантной относительно просматриваемого товара, а не истории пользовательского опыта и бренда – именно поэтому разрабатываемая модель будет обучаться на данных о товарах, чтобы научиться определять среди них похожие.

Разрабатываемая модель будет состоять из четырех концептуальных блоков, их взаимодействие представлено на рисунке 5, поток данных изображен на рисунке 6, а подробное описание ниже:

1. Генерация признаков
2. Генерация классов
3. Классификация входящих данных
4. Ранжирование рекомендаций

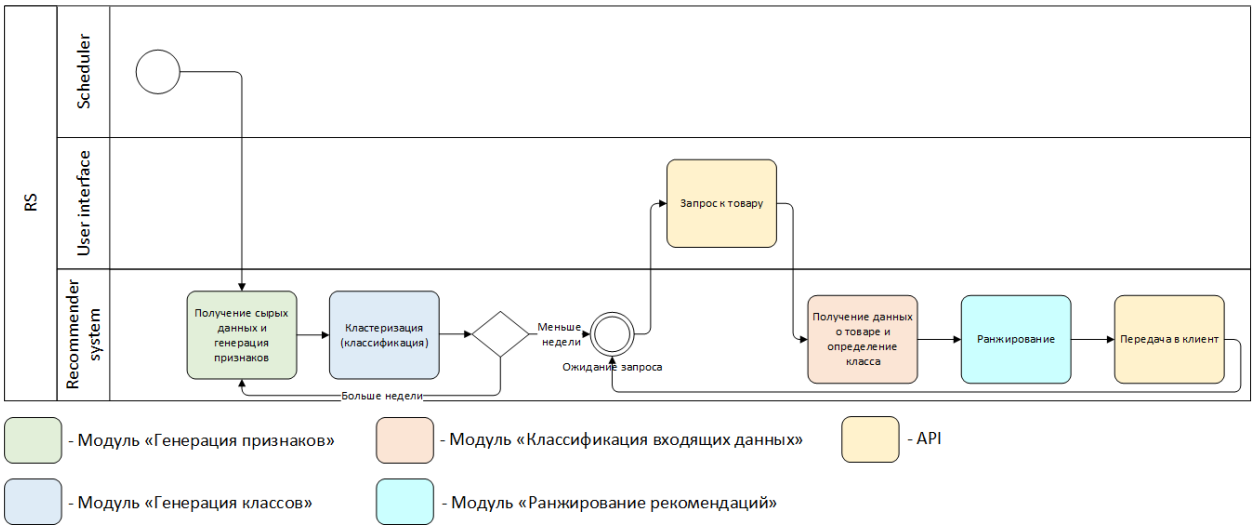

Рисунок 5. Описание концепции работы разрабатываемой пилотной рекомендательной системы

На рисунке 5 выше изображено взаимодействие разных концептуальных блоков пилотного решения, разрабатываемого в рамках данного проекта, в то время как рисунок 6 описывает процесс движения данных внутри системы.

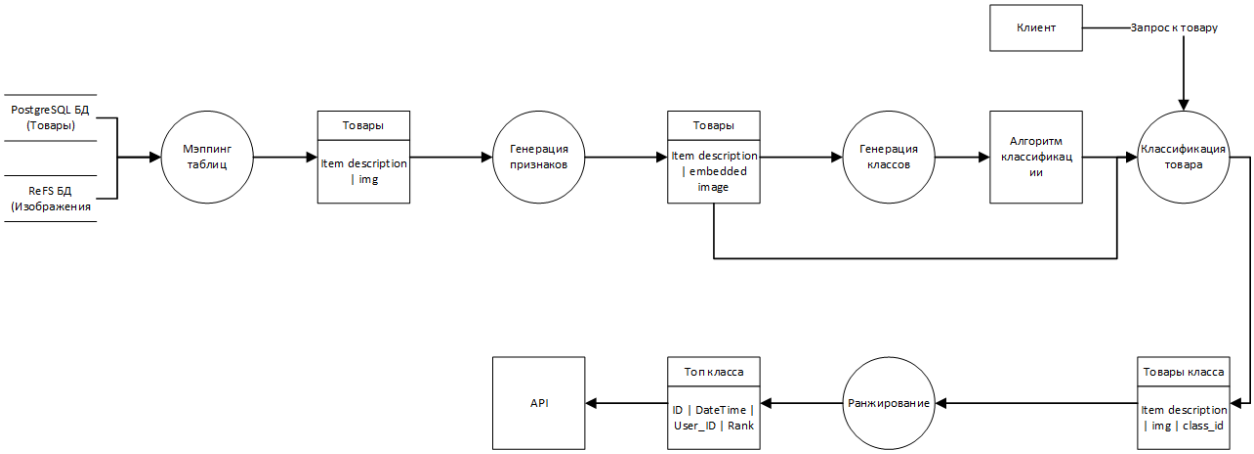

Рисунок 6. Data flow diagram для ML-модели RTB

Блок генерации признаков получает на вход набор данных о товарах (id товара, изображение, категория товара, подкатегория товара, целевой пол товара, наименование товара, наименование бренда, цена товара, доступные размеры) (рис. 6). После чего для каждого объекта в наборе с применением сверточных нейронных сетей (ResNet, Inception, VGG, MobileNet)

генерируются признаки на основе изображений (выявляются паттерны, формы и прочие признаки, которые можно извлечь из изображения). Затем данные признаки объединяется с теми, что уже были в начале. Выходной результат модели – данные с новыми признаками по изображениям.

Блок генерации классов получает на вход выходной результат прошлого блока (рис. 6), затем данные будут поделены в соответствии с их подкатегориями на отдельные наборы. Внутри каждого отдельного набора будет решаться задача кластеризации на основе признаков, сгенерированных в прошлом блоке. Качество кластеризации будет сравниваться с результатом модели K-means по среднему расстоянию в кластерах в качестве метрики. Выходной результат модели – вновь объединенные данные в один набор с новым признаком – кластером, в который входит товар. Кластеризация позволит сгруппировать товары по похожести.

Блок классификации входящих данных получает на вход выходной результат предыдущего блока (рис. 6). В данном блоке данные будут разделены на обучающую и валидационную выборки, также будут разделены таргет и признаки. На обучающей выборке мы обучим алгоритм классификации на основе градиентного бустинга. Качество будем проверять на валидационной выборке, используя метрику F1-меры[10]. Нам необходимо иметь значение данной метрики на уровне не менее 90%. Выходной результат – модель для классификации товара.

Блок ранжирования рекомендаций получает на вход модель для классификации товара, модель для генерации признаков, метаданные пользователя (рис. 6). В данном блоке мы получаем класс товара, затем ранжируем объекты в данном классе по следующим параметрам:

- Вероятность похожести – оценка вероятности получается с применением логарифмической функции потерь на основе расстояния между векторами объектов.
- Цена товара
- Соответствие бренда товара предпочтения пользователя.

Качество всей модели оценивается метрикой средней релевантности pFound[11] – требуемое качество должно быть 70%.

1.5 ИСТОЧНИКИ ДАННЫХ

Предполагается использование одного из двух источников:

[10] F1-мера – метрика, используемая для оценки качества классификации, требующая одновременно высокой точности и полноты от модели

[11] Метрика релевантности, разработанная Яндекс. URL - https://catboost.ai/en/docs/references/pfound

- Исходный датасет, предоставляемый Farfetch[12];
    - Следующие признаки присутствуют в датасете:
        - Доступные размеры
        - ID бренда и название бренда
        - ‘Пол’
        - Кастомизируемость
        - Коллекция
        - ID продавца (бутика)
        - Цена изначальная
        - Цена актуальная
        - Короткое описание
        - Количество товара на складе
        - Изображение товара
- Данные напрямую из каталога на сайте[13], полученные с помощью парсинга.

Исходный датасет предлагает достаточно большое количество признаков, но среди них нет категории товара или подкатегории в случаях, когда такая подкатегория существует. Несмотря на это, такие данные очевидно являются атрибутами таблиц с товарами в хранилище данных Farfetch. В связи с этим ML решение не должно предварительно распределять товары по кластерам категорий, следовательно исходный датасет будет обогащен дополнительным признаком «категория», либо будут использованы менее детальные, но достаточные для обучения модели данные, полученные в ходе парсинга каталога Farfetch. В случае использования последнего, следующая информация о товарах может быть получена:

- ID товара
- Категория товара
- «Пол»
- Наименование товара
- Наименование бренда
- Цена товара

[12] Kaggle, Farfetch Listings Dataset. URL - https://www.kaggle.com/alvations/farfetch-listings

[13] Каталоги одежды на ресурсе Farfetch. URL: https://www.farfetch.com/ru/shopping/men/items.aspx; https://www.farfetch.com/ru/shopping/women/items.aspx

- Изображение товара
- Доступные размеры

Для обучения модели(-ей) могут использоваться все признаки, полученные в ходе парсинга. Часть из них будет актуальна исключительно для ранжирования рекомендаций (бренд, цена, размеры).

1.5.1 ПАРСИНГ

Данные из каталога были получены в процессе постраничного прохода по каждой категории каталога и записи текста под тэгом div с соответствующим классом. Ссылки на изображения находятся в соседнем классе под тэгом meta в поле “contents”. Также, к изображениям можно было обращаться по тэгу img, но эта часть разметки обладает свойством loading=”lazy”, поэтому без использования вэб-драйвера получить содержимое невозможно. В свою очередь использование Selenium (вэб-драйвер библиотеки) сильно замедляет процесс обращения к страницам. Итого, парсинг осуществлялся на языке Python, с использованием библиотек requests, BeautifulSoup, time, csv.

1.5.2 КОНЦЕПЦИЯ ПОЛУЧЕНИЯ ДАННЫХ РЕАЛИЗУЕМОЙ МОДЕЛИ

На этом этапе мы также определяем, как рекомендательная система будет получать эти же данные в лайв-режиме. Предполагается, что, когда пользователь запрашивает страницу с товаром из каталога, эти же данные поступают на вход рекомендательной системе (параллельным запросом), которая в свою очередь отбирает визуально похожие товары, ранжирует их и в ленивом режиме передает их запросом POST в соответствующую часть разметки.

1.6 ИДЕНТИФИКАЦИЯ СТЕЙКХОЛДЕРОВ

Таблица 2. Реестр стейкхолдеров

Идентификационная информация	**Оценочная информация**			
Стейкхолдер	**Основные ожидания**	**Основные требования**	**Настрой по отношению к проекту**	**Уровень влияния на проект**
CEO компании (генеральный директор)	Финансовая выгода для компании	Проект должен быть выполнен в установленные сроки	Положительный	Высокий
Крупные бренды	Увеличение продаж	Проект не должен снизить продажи продукции крупных брендов; Рекомендательная система не должна пренебрегать карточками крупных брендов	Негативный	Высокий
Средние бренды	Финансовая прибыль; повышение узнаваемости бренда	Размещение платформой их товаров	Положительный	Средний
Пользователи	Более удобная система поиска подходящих вещей	Появление более релевантных рекомендаций на платформе	Положительный	Низкий
Финансовый директор	Увеличение выручки компании	Расходы по проекту не должны превышать установленную заказчиком сумму; Срок окупаемости не должен превышать 2 лет; IRR> WACC.	Нейтральный	Высокий
Директор команды коммуникации с брендами	Увеличение количества брендов на платформе	Рекомендательная система не должна снижать выручку крупных брендов	Положительный	Низкий
Platform Partner Success Manager	Увеличение количества брендов и бутиков на платформе		Положительный	Низкий
Директор отдела логистики	Расширение транспортной сети компании		Положительный	Низкий
Директор ИТ-отдела	Рекомендательная система должна работать как минимум не хуже, чем существующая	Рекомендательная система должна быть развернута на имеющемся оборудовании	Негативный	Средний
Бутики	Увеличение продаж через платформу		Положительный	Высокий

1.6.1 ПОЛИТИКА ВЗАИМОДЕЙСТВИЯ СО СТЕЙКХОЛДЕРАМИ

Особое внимание нужно уделять стейкхолдерам с негативным отношением к проекту и высоким уровнем влияния на проект (выделены красным). Основной задачей проводимой политики по отношению к таким стейкхолдерам будет, несомненно, выстраивание эффективной коммуникации и дальнейшее плодотворное сотрудничество с участниками диалога. К таким стейкхолдерам относятся: крупные бренды и директор ИТ-отдела. Негативное отношение к проекту со стороны крупных брендов возникает в первую очередь из-за того, что на платформе будут появляться новые менее известные бренды, которые возможно смогут привлечь аудиторию более крупных, что в свою очередь приведет к снижению количества продаж и уменьшению выручки с платформы. Чтобы убедить их в том, что опасения с их стороны напрасны, мы готовы провести A/B тестирование, предоставив доступ к новой рекомендательной системе лишь половине пользователей, не изменив рекомендации другой. На основе полученных статистических данных будет принято решение о дальнейших действиях. Если результаты работы новой системы значительно уменьшат выручку бренда, будет предложено пересмотреть концепт внедряемого решения, добавив еще одну строку для рекомендаций с названием “Похоже на то что вы искали”, а вещи бренда будут в строке с названием “От этого же бренда”. Следующий стейкхолдер с негативным отношением к проекту - директор ИТ-отдела. Одна из главных его претензий заключается в том, что он не хочет менять пайплайн[14] работы с данными (изменение ETL процедур, создание новых витрин данных). С нашей стороны будет правильней минимизировать вмешательство в существующую структуру баз и хранилищ данных.

Вторая категория стейкхолдеров, выделенная желтым, также требует не меньшего внимания. Основная задача проводимой политики взаимодействия - склонить к положительному отношению к проекту, не допустив перехода от нейтралитета к отрицательному настрою. Финансовый директор относится к стейкхолдерам такого типа по ряду причин. Во-первых, его отношение к предполагаемым финансовым показателям проекта достаточно скептично. Во-вторых, этот стейкхолдер считает, что в сфере высокой моды рекомендательные системы не играют определяющую роль и что внедрение нашего решения не особо повлияет на сложившуюся в компании ситуацию. В-третьих, финансовый директор не уверен в том, что нужно делать инвестиции в разработку рекомендательной системы из-за того, что опасается возможности повторного переделывания системы после нашей работы.

[14] Последовательность операций с одними и теми же исходными данными.

Основная наша деятельность по изменению его отношения к проекту будет заключаться в проведении исследования на основе уже опубликованных показателей внедренных рекомендательных систем, в том числе в индустрии моды.

Стейкхолдеры, выделенные зеленым цветом, готовы к плодотворному сотрудничеству, однако не стоит забывать про их интересы, требования и ожидания. Мы планируем получать обратную связь, делиться результатами этапов проекта, демонстрировать прогресс и будем актуализировать их требования. Для пользователей будет доступно бета-тестирование обновленного приложения, для средних брендов - тестирование рекомендательной системы на их продукции.

1.7 ВЫВОДЫ ПО РАЗДЕЛУ

Результатом этого этапа работ стало понимание текущих проблем в рекомендательной системе, получение данных, определение концепта нашего решения и его отличий от существующей системы, готовая концепция решения, включающая в себя как алгоритмы классификации, так и подход к ранжированию членов класса. Для алгоритмов определены метрики качества, подробное описание которых приводится в разделе 2.

Для бизнеса ключевым показателем качества решения является метрика релевантности подаваемых объектов, которой, как было ранее упомянуто, будет pFound и целевым значением будет 70% (0.7).

Также, определение стейкхолдеров и проблем бизнеса позволит нам эффективнее вести диалог с бизнесом при реализации системы, в том числе учесть пожелания и опасения стейкхолдеров касательно результата.

Итого, на данный момент мы имеем полное и необходимое представление о структуре работ, бизнеса и решения для начала разработки.

РАЗДЕЛ 2. ФОРМАЛИЗАЦИЯ ФУНКЦИОНАЛЬНЫХ И НЕФУНКЦИОНАЛЬНЫХ ТРЕБОВАНИЙ

2.1 НЕФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ К СИСТЕМЕ

Ниже на рисунке 7 представлена часть ИТ-архитектуры компании FarFetch, косвенное и прямое взаимодействие с которой предполагается в рамках внедрения разрабатываемой рекомендательной системы. На схеме предусмотрено отображение новых объектов (зеленый), объектов инфраструктуры, от которых предлагается отказаться (красный) и объектов, в которые будут вноситься изменений (голубой пунктир). Схема инфраструктуры построена на основе изучения публичной информации о технологиях компании в блоге[15], который она ведёт. Также на схеме под блоком real-time bidding (RTB) имеется в виду описанная в разделе 1.4 модель.

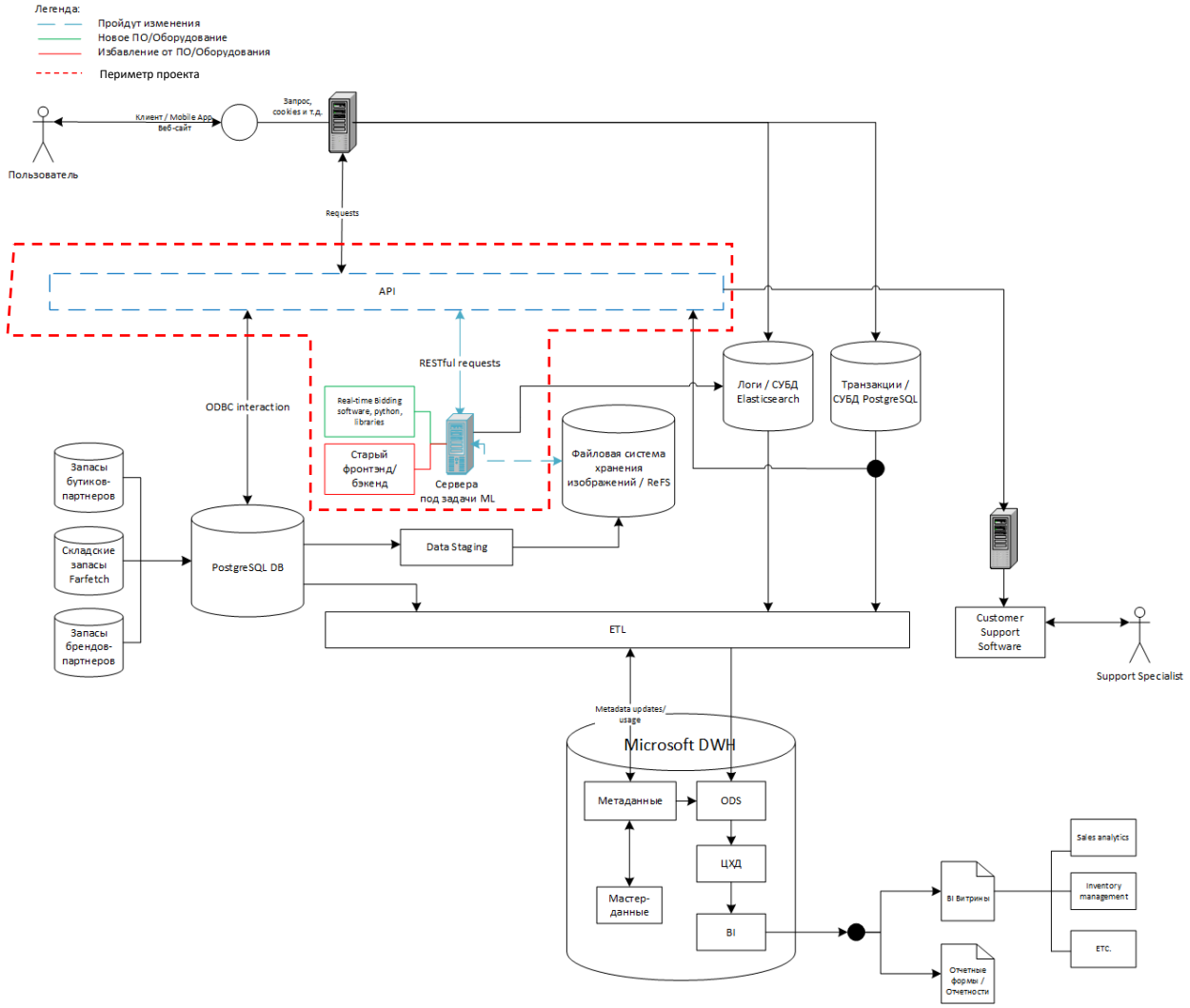

Рисунок 7. ИТ-архитектура FarFetch и место разрабатываемой системы в ней

[15] - Блог о технологиях Farfetch https://www.farfetchtechblog.com/en/blog/post/architecture-farfetch/#:~:text=At%20Farfetch%20we%20don't,%2C%20communication%2C%20planning%20and%20execution

2.1.1 ТРЕБОВАНИЯ К СПОСОБАМ И СРЕДСТВАМ СВЯЗИ ДЛЯ ИНФОРМАЦИОННОГО ОБМЕНА МЕЖДУ КОМПОНЕНТАМИ СИСТЕМЫ

Программно-технические средства компонент системы должны соответствовать стандартам обмена с использованием протокола TCP/IP.

2.1.2 ТРЕБОВАНИЯ К ХАРАКТЕРИСТИКАМ ВЗАИМОСВЯЗЕЙ СОЗДАВАЕМОЙ СИСТЕМЫ СО СМЕЖНЫМИ СИСТЕМАМИ

Программно-технические средства разрабатываемой системы должны соответствовать стандартам сети Интернет и поддерживать прием-передачу данных по протоколу HTTPS. Физический сервер, на котором размещаются программные модули автоматизированной информационной системы, должен иметь постоянное подключение по сети Интернет по протоколам TCP/IP.

Создаваемая Система должна интегрироваться с OLTP базой данных на основе СУБД PostgreSQL через ODBC подключение, что реализуется в API разрабатываемой системы. API должно позволять пользовательским интерфейсам (приложения, веб-сайты) получать информацию о рекомендованных товарах.

2.1.3 ТРЕБОВАНИЯ К РЕЖИМАМ ФУНКЦИОНИРОВАНИЯ СИСТЕМЫ

Разрабатываемая информационная система предназначена для работы в круглосуточном режиме с показателем uptime в 99,99%. Максимально допустимое время простоя системы не должно превышать 9 секунд в день.

В нештатной ситуации система должна сохранить данные, сделать резервную копию важных данных, а время до восстановления ее работы не должно превышать 1 минуты. В нештатной ситуации должна резервироваться логика работы системы.

2.1.4 ТРЕБОВАНИЯ К ИНФОРМАЦИОННОМУ ОБЕСПЕЧЕНИЮ

Система использует различные источники данных и системы для их хранения для функционирования:

1. Файловая система типа ReFS хранения данных, используемых для обучения и применения алгоритмов машинного обучения.
2. Нереляционная база данных под управлением СУБД Elasticsearch для хранения и анализа логов системы.
3. Реляционная база данных под управлением СУБД PostgreSQL, хранящая данные о транзакциях (результатах работы системы, сессиях пользователей).

4. Реляционная база данных Farfetch под управлением СУБД PostgreSQL, хранящая данные каталога.

Данные между компонентами системы передаются через единый узел – процедуры ETL, что обеспечивает полноту, целостность и непротиворечивость данных, используемых различными компонентами системы.

Для обеспечения сохранности информации в аварийных ситуациях в Системе должны быть предусмотрены средства бесперебойного питания, дублирования и резервного копирования (архивирования) информации.

Средства бесперебойного питания должны обеспечивать работоспособность КСА при импульсных помехах и перерывах в электропитании на время до 15 минут.

Процедуры резервного копирования и программная поддержка источников бесперебойного питания должны обеспечиваться средствами сетевого программного обеспечения и СУБД.

Система должна контролировать корректность входной информации, а также проверять логическую целостность информации в базах данных при выполнении любой прикладной операции.

В Системе должно быть предусмотрено резервное копирование (архивирование) информации из базы данных со следующей периодичностью:

- оперативное ежедневное архивирование текущей информации с хранением архивных копий за три последних дня работы;
- еженедельное полное архивирование данных и хранение их в течение месяца;
- ежемесячное архивирование всей базы данных и долговременное хранение этих копий.

Перед созданием резервной копии системного диска должна контролироваться корректность информации на диске.

При создании резервной копии файловой системы вместе с содержимым файлов должны сохраняться и данные о владельце файла, группе файлов и их маска доступа.

Система должна обеспечивать ведение электронного архива по всей используемой информации. При этом правила передачи информации в архив и ее удаления должны определяться параметрами Системы для каждого типа информации.

Система должна обеспечивать при работе с архивом следующие сервисные функции:

- администрирование архива и регенерация магнитных носителей;

- поддержка прав доступа к архивной информации;
- обработке стандартных и нестандартных запросов к архивной информации.

Система должна вести журналы регистрации всех выполненных действий, связанных с изменением информации и иметь процедуры обработки (ROLLFORWARD), позволяющие обновить состояние системы с последней резервной копии до текущего состояния.

Данные пользователя, собираемые системой, должны быть защищены. Данные пользователя, собираемые системой, должны храниться не более срока, установленного Политикой конфиденциальности Farfetch.

2.1.5 ТРЕБОВАНИЯ К УДОБСТВУ ИСПОЛЬЗОВАНИЯ СИСТЕМЫ

Система должна стабильно и корректно работать на мобильных устройствах под управлением операционных систем iOS (13.0 и выше), Android (6.0 и выше), а также в браузерах мобильных устройств Chrome и Safari, и браузерах Chrome, Safari, Yandex Browser, Firefox на персональных компьютерах под управлением MacOS, Windows.

Дизайн-код развернутой в приложении или на сайте системы не должен противоречить дизайн-коду сайта.

Система должна быть локализована на все доступные в приложении и на сайте языки (русский, английский, арабский, голландский, датский, испанский, итальянский, корейский, немецкий, португальский, китайский, французский, шведский, японский).

2.1.6 ТРЕБОВАНИЯ К ПРОИЗВОДИТЕЛЬНОСТИ СИСТЕМЫ

Система[16] должна сохранять корректную и безотказную работоспособность в случае, если число обслуживаемых транзакций не более 10 000 в секунду[17].

Данная производительность должна быть обеспечена при использовании серверного оборудования со следующими характеристиками:

- CPU с тактовой частотой не менее 3.0 ГГц, числом ядер не менее 32 (не менее 64 потоков).
- GPU NVIDIA TITAN RTX с объемом видео-памяти не менее 24 Гб типа GDDR6.

[16] Указано требование к разрабатываемой системе RTB, так как производительность смежных систем находится вне рамок проекта

[17] Пиковая нагрузка рассчитана на основе анализа трафика сайта https://www.similarweb.com/ru/website/farfetch.com/#overview , информации о пиковых нагрузках https://newrelic.com/blog/how-to-relic/black-friday-observability-farfetch

- Оперативным запоминающим устройством с объёмом памяти не менее 32 Гб по стандарту DDR4.
- Твердотельным постоянным запоминающим устройством с объёмом памяти не менее 512 Гб.

2.1.7 ТРЕБОВАНИЯ К ПО И ПЛАТФОРМАМ

Для разработки системы не выдвигаются специальные требования и ограничения по использованию платформ и ПО.

Специальное программное обеспечение (СПО) должно базироваться на принципах модульности и концепции "открытых систем", предусматривающих возможность независимой разработки и включения в систему функциональных модулей, а также обеспечивающих возможность информационного взаимодействия с другими системами. При этом концепция "открытых систем" предполагает независимость разрабатываемых программных средств от используемых аппаратных и системных платформ, что поддерживается лучшими современными системными программными средствами (операционными системами и СУБД), использующими стандарты открытых систем.

Система должна корректно поддерживать многопользовательскую работу. Это означает, что должны быть задействованы механизмы блокировок, распознаваться ситуации deadlock, выдаваться необходимые сообщения пользователю, не допуская при этом искажения информации или выхода системы из рабочего состояния.

СПО должно быть "прозрачным" для пользователя. Это означает, что пользователь должен одинаково работать со всеми доступными ему данными и функциями независимо от их физического размещения и технических способов доступа к ним. Подобные проблемы должны решаться системой на нижнем уровне.

2.2 ФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ К СИСТЕМЕ

2.2.0 ФУНКЦИОНАЛЬНЫЕ БЛОКИ СИСТЕМЫ

Разрабатываемая система должна быть представлена в виде следующих функциональных блоков со специфическим набором функций:

1. RTB (real-time bidding) – серверное ПО, отвечающее за исполнение алгоритмов машинного обучения и предоставления клиенту набора рекомендаций (включает в себя модель, концепция разработки которого представлена в разделе 1.4).

2. API – дополнительное ПО, обеспечивающее взаимодействие визуальных объектов клиента (веб-сайт или мобильное приложение) с RTB и базой данных каталога Farfetch.

2.2.1 ТРЕБОВАНИЯ К ПРОЦЕССУ ПОДБОРА ТОВАРОВ В КЛИЕНТЕ И ВЗАИМОДЕЙСТВИЮ С RTB И API

Согласно требованиям заказчика и согласованной архитектуре пилотного решения, была сформирована схема движения данных в разрабатываемой системе – рисунок 8.

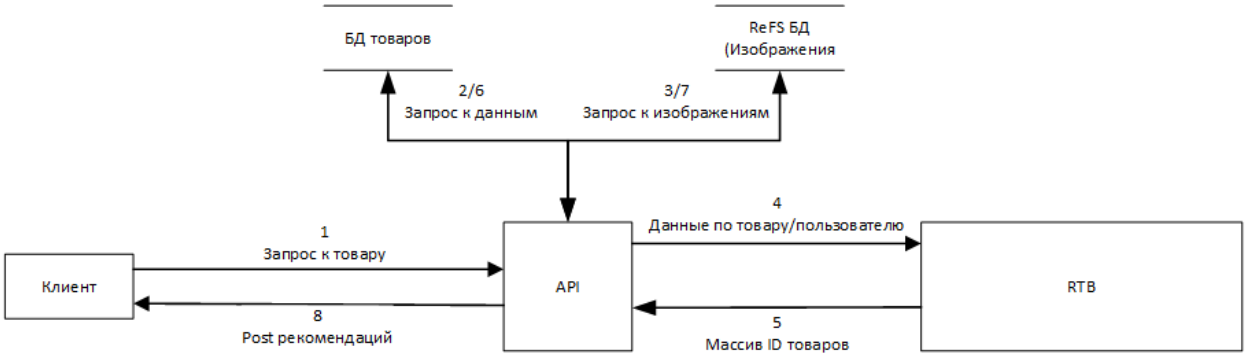

Рисунок 8. DFD разрабатываемого пилотного решения

Клиент реализует контейнер для отображения визуального объекта подборки рекомендованных товаров.

Контент подборки рекомендованных товаров должен быть динамическим, то есть изменяться в результате получения ответа от сервера, а также взаимодействия с ним пользователя.

Клиент передает API информацию о пользователе и открытом товаре в виде REST запроса, конфигурация которого будет уточнена на этапе формирования ТЗ.

API передает RTB информацию о пользователе и открытом товаре (в том числе его изображение для работы алгоритма классификации и прочую информацию, необходимую для ранжирования рекомендаций) в виде REST запроса.

RTB возвращает API массив уникальных идентификаторов товаров, порядок которых соответствует порядку показа объектов в контейнере клиента, в виде REST запроса.

API, получив ответ от RTB, забирает из базы данных товаров (путём подключения к ней через ODBC соединение) информацию о товарах, необходимую для отображения объектов в визуальном контейнере клиента (название, бренд, краткое описание, цена, доступные размеры).

API возвращает клиенту наполнение контейнера. Наполнением контейнера является массив товаров (объектов класса, содержащих необходимые атрибуты для отображения).

Передача данных между системами производится в одном формате. Передаваемые данные должны быть зашифрованы.

В случае некорректного ответа API клиент должен корректно отобразить другие элементы интерфейса

Клиент должен корректно обрабатывать ошибки ответа от API, в связи с этим API должно давать следующие ответы при соответствующих ошибках:

- Ошибка некорректного запроса от клиента:
  - Несуществующий товар
  - Несуществующая категория/подкатегория
  - Несуществующий клиент
  - Некорректный формат запроса
- Ошибка подбора рекомендаций:
  - Превышено время ожидания от RTB
  - Недостаточный размер изображения
  - Не удалось подобрать рекомендации
- Ошибка ответа от каталога:
  - Превышено время ожидания от БД
  - Отсутствует информация о товаре

2.2.2 ТРЕБОВАНИЯ К ФУНКЦИЯМ СИСТЕМЫ RTB

Система RTB должна выполнять следующие функции, в соответствии с концепцией модели машинного обучения рекомендательной системы:

1. Периодическое дообучение на основе обновления данных каталога Farfetch. Происходит раз в 2 недели – новые позиции каталога добавляются в изначальную выборку, на которой система применяет алгоритм кластеризации, а затем дообучение проходит алгоритм классификации.
2. Кластеризация объектов внутри категория по принципу наибольшей близости по параметрам.
3. Генерация признаков объектов по изображениям объектов.
4. Классификация объектов по классам похожести.
5. Ранжирование объектов по релевантности.

2.2.3 МЕТРИКИ КАЧЕСТВА ЭЛЕМЕНТОВ RTB

Данный раздел описывает метрики качества, используемые для оценки, валидации и обучения алгоритмов машинного обучения, используемых в концептуальных блоках RTB, обозначенных в разделе 1.4.

Для блока «Генерация признаков» согласно постановке задачи – наиболее точное и полное извлечение признаков из изображений – может быть оценено с использованием базовой модели машинного обучения для классификации – логистической регрессией (которая и будет являться бейзлайном для блока 3 – классификатора): на основе полученных данных мы обучим линейную регрессию и оценим качество её работы на валидационной выборке следующими метриками:

- Precision – точность классификации (процент правильно отнесенных к классу объектов от всех объектов, отнесенных к данному классу) для каждого класса; Рассчитывается по формуле ниже:

$$Precision = \frac{TP}{TP + FP}$$

- Recall – полнота классификации (доля от всех объектов данного класса объектов, отнесенных к данному классу); Формула для расчета:

$$Recall = \frac{TP}{TP + FN}$$

- Accuracy – доля правильных ответов модели. Ниже формула для расчета метрики:

$$Accuracy = \frac{TP + TN}{TP + \ FP + TN + FN},$$

где TP-верно отнесенные к классу объекты, FP-ошибочно отнесенные к классу объекты, TN-верно не отнесенные к классу объекты, FN-неверно не отнесенные к классу объекты.

Данные метрики были выбраны, так как являются наиболее частым способом оценки качества мультиклассовой классификации, а также позволяют выяснить, как качество распределено между разными классами, и где модель ошибается больше всего.

Для блока «Генерация классов», задача которого формулируется как кластеризация объектов на основе признаков, качество можно оценить путем сравнения результатов с результатами базовой модели – k-means, являющейся бейзлайном для данного блока. В качестве метрики, по которой будем сравнивать, используем среднее расстояние от центра класса, которое будем вычислять через формулу евклидового расстояния. Ниже приведена формула для подсчета расстояния:

$$D = \sqrt{\sum_{i=1..N} (A_i - B_i)^2}$$

где A, B – векторы, между которыми ищем расстояние, N – размерность пространства.

Следующий блок – блок «Классификация входных данных» - как уже было сказано выше, будет оцениваться в сравнении с бейзлайном – логистической регрессией по тем же метрикам качества, что и блок 1, но добавятся следующие метрики:

- F1-мера – геометрическое среднее между точностью и полнотой, соответственно позволяет судить об обеих метриках одновременно.

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall}$$

Блок «Ранжирование», решающий задачу алгоритмического ранжирования на основе нескольких параметров, оценивается метрикой pFound, которая вычисляется следующим образом:

$$pFound = \sum_{i=1}^{N} pLook_i * pRel_i,$$

где $pLook_i$ – вероятность просмотреть i-й объект из списка рекомендаций, $pRel_i$ – вероятность того, что i-й объект окажется релевантным пользователю, в качестве которой используем оценку релевантности. Вероятность просмотра i-го объекта вычисляется по следующей формуле:

$$pLook_i = pLook_{i-1} * (1 - pRel_{i-1}) * (1 - pBreak),$$

где $pLook_{i-1}$ – вероятность того, что пользователь посмотрит (i-1)-ю позицию; $1 - pRel_{i-1}$ – вероятность неудовлетворения предыдущей рекомендацией, $1 - pBreak$ – вероятность того, что он не остановится по независящим от нас причинам.

2.3 ВЫВОДЫ ПО РАЗДЕЛУ

На данном этапе работы мы изучили подходы к решению и метрикам, широко применяемых в сфере рекомендательных систем [9-11], на основе чего сформулировали список требований к системе, как функциональных, так и нефункциональных. Это поможет нам приступить к разработке, так как теперь Архитектор ПО и ML-инженер могут разработать ТЗ и архитектуру пилотного решения, а Data Scientist и Аналитик данных могут начать подготовку данных.

РАЗДЕЛ 3. ВЫБОР ИТ-РЕШЕНИЯ

3.1 ГРАНИЦЫ ПРОЕКТА И СТРАТЕГИЯ РАЗВИТИЯ

В данном разделе проводится анализ бизнес-требований к проекту, а также функциональных и нефункциональных требований к разрабатываемому пилотному решению с целью формулирования подхода к разработке с точки зрения ИТ-решений.

Границы проекта показывают, какая область конечного продукта будет реализована в текущем проекте (подробнее описаны в Приложении 1). Данный проект не предполагает полной реализации рекомендательной системы – его главная цель заключается в подтверждении существования более качественного с точки зрения метрики pFound подхода к реализации рекомендательной системы для маркетплейса в сфере моды, которое бы решало основные известные проблемы существующих решений, более подробно описанные в разделе 1.3 настоящего Отчета.

Стратегия работы над проектом и его последующего развития основывается на том факте, что данный проект ограничен реализацией основной логики, которая включает алгоритмы машинного обучения в рамках функционального блока RTB, а также RESTful API – функционального блока, отвечающего за доступ к RTB и обмену данными с ним. Блок интерфейса настройки системы в рамках проекта не реализуется, однако потенциал его появления должен быть предусмотрен. На данном этапе в том числе нет жесткого требования к быстродействию алгоритма, поэтому при выборе способа реализации проекта мы будем в большей степени учитывать другие критерии, в особенности, те, которые позволяют сократить издержки интеграции, разработки и масштабирования. При это для нас будет важным критерий гибкой настройки разрабатываемого пилотного решения, что тоже отразится на выборе решения. Тем не менее, в будущем возможна переоценка значимости критериев, поэтому важно продумать возможность перехода с выбранного решения на другие с наименьшими потерями.

Таким образом, выбор ИТ-решения сводится к выбору инструментов, с помощью которых будет реализовываться проект, а также архитектуре алгоритмов машинного обучения в рамках функционального блока RTB.

3.2 ОПРЕДЕЛЕНИЕ СОСТАВА КРИТЕРИЕВ ДЛЯ ВЫБОРА ИТ-РЕШЕНИЯ

Анализ функциональных и нефункциональных требований к системе, выдвигаемых Заказчиком, помогает определить состав критериев, на основе которых должны приниматься решения о выборе программных средств при проектировании модели с алгоритмами машинного обучения. Нужно обратить

внимание как на отдельные требования, выдвинутые конкретными стейкхолдерами, так и на общие требования к системе, чтобы сформировать критерии для выбора.

Таблица 3. Список показателей и выдвигаемых критериев выбора для ИТ-решения

Показатель	**Критерий (допустимые значения)**
Быстродействие алгоритма	Выбранное ИТ-решение (включая его архитектуру) должно позволять разработанному алгоритму выполнять один запрос не более чем за 10 миллисекунд, при этом ожидание очереди для каждого запроса не должно превышать 1 секунды (что при указанной в требованиях нагрузке означает о необходимости параллельного выполнения запросов) – оценка по скорости выполнения операций в зависимости от сложности алгоритма (выраженной через О-большое).
Использование отдельного сервера для модели на продуктовом контуре	Поддержка и удобство подключения к алгоритму через запросы REST архитектуры
Использование графического процессора NVIDIA для обучения	Поддержка ОС Cuda для вычислений
Использование различных подходов для решения задачи	Наличие и поддержка большого количества разнообразных модулей и библиотек для машинного и глубинного обучений (решение задач: классификации, кластеризации, регрессии и ранжирования)
Устранение проблемы различий в жизненном цикле рекомендуемых продуктов	Решение должно регулярно обновлять базу обучения и рекомендаций для обеспечения актуальности результатов выполнения рекомендаций
Устранение проблемы «холодного старта», когда модель не успевает получить достаточно новой информации о пользователе для составления рекомендаций	Решение должно строиться на общем принципе схожести товаров, обеспечивая релевантные рекомендации для любого пользователя, полностью основываясь на его текущих действиях (просмотре конкретного товара).
Устранение проблемы высокой значимости признака бренда для формирования рекомендации	Решение должно основываться на принципе схожести товаров при этом игнорируя изображение логотипа бренда на товаре, чтобы не обучаться на него.
Обмен данными	Решение должно позволять выдавать рекомендации по полученном от фронта запросу в режиме онлайн
Гибкость системы	Решение должно позволять потенциально расширить архитектуру добавлением UI[18] для гибкой настройки RTB
Доступный с различных устройств UI	Решение должно позволять расширить архитектуру с добавлением вэб UI
Локализация UI	Решение должно позволять быстро локализовать UI для настройки RTB на все языки, которые используются на платформе Заказчика.

[18] - здесь: пользовательский интерфейс для администраторов платформы Заказчика

3.3 АНАЛИЗ СУЩЕСТВУЮЩИХ ИТ-РЕШЕНИЙ

В данном разделе мы проведем анализ существующих подходов и архитектур рекомендательных систем с целью получения более полной картины о лучших практиках и выявления наиболее подходящего для нас решения (набора решений).

Существует несколько вариантов реализации:

1. No-Code
2. Low-Code
3. Реализация путем написания кода

Далее рассмотрим различные ИТ-решения в соответствии с типом реализации. Начнём с No-Code. No-Code решения предполагают, что для выполнения проекта будут использоваться специальные программные средства, позволяющие с помощью графического интерфейса настроить и обучить рекомендательную систему прямо на сайте, использовав заранее определённые в системе подходы и алгоритмы машинного обучения. К таким решениям можно отнести, например, easyrec[19], MyMediaLite[20], LensKit[21], Waffles[22]. Все перечисленные решения имеют несколько общих недостатков:

- Низкая степень кастомизации
- Ограниченный выбор методов и алгоритмов машинного обучения
- Отсутствие гибкости в требованиях к данным
- Ограниченный выбор метрик
- Недостаточный уровень контроля процесса обучения и применения алгоритма.

Тем не менее, имеются также и некоторые преимущества, такие как:

- Упрощенная интеграция через REST-архитектуру,
- Наличие подробных инструкций по работе с системами,
- Поддержка пользователей (не для всех решений).

Если рассматривать решения типа Low-Code, к которым относятся программные продукты, фреймворки и библиотеки, позволяющие сократить объем кода, который требуется писать на низком уровне, используя максимальную абстракцию на самом высоком уровне. Это позволяет сконструировать рекомендательную систему даже не разбирающимся в программировании пользователям. К таким решениям относятся: Suggest[23],

[19] - https://github.com/alibaba/EasyRec
[20] - http://www.mymedialite.net/
[21] - https://lenskit.org/
[22] - https://github.com/mikegashler/waffles
[23] - http://glaros.dtc.umn.edu/gkhome/suggest/overview

Duine[24], Crab[25], Recommenderlab[26], Apache Mahout[27], likelike[28], OpenSlopeOne[29], Voogoo[30]. Данные решения связаны со следующими недостатками:

- Недостаточный уровень кастомизации
- Выбор алгоритмов и методов шире, чем для No-Code решений, но недостаточно
- Применимы не для всех типов задач и данных
- Нельзя комбинировать разные решения.

Преимущества такие же, как у No-Code подходов.

Последним типом подхода является использование языков программирования для разработки решения, полностью адаптированного под конкретную задачу – именно полная свобода в настройке, обработке данных, сложности системы, качестве предсказания, выборе метрик и является основным преимуществом данного подхода. Если говорить о недостатках, то нельзя не упомянуть, что они вытекают из преимущества:

- Для построения системы необходимы компетенции в машинном обучении, программировании, понимании алгоритмов и структур данных;
- Разработанная система потребует оптимизации и более детальной проработки архитектуры.

К последнему типу подходов можно отнести любую разработку алгоритмов машинного обучения на языках Python (преимущества – огромный набор готовых инструментов, удобство в адаптации алгоритмов, большое количество бейзлайнов, научных статей и примеров кода; недостатки – неоптимальная работа с памятью, медленные встроенные функции), C/C++/C# (преимущества - разработка на более низком уровне, оптимизация памяти, более оптимизированные внутренние алгоритмы, недостатки – маленькая база готовых алгоритмов и примеров, специализированных модулей и библиотек), R/Matlab (недостатки – неоптимальные алгоритмы, неоптимальная работа с памятью, слабая интеграция с внешними системами, преимущества – быстрее, чем Python, много специализированных модулей и библиотек).

[24] - https://girlincomputerscience.blogspot.com/2012/11/duine-open-source-recommender.html
[25] - http://muricoca.github.com/crab/
[26] - https://cran.r-project.org/web/packages/recommenderlab/index.html
[27] - http://mahout.apache.org/
[28] - https://github.com/takahi-i/likelike
[29] - http://code.google.com/p/openslopeone/
[30] - http://sourceforge.net/projects/vogoo/

3.4 ОБОСНОВАНИЕ ВЫБОРА ИТ-РЕШЕНИЯ

3.4.1 ОБОСНОВАНИЕ ВЫБОРА ПЛАТФОРМЫ РАЗРАБОТКИ

На основе представленного анализа существующих решений и выдвигаемых требований, а также потенциального развития разрабатываемой пилотной Системы, был сделан выбор в пользу использования для проекта решения, предполагающего написание кода для реализации. Данный выбор удовлетворяет потребностям и бизнес-целям Заказчика, так как позволяет разрабатывать наиболее удобную, оптимизированную, кроссплатформенную Систему, которая не требует изменения во внутренней архитектуре Заказчика (что позволяет устранить конфликты с одним из стейкхолдеров - директора ИТ-отдела, который высказывал опасения по данному поводу), а также позволяет выполнить бизнес-задачи в полной мере.

Выбор конкретного ИТ-решения сделан на основе критериев, описанных в разделе 3.1 данного отчета и исследования, сделанного в пункте 3.2. Ниже приведена таблица соответствия решений критериям выбора (таблица 5).

Пояснения к оценкам: все оценки выставлялись по 5-балльной шкале (от 1 до 5) в целых числах с округлением по арифметическим правилам. Итог был подсчитан как средневзвешенное оценок по критериям с весами, представленными в таблице 4.

Таблица 4. Веса значимости оценки по критерию

К1	К2	К3	К4	К5	К6	К7	К8	К9	К10	К11
15%	9%	10%	15%	9%	8%	8%	6%	9%	2%	9%

Указанные веса были выбраны, исходя из оценки целей текущего проекта – в рамках разработки пилотного решения, значимость быстродействия не столь высокая, как когда речь идет о решении, внедряемом в промышленную эксплуатацию. В то же время данный проект предполагает создание гибкого решения, которое может быть легко перестроено и настроено для проверки различных продуктовых гипотез. Важным критерием также является возможность последующей интеграции с вэб-интерфейсом для настройки и управления системой, так как это является одной из целей развития системы согласно стратегии проекта. При этом критерии к самому интерфейсу на данный момент имеют небольшой вес, так как реализация интерфейса выходит за рамки границ проекта. Большое внимание также уделяется критериям, требующим от платформы для разработки наличие реализованных проектов в сфере машинного обучения, так как это может быть важно для сравнения результатов получившегося решения с существующими.

Таблица 5. Таблица соответствия решений критериям выбора

Критерий[31]	**Python**	**C**	**C++/C#**	**R/Matlab**
Быстродействие алгоритма[32]	1	5	5	3
Использование отдельного сервера для модели на продуктовом контуре[33]	5	2	3	1
Использование графического процессора NVIDIA для обучения[34]	5	5	5	4
Использование различных подходов для решения задачи[35]	5	2	3	4
Устранение проблемы различий в жизненном цикле рекомендуемых продуктов[36]	4	2	3	3
Устранение проблемы «холодного старта», когда модель не успевает получить достаточно новой информации о пользователе для составления рекомендаций[37]	4	3	3	4
Устранение проблемы высокой значимости признака бренда для формирования рекомендации[38]	5	1	2	3
Обмен данными[39]	3	4	4	2
Гибкость системы[40]	5	2	3	3
Доступный с различных устройств UI[41]	5	2	3	1
Локализация UI[42]	5	1	5	1
ИТОГ	**4,11**	**2,78**	**3,66**	**2,87**

Таким образом, было принято решение работать с Python.

[31] - здесь: соответствие пороговым значениям и критериям согласно таблице 3

[32] - оценки выставлялись согласно результатам испытаний: https://medium.com/swlh/a-performance-comparison-between-c-java-and-python-df3890545f6d и https://towardsdatascience.com/r-vs-python-vs-julia-90456a2bcbab#:~:text=R%20is%20not%20the%20fastest,in%20Julia%20is%20~3x)%3B

[33] - оценки выставлялись согласно ответам https://www.quora.com/What-is-the-best-language-framework-for-developing-a-REST-API-Im-currently-using-PHP-with-MongoDB-I-tried-CodeIgniter-but-I-found-it-poor-when-it-comes-to-creating-a-REST-API-so-now-Im-not-using-any-framework#:~:text=Originally%20Answered%3A%20Which%20language%20is,beautiful%20framework%20for%20REST%20APIs.

[34] - оценки выставлены согласно статье https://developer.nvidia.com/tools-overview

[35] - оценки выставлены согласно статье https://www.techtarget.com/searchapparchitecture/tip/Frameworks-libraries-and-languages-for-machine-learning

[36] - оценки по мнению сообщества относительно простоты работы с БД - https://www.reddit.com/r/learnprogramming/comments/1lo5d9/best_language_to_interface_with_odbc/

[37] - оценки выставлены на основе выбора лучшего языка для реализации градиентного бустинга - https://www.quora.com/What-is-the-best-language-for-implementing-gradient-boosted-decision-tree

[38] - оценки выставлены на основе статьи (аугментация реализована наиболее полно для Python) https://journalofbigdata.springeropen.com/articles/10.1186/s40537-019-0197-0

[39] - оценки выставлены как средневзвешенное между быстродействием (50%), удобством использования RESTfull API (30%) и ODBC (20%)

[40] - оценки выставлены как среднеарифметическое 37 и 43

[41] - оценки выставлены согласно статистике: https://www.monocubed.com/blog/most-popular-web-frameworks/

[42] - оценки основаны на функционале существующих вэб-фреймворков

3.4.2 ВЫБОР АЛГОРИТМОВ ДЛЯ RTB

Следующим шагом является выбор архитектуры функционального блока RTB, так как конкретный подход к API (при условии использования архитектуры REST) не влияет на метрики проекта и может быть выбран любым в зависимости от предпочтения команды разработки на выбранном языке программирования.

Как было описано в разделе 1.4 – наш функциональный блок, отвечающий за выполнение всех вычислений и формирование списка рекомендаций реализуется через 4 концептуальных блока (генерация признаков, генерация классов, классификация входных объектов, ранжирование результатов). Для каждого блока также необходимо выбрать наиболее подходящее решение – мы делали это, основываясь на экспертном мнении (научные работы) и сравнении результатов на имеющихся данных.

Общее применение в современном машинном обучении нашло использование обученных для классификации изображений нейронных сетей в качестве самообучаемых алгоритмов извлечения признаков из изображений. Благодаря применению сверточных нейронных сетей, различные архитектуры справляются с выделением определенных паттернов и кодированием их – именно эту особенность мы будем использовать для решения нашей задачи. Нейронная сеть представляет из себя набор матриц с параметрами для данных, соответствующих определенному размеру, поэтому для неё существует возможность «переноса обучения», когда параметры подбираются на выборке с размеченными данными, а затем из модели выбираются все слои, кроме последнего (он генерирует ответ) и применяются на другом наборе данных. Важно подобрать модель, которая обучалась на изображениях, похожих по специфике на те, с которыми мы будем работать, или дообучить ее дополнительно на размеченных данных. Наиболее известные архитектуры, которые применяют при «переносе обучения» [12]:

- Inception[43]
- VGG-16[44]
- ResNet-50[45]
- MobileNet[46]

Следующий блок – генерация признаков – решает задачу кластеризации, для чего мы будем использовать одну из передовых разработок в данном направлении – модель SPICE, которая основана на нейронных сетях и решает

[43] - https://arxiv.org/pdf/1409.4842.pdf
[44] - https://arxiv.org/pdf/1409.1556.pdf
[45] - https://arxiv.org/pdf/1512.03385.pdf
[46] - https://arxiv.org/pdf/1704.04861.pdf

задачу кластеризации. Данная модель описана в статье, опубликованной 14 января 2022 года [13].

Блок классификации входных данных включает в себя несколько этапов: 1 – обучение классификатора (здесь мы будем иметь на вход табличные данные и, согласно многим исследованиям [14] на таких данных лучше себя показывают методы классического машинного обучения, а именно – градиентный бустинг) и 2 – применение классификатора (на данном этапе нужно провести предварительную обработку входных данных, а затем передать их обученному на первом этапе алгоритму для получения ответа).

Блок ранжирования будет применяться без использования алгоритмов машинного обучения – ранг будет рассчитываться по следующей формуле:

$$rank_i = rank\left(order\left(pRel_i * price_i * P(brand\ \epsilon\ UserBrands)\right)\right)[i],$$

где $\mathrm{pRel_i}$ – релевантность рекомендации, $\mathrm{p}price_\mathrm{i}$ – цена позиции, $brand$ – брэнд позиции, $UserBrands$ – брэнды, которые нравятся пользователю.

Таким образом, в качестве ИТ-решения был выбран стэк из языка программирования Python и различных модулей, и алгоритмических решений, реализованных для него, а именно:

1. Сверточные нейронные сети для выделения признаков из изображений:
    a. Inception
    b. VGG-16
    c. ResNet-50
    d. MobileNet
2. Модель SPICE – самый передовой алгоритм для кластеризации изображений на данный момент – в качестве алгоритма генерации классов
3. Алгоритмы градиентного бустинга для классификации объектов каталога, как лучшие алгоритмы по анализу табличных данных.

3.5 ВЫВОДЫ ПО РАЗДЕЛУ

В данном разделе нашей командой проделана работа по изучению различных вариантов реализации проекта и выбору ИТ-решения. Мы рассмотрели как разные платформенные решения, так и разные алгоритмы и модели, которые можем применять к решению задачи. В результате данной проделанной работы был определен инструментарий, который будет использовать для разработки пилотного решения в дальнейшем.

РАЗДЕЛ 4. ПРАКТИЧЕСКАЯ РЕАЛИЗАЦИЯ

4.1 РЕАЛИЗАЦИЯ АЛГОРИТМОВ РЕКОМЕНДАЦИЙ RTB

4.1.1 РЕАЛИЗАЦИЯ МОДУЛЯ «ГЕНЕРАЦИЯ ПРИЗНАКОВ»

Как было отмечено в прошлых разделах, RTB делится на 4 концептуальных блока, которые последовательно выполняют задачи обработки данных для выполнения предсказаний. Первый из таких модулей – модуль генерации признаков. Он предназначен для генерации признаков из изображений, которые бы позволили модели работать с ними.

Концептуально для решения задачи рассматривалось несколько вариантов подходов:

1. Реализация генерации признаков с помощью предобученной глубокой нейронной сети классификации изображений
2. Реализация генерации признаков с помощью вариационного автокодировщика, обученного на имеющихся изображениях.

В своей работе мы применили оба варианта и сравнили результаты. Для первого варианта мы подобрали модели, которые зарекомендовали себя как наиболее успешные в задаче классификации изображений, исходя из предположения, что хорошая способность классификации является следствием построения высокого качества признакового описания изображений, позволяющего построить разделяющую поверхность для разных изображений.

Конечно же, вышеупомянутое предположение может быть парировано тем, что модели классификации изображений обучаются на изображениях, которые совсем не похожи на те, с которыми работаем мы, из-за чего довольно маловероятно, что такая модель умеет хорошо разделять разные виды одежды между собой, тем более не умеет разделять между собой брюки, например. Именно поэтому мы применили дообучение, чтобы подкорректировать параметры моделей так, чтобы они могли лучше выделять признаки с фотографий одежды. Дообучение проводилось на датасете с фотографиями одежды.

В качестве метрики качества для выделения признаков мы использовали MSSSIM[47] – метрику, которая позволяет определить, насколько похожи два изображения между собой и простое евклидово расстояние -метрику, которая позволяет понять, насколько удалены между собой векторы в метрическом пространстве R^n. Формула для подсчета MSSSIM приведена ниже:

[47] - Multi-scale Structural Similarity Index – метрика, измеряющая структурные отличия в изображении. Впервые упоминается на конференции «The Thirty-Seventh Asilomar Conference on Signals, Systems & Computers» в 2003 году. https://ieeexplore.ieee.org/abstract/document/1292216

$$MSSSIM(A,B) = [l_M(A,B)]^{\alpha_M} * \prod_{j=1}^{M} [c_j(A,B)]^{\beta_j} [s_{j(A,B)}]^{\gamma_j},$$

$$l_M(x,y) = \frac{1}{M}\sum_{i=1}^{M} \frac{2\mu_{xi} * \mu_{yi} + C_i}{\mu_{xi}^2 + \mu_{yi}^2 + C_i},$$

$$c(x,y) = \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2},$$

$$s(x,y) = \frac{\sigma_{xy} + C_3}{\sigma_x\sigma_y + C_3},$$

α, β, γ – параметры из полуинтервала (0, 1];
M - число размерностей изображения;
μ_{ij}- математическое ожидание;
σ_{ij}- среднеквадратическое отклонение;

Подход к оценке заключался в следующем: случайным образом из выборки выбирались 10000 пар фотографий, для каждой пары рассчитывался MSSSIM и евклидово расстояние между признаковыми описаниями фотографий, сгенерированных моделью, затем каждой паре в соответствие ставился ранг по MSSSIM и по евклидовому расстоянию. В качестве итоговой метрики (AARD) измерялся средний модуль разницы между рангами, нормированный на наибольший возможный средний модуль разницы, равный 2500.

Результаты моделей для классификации изображений можно увидеть в таблице 6.

Таблица 6. Таблица оценки моделей

Модель	**AARD, 10 epochs**	**AARD, 50 epochs**	**AARD, 100 epochs**	**AARD, 150 epochs**
ResNET-50	0,6154	0,5210	0,4257	0,3895
VGG-16	0,5752	0,5124	0,4025	0,3799
Inception	0,5802	0,4996	0,4122	0,3801
MobileNet	0,5992	0,5195	0,4265	0,3854
ResNET-50 with augment[48]	0,5528	0,4821	0,3724	0,3365
VAE[49]	0,4102	0,2501	0,1934	0,1542

Реализация с применением вариационного автокодировщика подразумевает использование скрытых состояний в качестве признакового

[48] Аугментация данных (намеренное случайное искажение изображений)
[49] VAE – вариационный автокодировщик

описания изображения, в предположении, что если из него получается восстановить исходное изображение, то оно хорошо описывает его.

Принцип работы вариационного автокодировщика описывается следующим алгоритмом: для каждого изображения путем сверток и применения линейных слоев генерируется вектор, называемый скрытым состоянием, а затем из данного вектора восстанавливается исходное изображением с применением функций обратной свертки (то есть расширяя изображение). Подробнее архитектура данной нейронной сети изображена на рисунке 9.

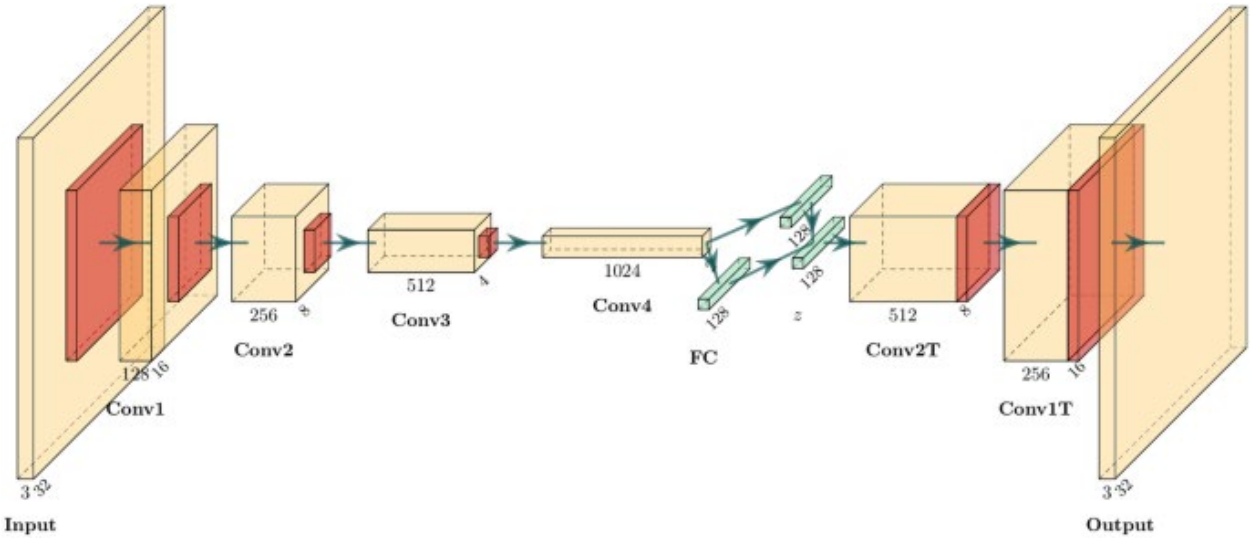

Рисунок 9. Архитектура нейронной сети VAE, Input - входное изображение, Conv - свертка, FC - полносвязный слой, Output - восс тановленное изображение

Результаты проведенных экспериментов, представленные в таблице 6, демонстрируют, что лучшим решением для генерации признаков является подход, основанный на вариационном автокодировщике. Таким образом, в нашем пилотном решении будет использоваться обученный на датасете фотографий с сайта маркетплейса FarFetch автокодировщик, а именно ту его часть, которая используется для получения скрытого пространства, то есть Encoder[50], а само скрытое пространство, результат применения Encoder сети, используем в качестве признакового описания изображения для генерации классов внутри одной категории, а также для последующей классификации новых товаров.

Данное улучшение у нас появилось уже после того, как было сформировано первичное понимание о том, как мы будем реализовывать модель рекомендаций RTB и представили демонстрационную модель (с использованием глубоких сверточных сетей для классификации изображений). Тем не менее, новый подход позволяет выиграть в качестве

[50] - сверточная нейронная сеть для извлечения признаков из изображений

извлечения признаков, что должно приводить к повышению качества кластеризации, а значит лучше использовать его.

4.1.2 РЕАЛИЗАЦИЯ МОДУЛЯ «ГЕНЕРАЦИЯ КЛАССОВ»

Модуль генерации классов является одним из основных в разрабатываемой модели, так как именно от его качества зависит релевантность рекомендаций, поэтому к поиску наилучшего решения для него необходимо подходить особо тщательно. Прежде всего, нужно понять, какие методы можно применить и почему. В разделе 3 данного отчета мы уже обсуждали данный вопрос и пришли к следующим вариантам решения:

1. Модель SPICE (целевое решение)
2. Модель K-Means
3. Модель DBSCAN
4. Модель Mean-Shift

В данном разделе мы еще глубже рассмотрим каждый из вариантов, проведем эксперименты, и примем решение о использовании одного из них.

Начнем с изучения модели SPICE. Модель SPICE решает задачу разнесения M изображений по k классам, и концептуально ее можно разделить на две части: первая – модель признаков, вторая – модель кластеризации, которая присваивает векторам вероятности принадлежности всем k классам. В отличие от существующих подходов к кластеризации данная модель использует признаковые векторы для выявления схожести изображений и модель кластеризации с псевдо-метками для поиска различий, как показано на рисунке 10.

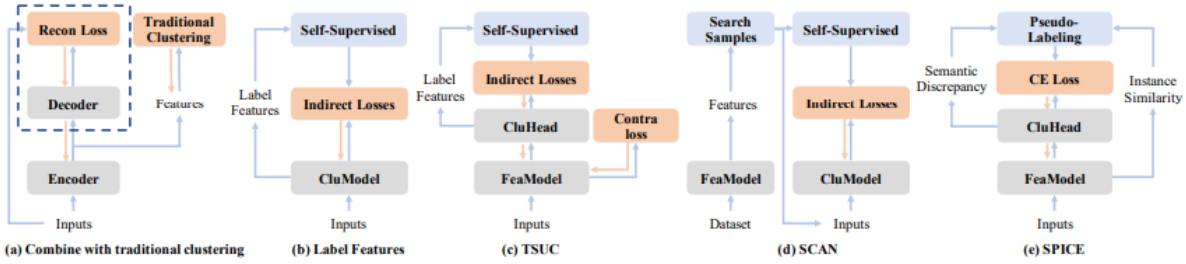

Рисунок 10. Сравнение существующих подходов к кластеризации изображений. (a) Стандартные методы кластеризации в комбинации с вариационным автокодировщиком (или только кодировщиком). (b) Модели кластеризации, основанные на признаковых описаниях кластеров. (c) Двуступенчатые модели: отдельная модель для признаков и отдельная для классификации по кластерам, сгенерированным по прочим признакам. (d) Модель ограничивающая различия в эмбедингах одного кластера. (e) SPICE модель

Подход, используемый в SPICE модели, представленный на рисунке 11:

1. Обучение признаковой модели для изображений с псевдо-классами.
2. Обучение модели кластеризации с фиксированной признаковой моделью и использованием псевдо-классов.

3. Совместное обучение обеих моделей с использованием достоверных псевдо-классов.

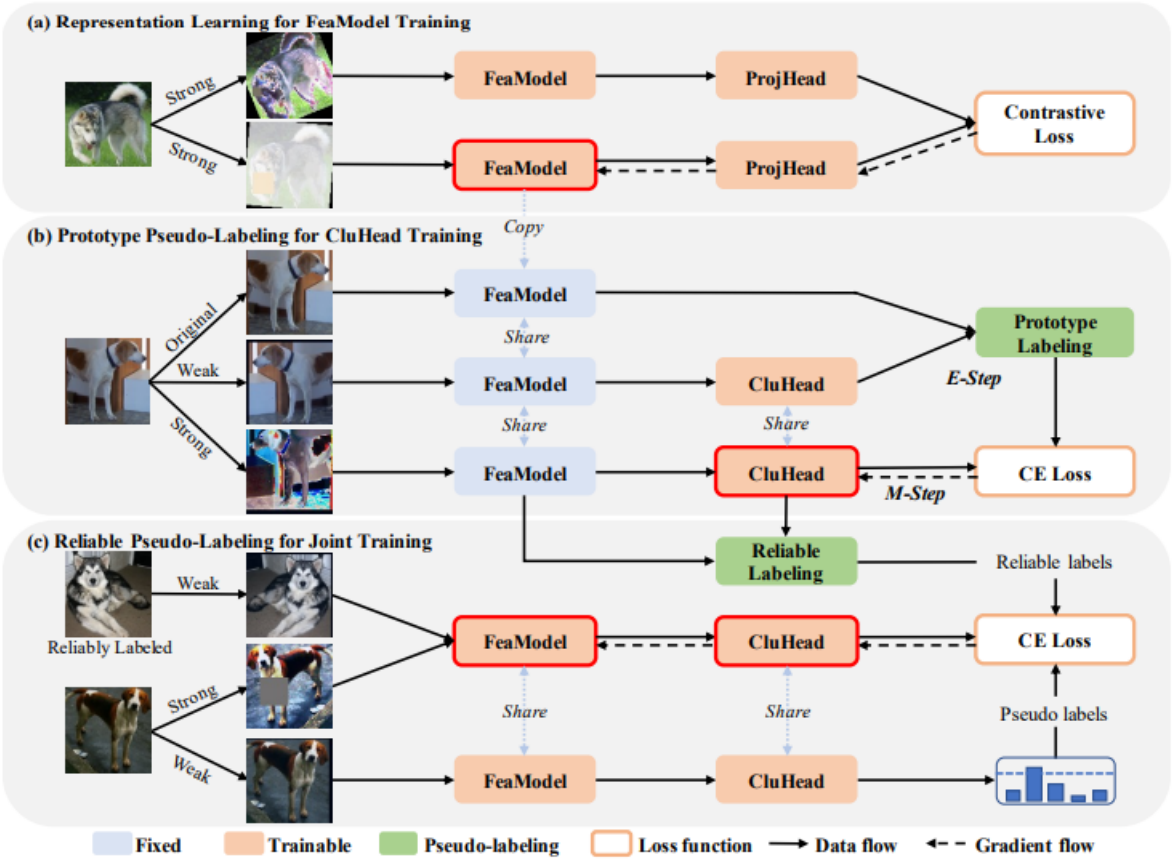

Рисунок 11. Иллюстрация алгоритма обучения SPICE модели. (a) Обучение без учителя модели признаков с контрастной функцией потерь. (b) Обучение модели кластеризации с использованием псевдо-классов и максимизации ожиданий. (c) Совместное обучение обеих моделей с использованием алгоритма «надежных» псевдо-классов.

Описанный алгоритм позволяет добавить модели понимание не только визуальной схожести изображений, но и семантических различий в них, что приводит к повышению качества кластеризации, измеренного на основе датасетов для классификации (задача для модели: кластеризовать изображения без учителя как можно ближе к реальным прометкам). Сравнение SPICE модели с другими алгоритмами кластеризации можно увидеть на рисунке 12.

Method	STL10			ImageNet-10			ImageNet-Dog-15			CIFAR-10			CIFAR-100-20			Tiny-ImageNet-200		
	ACC	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI
k-means [56]	0.192	0.125	0.061	0.241	0.119	0.057	0.105	0.055	0.020	0.229	0.087	0.049	0.130	0.084	0.028	0.025	0.065	0.005
SC [60]	0.159	0.098	0.048	0.274	0.151	0.076	0.111	0.038	0.013	0.247	0.103	0.085	0.136	0.090	0.022	0.022	0.063	0.004
AC [61]	0.332	0.239	0.140	0.242	0.138	0.067	0.139	0.037	0.021	0.228	0.105	0.065	0.138	0.098	0.034	0.027	0.069	0.005
NMF [62]	0.180	0.096	0.046	0.230	0.132	0.065	0.118	0.044	0.016	0.190	0.081	0.034	0.118	0.079	0.026	0.029	0.072	0.005
AE [63]	0.303	0.250	0.161	0.317	0.210	0.152	0.185	0.104	0.073	0.314	0.239	0.169	0.165	0.100	0.048	0.041	0.131	0.007
SDAE [17]	0.302	0.224	0.152	0.304	0.206	0.138	0.190	0.104	0.078	0.297	0.251	0.163	0.151	0.111	0.046	0.039	0.127	0.007
DCGAN [64]	0.298	0.210	0.139	0.346	0.225	0.157	0.174	0.121	0.078	0.315	0.265	0.176	0.151	0.120	0.045	0.041	0.135	0.007
DeCNN [65]	0.299	0.227	0.162	0.313	0.186	0.142	0.175	0.098	0.073	0.282	0.240	0.174	0.133	0.092	0.038	0.035	0.111	0.006
VAE [66]	0.282	0.200	0.146	0.334	0.193	0.168	0.179	0.107	0.079	0.291	0.245	0.167	0.152	0.108	0.040	0.036	0.113	0.006
JULE [21]	0.277	0.182	0.164	0.300	0.175	0.138	0.138	0.054	0.028	0.272	0.192	0.138	0.137	0.103	0.033	0.033	0.102	0.006
DEC [2]	0.359	0.276	0.186	0.381	0.282	0.203	0.195	0.122	0.079	0.301	0.257	0.161	0.185	0.136	0.050	0.037	0.115	0.007
DAC [11]	0.470	0.366	0.257	0.527	0.394	0.302	0.275	0.219	0.111	0.522	0.396	0.306	0.238	0.185	0.088	0.066	0.190	0.017
DeepCluster [45]	0.334	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	0.374	N/A	N/A	0.189	N/A	N/A	N/A	N/A	N/A
DDC [67]	0.489	0.371	0.267	0.577	0.433	0.345	N/A	N/A	N/A	0.524	0.424	0.329	N/A	N/A	N/A	N/A	N/A	N/A
IIC [12]	0.610	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	0.617	N/A	N/A	0.257	N/A	N/A	N/A	N/A	N/A
DCCM [13]	0.482	0.376	0.262	0.710	0.608	0.555	0.383	0.321	0.182	0.623	0.496	0.408	0.327	0.285	0.173	0.108	0.224	0.038
DSEC [24]	0.482	0.403	0.286	0.674	0.583	0.522	0.264	0.236	0.124	0.478	0.438	0.340	0.255	0.212	0.110	0.066	0.190	0.017
GATCluster [14]	0.583	0.446	0.363	0.762	0.609	0.572	0.333	0.322	0.200	0.610	0.475	0.402	0.281	0.215	0.116	N/A	N/A	N/A
PICA [25]	0.713	0.611	0.531	0.870	0.802	0.761	0.352	0.352	0.201	0.696	0.591	0.512	0.337	0.310	0.171	0.098	0.277	0.040
CC [15]	0.850	0.746	0.726	0.893	0.859	0.822	0.429	0.445	0.274	0.790	0.705	0.637	0.429	0.431	0.266	0.140	0.340	0.071
IDFD [16]	0.756	0.643	0.575	0.954	0.898	0.901	0.591	0.546	0.413	0.815	0.711	0.663	0.425	0.426	0.264	N/A	N/A	N/A
SPICE$_s$	0.908	0.817	0.812	0.921	0.828	0.836	0.646	0.572	0.479	0.838	0.734	0.705	0.468	0.448	0.294	**0.305**	**0.449**	**0.161**
SPICE	**0.938**	**0.872**	**0.870**	**0.959**	**0.902**	**0.912**	**0.675**	**0.627**	**0.526**	**0.926**	**0.865**	**0.852**	**0.538**	**0.567**	**0.387**	N/A	N/A	N/A

*Рисунок 12. Сравнение различных моделей кластеризации в задаче кластеризации размеченных данных без использования разметок. Качество измерялось с помощью метрик Accuracy (ACC), Normalized Mutual Information (NMI), Adjusted Rand Index (ARI). Лучшие показатели по метрикам выделены **жирным***

Изучение подхода SPICE дало понимание, что мы можем использовать тот же подход, доработав его под нашу задачу.

В качестве признаковой модели мы будем использовать уже обученный нами модуль «Генерация признаков», в качестве модели кластеризации – модель классификации (полносвязная нейронная сеть), используя в качестве кросс-энтропию по псевдо-классам, сгенерированным на основе текстового описания товаров. После чего проведем совместное обучение: генератор из первого модуля и модель кластеризации будем обучать, повышая кросс-энтропию.

Описанный выше подход мы сравним со следующим: кластеризация с использованием модели DBSCAN, которая с помощью случайных инициализаций находит наиболее плотные точки в пространстве, объединяя их в один кластер. Данная модель часто позволяет с большой точностью разделять объекты, если они отличаются по плотности распределения в пространстве. Предположение о плотности распределения кажется разумным, ведь если признаковые векторы хорошо описывают объекты, то разница между ними должна выражаться в отдаленности друг от друга.

В качестве бейзлайна так же будем использовать k-means кластеризатор, который мы применяли в демонстрационной модели, и который позволил получить достаточный уровень качества, измеренный целевой метрикой.

Данный алгоритм ищет k центров в пространстве, вокруг которых строит кластеры.

Еще одной интересной моделью, которую интересно было бы проверить, является Mean-Shift. Mean-Shift кластеризация – это алгоритм на основе скользящего окна, который пытается найти плотные области в пространстве. В качестве скользящего окна используется открытый шар в метрическом пространстве, центр которого движется в сторону увеличения количества точек внутри шара. Все найденные точки-кандидаты центров затем фильтруются для исключения дубликатов, после чего формируется окончательный набор центральных точек, вокруг которых строятся кластеры. В отличие от k-means кластеризации данный алгоритм не имеет гиперпараметра k – количество кластеров, так как он сам его определяет, что является значительным преимуществом, однако задача подбора радиуса шара остается решаемой только с помощью эвристик.

Выбирать между моделями будем, используя специально размеченный набор изображений одежды, являющийся на данный момент эталонным для задач компьютерного зрения, связанных с модой и одеждой[51]. Сравнение подходов проводится по метрике accuracy множества классов (результаты в таблице 7):

$$Accuracy = \frac{1}{N}\sum_{i} I[a(x_i) = y_i],$$

N - число объектов,
I[z=y] - индикатор,
a(x) - ответ модели на объекте,
y - верный класс объекта.

Таблица 7. Accuracy кластеризации различными моделями

Metric	**SPICE custom**	**K-Means**	**DBSCAN**	**Mean-Shift**
Accuracy	0.7154	0.6785	0.4251	0.6259
Время обучения, сек.	324	24	12	125
Время применения, сек.	15	12	10	22

По совокупности показателей разных моделей мы приняли решение использовать в нашей работе модель K-Means, так как она не сильно уступает

[51] - https://www.kaggle.com/code/igorlukhnev/stage-3-feature-engineering/data

в качестве SPICE, но при этом гораздо быстрее обучается и выигрывает на этапе применения.

4.1.3 РЕАЛИЗАЦИЯ МОДУЛЯ «КЛАССИФИКАЦИЯ ОБЪЕКТОВ»

Модуль классификации объектов работает с данными, которые были обработаны предыдущими модулями системы, поэтому его реализация должна зависеть от того, как меняются данные на предыдущих этапах, и в то же время с максимальным качеством обучаться на них. Очевидно, что модели, используемые в первых модулях RTB, возвращают табличные данные, на основе которых необходимо построить модель классификации.

На данный момент множество исследований [14] и опросов [16] подтверждают, что для табличных данных в вопросах качества и скорости обучения нейронные сети все еще не могут показывать лучшие результаты в сравнение с алгоритмами градиентного бустинга (на рисунке 13 продемонстрированы результаты различных моделей для датасета с табличными характеристиками человека в задаче предсказания возраста).

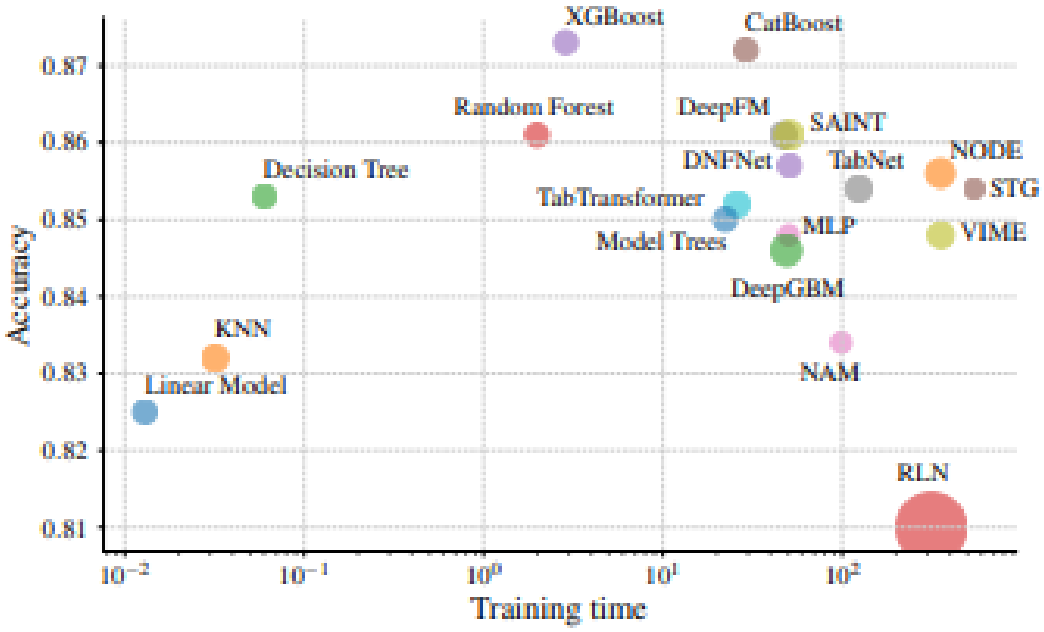

Рисунок 13. Показатели моделей в задаче регрессии по табличным данным [16]. По оси абсцисс отложено время обучения, по оси ординат - метрика качества accuracy. XGBoost и CatBoost - алгоритмы градиентного бустинга.

Для решения нашей задачи мы выбрали реализацию градиентного бустинга через решение, разработанное командой из компании Яндекс, известное как CatBoost. Обучение данной модели на данных, полученных в результате доработки, позволило улучшить результаты по сравнению с демонстрационной версией RTB. Результаты текущей итерации представлены также на рисунке 14 (измерены в метриках precision, recall, f1-score, accuracy).

Формула для метрики f1-score:

$$f1 = 2 * \frac{precision * recall}{precision + recall}$$

	precision	recall	f1-score	support
0.0	0.97	0.97	0.97	315
1.0	0.98	0.99	0.98	473
2.0	0.98	0.98	0.98	373
3.0	0.99	0.98	0.99	442
accuracy			0.98	1603
macro avg	0.98	0.98	0.98	1603
weighted avg	0.98	0.98	0.98	1603

Рисунок 14. Результаты обучения модели на данных, полученных из модулей 1 и 2

4.1.4 РЕАЛИЗАЦИЯ МОДУЛЯ РАНЖИРОВАНИЯ РЕКОМЕНДАЦИЙ

Задача ранжирования является одной из самых нетривиальных в рекомендательных моделях, обычно являющихся частью единой с кодировщиком модели, которая обучается на матрице отношений – матрице, показывающей степень заинтересованности пользователя в объекте (составляется по историческим данным). Такая система ранжирования используется на маркетплейсе сейчас, однако у нее есть недостатки, которые упоминались в разделах 2 и 3, в связи с чем мы приняли решение пересмотреть подход к ранжированию и рекомендациям в целом.

Наш подход к ранжированию будет строиться на оценке релевантности, полученной по следующему алгоритму:

1. На все k мест в ленте рекомендаций проводится аукцион по вероятности принадлежности просматриваемого пользователем объекта к классу.
2. Среди объектов выбранного класса выбирается объект, наиболее похожий на просматриваемый. Похожесть определяется по следующей формуле (чем меньше значение, тем более похожи объекты):

$$similarity = \alpha * (price_y - price_x) + \beta * D(x, y) + \gamma * U(brand_x),$$

где y – просматриваемый объект,

x - объект из выбранного класса,

$price_z$- цена объекта,

D(x, y) - расстояние между векторами,

U($brand_x$) - позиция бренда в топе пользователя,

α, β, γ - веса критериев

3. Позицию в ленте занимает продукт с наименьшим значением similarity из продуктов выбранного класса, которые еще не попали в рекомендации.

Параметры α, β, γ – являются обучаемыми и индивидуальными для каждого пользователя, метрика расстояния между векторами является стандартной для метрических пространств – Евклидовой.

Данный алгоритм позволил нам получить на тестовой выборке усредненное по парам пользователь-объект значение целевой метрики $pFound$ равное 78%.

4.2 РАЗРАБОТКА API

Все системы связаны между друг другом API. Если обратиться к классическому случаю, то Application Programming Interface – это любой код, предоставляющий какой-то интерфейс. В случае сетевой инфраструктуры API это сервер, к которому по определенному контракту обращается клиент. В свою очередь API выполняет все функции маршрутизации, обогащения данных и т. п., снимая с клиента необходимость делать это локально и позволяя инкапсулировать логику внутренних систем.

Определенный интерфейс и четкий контракт, т. е. прописанный регламент взаимодействия с системой позволяют подключаться к API любой системе, соблюдающей контракт. Также, подобное построение сетевой инфраструктуры позволяет производить изменения во внутренней логике, не нарушая обратную совместимость, т. е. изменения бека не приводят к необходимости изменений клиента.

Мы используем REST API. Representational State Transfer API это API, реализованное в архитектурном силе, имеющему следующие свойства:

1. Клиент-серверное устройство. Есть клиенты, которые по каким-то правилам обращаются к одному серверу.
2. Stateless-подход. Сервер не хранит данные о предыдущих обращениях пользователя, предыдущие результаты работы сервера для данного клиента не влияют на настоящий результат работы.

3. Единообразие интерфейса. Клиент получает как можно более полную информацию о возможностях действий с объектом.

4. Слоистая архитектура. То есть клиент и сервер не взаимодействую друг с другом напрямую. Они имеют посредников.

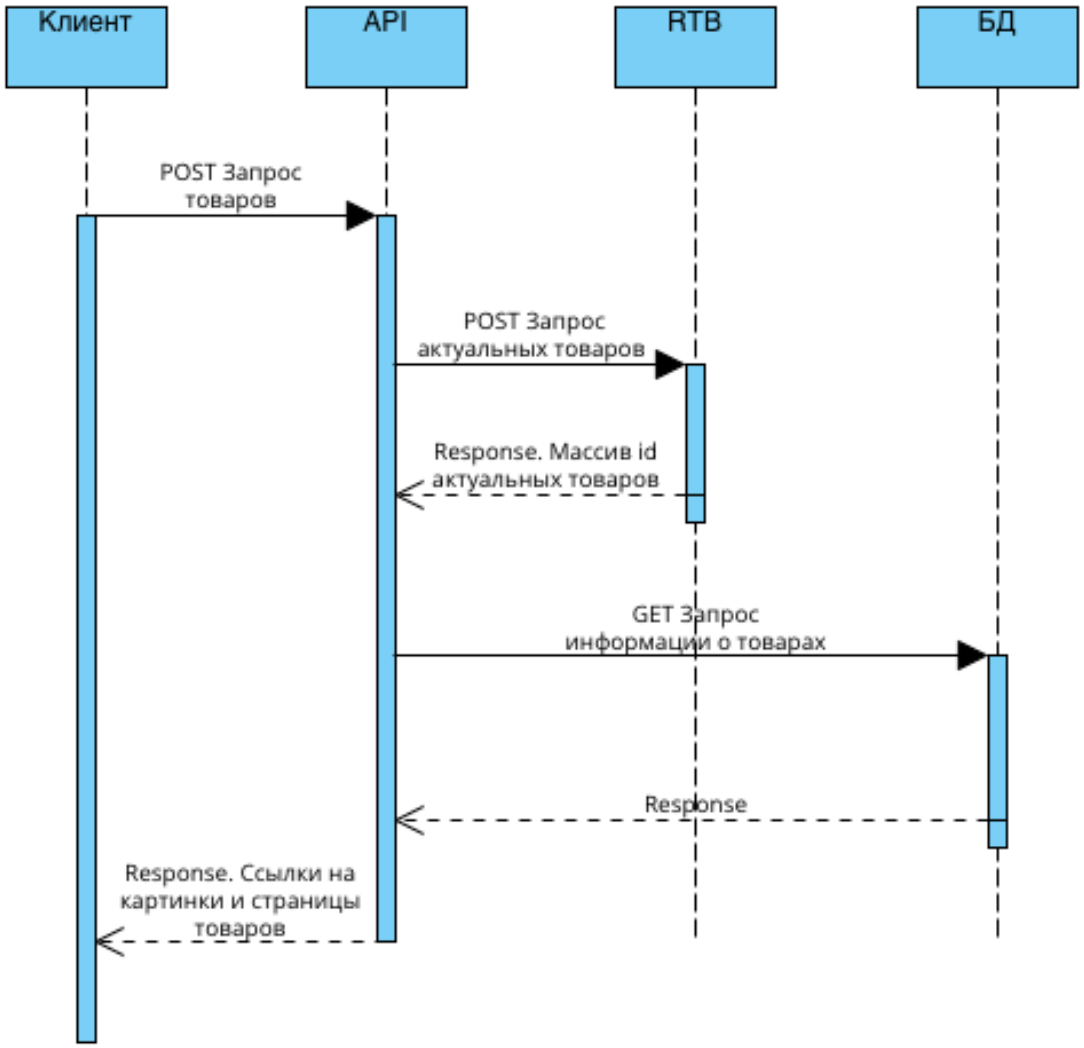

Рисунок 15. Sequence-диаграмма работы API

Клиентское приложение или веб-страница обращаются к API с помощью http запросов, передавая атрибуты товара, в query-параметрах и в body запроса цену, id и саму картинку. API маршрутизирует запрос в RTB, получая назад массив id подходящих товарах. API обращается к БД, получая ссылки на товары и ссылки на картинки. И отправляя их в ответ на запрос клиента. Все описанное представлено на рисунке 15.

API реализуется на языке Python с помощью пакетов Flask и flask_restful. Из них мы импортируем объекты Api и Flask. Далее создаем объект Quote, методы которого являются обработчиками запросов к Api. С помощью пакета requests мы реализуем запросы Api к другим сервисам.

4.3 ВЫВОДЫ ПО РАЗДЕЛУ

В данном разделе описаны подходы к реализации основных алгоритмов и компонентов разрабатываемой рекомендательной системы. В процессе разработки мы проверили различные гипотезы, выдвинутые в разделе 3 и изменили часть решений относительно модели машинного обучения для рекомендаций: улучшили процесс извлечения признаков из изображений (обучили и внедрили вариационный автокодировщик), получив векторы признаков меньшей размерности при более высоком качестве (оценка по метрике AARD – метод подсчета описан в разделе 4.1.1), повысили качество кластеризации за счет более тщательного подхода к оценке качества и подбора гиперпараметров, в результате получили намного более качественные результаты в классификации объектов, а новая модель ранжирования позволила значительно повысить значение целевой метрики. Кроме того, мы реализовали веб-архитектуру рекомендательной системы, основанную на REST-API и протоколе HTTP, в качестве формата передаваемых данных выбран JSON.

РАЗДЕЛ 5. СТРАТЕГИЯ ВНЕДРЕНИЯ

5.1 ИНФРАСТРУКТУРНЫЕ ИЗМЕНЕНИЯ

Внедрение предлагаемого решения не требует инфраструктурных изменений в ИТ-ландшафте Заказчика, так как не является высоко требовательным. Заказчику необходимо запустить сервер, отвечающий следующим техническим характеристикам:

- CPU с тактовой частотой не менее 3.0 ГГц, числом ядер не менее 32 (не менее 64 потоков).
- GPU NVIDIA TITAN RTX с объемом видео-памяти не менее 24 Гб типа GDDR6.
- Оперативным запоминающим устройством с объёмом памяти не менее 32 Гб по стандарту DDR4.
- Твердотельным постоянным запоминающим устройством с объёмом памяти не менее 512 Гб.

На данном сервере будет размещена компонента RTB, отвечающая за вычисления и предложение рекомендаций, сервер должен находиться в одной сети или одном кластере дата-центра с операционными базами данных и базами мастер-данных компании, чтобы иметь возможность посредством защищенных каналов через компоненту API получать доступ к необходимым для работы системы данным. Целевая архитектура решения представлена на рисунке 7, в разделе 2.1.

Отдельно так же предлагается развертывание OLAP-системы на одном из существующих серверов хранилища компании или на новом выделенном сервере, в которой будут строиться отчеты по логам системы, данный сервер так же необходимо обеспечить высокоскоростными запоминающими устройствами большого объема памяти, необходимых для хранения денормализованных витрин данных для отчетов, а также выполнения запросов, которые могут сильно нагружать оперативную память сервера.

Другие изменения в базах данных, серверах, сетях и шинах не требуются. Модель перехода представлена на рисунке 16.

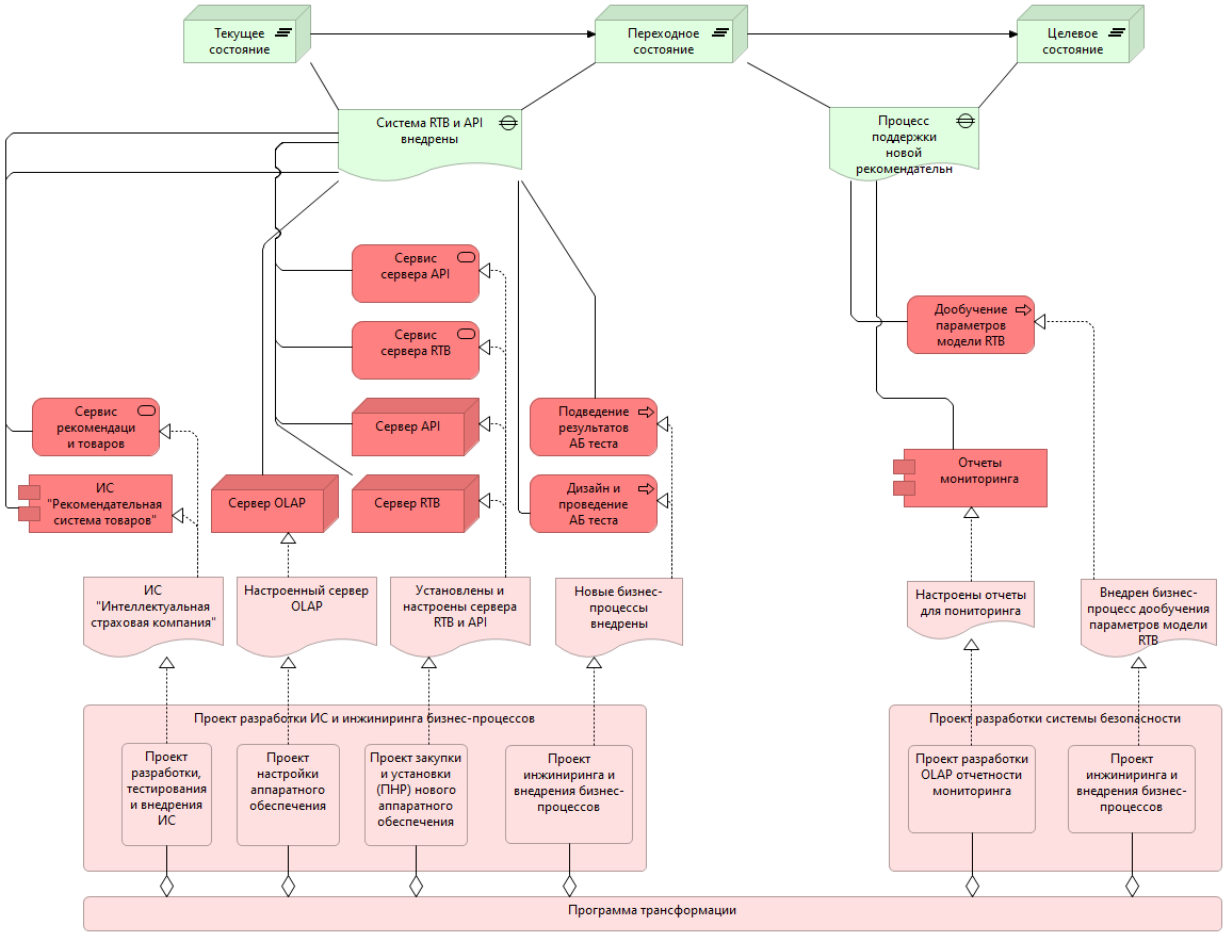

Рисунок 16. Модель перехода для маркетплейса FarFetch на новую систему рекомендации товаров

На представленной модели перехода видно, что трансформация будет проходить в два этапа: на первом этапе планируется разработать ПО, необходимое для работы системы, закупить, установить и настроить необходимое аппаратное обеспечение, развернуть на нем разработанное ПО и подготовить для будущей отчетности сервер OLAP, а также внедрить процессы АБ-тестирования, провести дизайн и запустить тест, после чего подвести его итоги и принять решение о внедрении; в случае успешного результата АБ-теста, планируется на втором этапе разработать отчетность для мониторинга состояния работы новой системы, а также разработать процесс дообучения параметров моделей по мере использования системы.

5.2 A|B ТЕСТ

Несмотря на то, что метрики качества наших алгоритмов демонстрируют, что пилотное решение позволяет делать более качественные рекомендации пользователям, мы не можем оценить, насколько лучше данное решение будет для бизнеса с точки зрения выручки. Однако мы можем проверить гипотезу о том, что внедрение новой системы рекомендаций позволит увеличить количество продаж и средний чек.

Для проверки данной гипотезы потребуется проведение A|B теста. В группе А будет использоваться рекомендательная система, которая используется сейчас, для группы В рекомендации будет выполнять пилотная система. Для нашего теста нулевой гипотезой будет отсутствие различий в группах, а альтернативной гипотезой будет рост среднего чека в группе В на 10%. Уровень значимости установим равным 5%, мощность критерия – 80%. Использовать будем t-критерий для нормального распределения. Оценку размера группы мы проведем, используя следующую формулу:

$$n = 2 * \frac{\sigma^2 * \left(T_{\frac{\alpha}{2}} + T_{\beta}\right)^2}{(\mu_B - \mu_A)^2}$$

Продолжительность теста определим в 1 месяц, исходя из ежемесячного потока пользователей на сайт (около 10 млн, такой размер достаточен для формирования статистически значимых выводов на основании теста), после чего посмотрим на результаты и примем решение о доработках/распространении пилотного решения.

Пример отчета для отслеживания результатов теста в таблице 8.

Таблица 8. Таблица с примером статистически значимого увеличения размера среднего чека в группе с пилотным решением.

Группа	База	Число покупок	Средний чек, $
А (старая система)	5 000 000	230 000	458
В (пилотное решение)	5 000 000	280 000	504

ЗАКЛЮЧЕНИЕ

В рамках данного проекта нами было разработано пилотное решение, состоящие из двух компонент: RTB – модель на основе алгоритмов машинного обучения, формирующая рекомендации пользователю и API – дополнительное ПО, обеспечивающее взаимодействие визуальных объектов клиента (веб-сайт или мобильное приложение) с RTB и базой данных каталога Farfetch. Выполнили поставленные задачи, удовлетворив требования, выдвинутые Заказчиком:

1. Рекомендации построены на основе визуальной схожести объектов, за счет чего удается уловить интересы пользователя с первого просмотра товаров (таким образом получается рекомендовать нишевые товары), а также рекомендовать аналоги просматриваемого товара,
2. Система подстраивается под каждого пользователя индивидуально, но при этом использует элемент случайности, что позволяет избежать наличие только товаров просматриваемого бренда в рекомендациях.

Удовлетворение описанных выше требований позволило достичь уровня целевой метрики выше установленного в ТЗ на 8 п. п. (78%).

Тем не менее, у проекта все еще остается много векторов для развития: для реализации проекта внедрения необходимо разработать отчеты для мониторинга основных метрик (распределения показов товаров в рекомендациях, изменения операционных показателей: количества продаж, среднего чека – в разрезе когорт пользователей, категорий товаров и наличия скидок). Кроме того, в случае успешного опыта внедрения, появится потребность в наличии пользовательского интерфейса для корректирования обучаемых параметров модели в ручном режиме при необходимости.

ИСПОЛЬЗОВАННЫЕ МАТЕРИАЛЫ

1. Dietmar Jannach, Michael Jugovac. Measuring the Business Value of Recommender Systems, ACM Transactions of Management Information Systems, 10(4), art.1, 2019 - Текст: электронный // Ресурс Arxiv – URL: https://arxiv.org/pdf/1908.08328.pdf
2. Lawrence R.D, et al. Personalization of Supermarket Product Recommendations, Data Mining and Knowledge Discovery, 5(1), 2001, pp.11-32 - Текст: электронный // Ресурс ResearchGate – URL: https://www.researchgate.net/publication/280483409_Personalization_of_Supermarket_Product_Recommendations
3. Malcolm B.D, et al. The value of personalised recommender systems to e-business, RecSys '08: Proceedings of the 2008 ACM conference on Recommender systems, 2008, pp.291-294 - Текст: электронный // Ресурс ResearchGate – URL: https://www.researchgate.net/publication/221141044_The_value_of_personalised_recommender_systems_to_e-business
4. Brovman Y.M, et al. Optimizing Similar Item Recommendations in a Semi-structured Marketplace to Maximize Conversion, RecSys '16: Proceedings of the 10th ACM Conference on Recommendation Systems, 2016, pp.199-202 - Текст: электронный // Ресурс ACM Digital Library – URL: https://dl.acm.org/doi/abs/10.1145/2959100.2959166
5. Chen Y., Canny J.F. Recommending ephemeral items at web scale, SIGIR `11: Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval, 2011, pp.1013-1022 - Текст: электронный // Ресурс ACM Digital Library – URL: https://dl.acm.org/doi/10.1145/2009916.2010051
6. Deldjoo Y., et al. A Review of Modern Fashion Recommender Systems. ACM Computer Surveys, 37(4), 111, 2021 – URL: https://arxiv.org/abs/2202.02757
7. Goncalves D., Brochado P. How to build a resommender system: it's all about rocket science – Part 1 – Текст: электронный // Ресурс FarFetchTech – URL: https://www.farfetchtechblog.com/en/blog/post/how-to-build-a-recommender-system-it-s-all-about-rocket-science-part-1/
8. Goncalves D., Brochado P. How to build a resommender system: it's all about rocket science – Part 2 – Текст: электронный // Ресурс FarFetchTech – URL: https://www.farfetchtechblog.com/en/blog/post/how-to-build-a-recommender-system-it-s-all-about-rocket-science-part-2/

9. Гулин А., Карпович П., Расковалов Д., Сегалович И. Яндекс на РОМИП'2009. Оптимизация алгоритмов ранжирования методами машинного обучения, Российский семинар по Оценке Методов Информационного Поиска. Труды РОМИП 2009 – НУ ЦСИ, Санкт-Петербург, 2009 – с. 163-168
10. Grandini M., Bagli E., Visani G. Metrics for Multi-Class Classification: an Overview. A White Paper – 2020 – Текст: электронный // Ресурс Arxiv – URL: https://arxiv.org/pdf/2008.05756.pdf
11. Бутаков Д., Иванов А. (2021). Канал Yandex.Cloud. Рекомендательные системы: архитектура и применение [обучающее видео, семинар] // YouTube 21 июня (https://www.youtube.com/watch?v=P9yXoBAmELQ) – Просмотрено: 20.02.2022
12. Fengzi L., Shashi K., Shunichi A., Sumer B., Swapna S. Neural Networks for Fashion Image Classification and Visual Search – 2020 – Текст: электронный // Ресурс Arxiv – URL: https://arxiv.org/pdf/2005.08170.pdf
13. Chuang N., Hongming S., Ge W. SPICE: Semantic Pseudo-Labeling for Image Clustering – 2022 – Текст: электронный // Ресурс Arxiv – URL: https://arxiv.org/pdf/2103.09382v3.pdf
14. Schwartz-Ziv R., Armon A. Tabular Data: Deep Learning Is Not All You Need – 2021 – Текст: электронный // Ресурс Arxiv – URL: https://arxiv.org/pdf/2106.03253.pdf
15. Z. Wang, E. P. Simoncelli and A. C. Bovik, "Multiscale structural similarity for image quality assessment," The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003 - с. 1398-1402 Vol.2
16. Borisov V., Leemann T., Sebler K., Haug J., Pawelczyk M., Kasneci G. Deep Neural Networks and Tabular Data: A Survey – 2022 – Текст: электронный // Ресурс Arxiv – URL: https://arxiv.org/pdf/2110.01889.pdf
17. Kohavi R., Longbotham R., Sommerfield D., Henne R. Controlled experiments on the web: survey and practical guide – 2021 – Текст: электронный // Ресурс Exp-Platform – URL: https://www.exp-platform.com/Documents/controlledExperimentDMKD.pdf
18. Fielding R. Architectural Styles and the Design of Network-based Software Architectures – 2000 – Текст: электронный // Ресурс University of California – URL: https://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm