МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ФАКУЛЬТЕТ ПРИКЛАДНОЙ МАТЕМАТИКИ И ИНФОРМАТИКИ
Кафедра компьютерных технологий и систем

ЛЕБЕДЕВИЧ Артем Владимирович

АЛГОРИТМИЧЕСКИЕ И ТЕХНИЧЕСКИЕ АСПЕКТЫ ИНТЕГРАЦИИ СРЕДСТВ КЛАСТЕРИЗАЦИИ, СОПОСТАВЛЕНИЕ ПРОГРАММНЫХ РЕАЛИЗАЦИЙ

Дипломная работа

Научный руководитель:
профессор, доктор
физико-математических наук
В.Б. Таранчук

Допущена к защите

«_____» _____ 2025 г.
Заведующий кафедрой КТС
профессор, доктор педагогических наук,
кандидат физико-математических наук
В.В. Казаченок

Минск, 2025

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
Факультет прикладной математики и информатики
Кафедра компьютерных технологий и систем

УТВЕРЖДАЮ
Заведующий кафедрой

_____
(подпись)(фамилия, инициалы)

14.11.2024 г.

ЗАДАНИЕ
на дипломную работу

Обучающемуся Лебедевичу Артему Владимировичу
Курс 4 Учебная группа 12
Специальность «Прикладная информатика»

Тема дипломной работы: «Алгоритмические и технические аспекты интеграции средств кластеризации, сопоставление программных реализаций».

Утверждена указом ректора БГУ от 12.11.2024 № 1295-ПС.

Исходные данные к дипломной работе:

1. Воронцов, К. В. Алгоритмы кластеризации и многомерного шкалирования. Курс лекций / К. В. Воронцов. – М. : МГУ, 2007. – 352 с.
2. Avazpour, I. Dimensions and Metrics for Evaluating Recommendation Systems / I. Avazpour, T. Pitakrat, J. Grunske. – Recommendation Systems in Software Engineering, 2014. – 29 с.
3. Burke, R. Hybrid Recommender Systems: Survey and Experiments / R. Burke, X. Chen, Y. Yu. – User Modeling and User-Adapted Interaction, 2002. – 331-370 с.
4. Ricci, F. Recommender Systems Handbook. / F. Ricci, L. Rokach, B. Shapira. – Springer, 2015. – 1003 с.

Перечень подлежащих разработке вопросов или краткое содержание расчетно-пояснительной записки:
1) систематизация теоретических основ рекомендатльных систем, уделяя внимание вариационным выводам и метрикам устойчивого развития;
2) формулировка задачи активного обучения и кластеризации;
3) разработка гибридного алгоритма Basket-Hybrid

4) описание практического внедрения и архитектуры DAG.

Примерный календарный график выполнения дипломной работы:

– **февраль (1-ая неделя)** – ознакомление с условиями работы, изучение основных теоретических вопросов; получение задания; изучение алгоритма рекомендательных систем;
– **февраль (2-3-я неделя)** – изучение и имеюшихся систем рекомендаций, поиски мест для улучшения и увеличение целевых метрик;
– **март (4-5-ая неделя)** – изучение и применение методов поиска резко выделяющихся значений;
– **март (6-7-ая неделя)** – анализ алгоритмов кластеризации;
– **апрель (8-9-ая неделя)** – разработка улучшенного рекомендательного алгоритма;
– **апрель (10-11-ая неделя)** – описание, оформление теоретической части работы; подготовка презентации;
– **май (12-15-ая неделя)** – подготовка презентации, прохождение системы «Антиплагиат», получение допуска, рецензирование дипломной работы.

Дата выдачи задания 14.11.2024.
Срок сдачи законченной дипломной работы 27.05.2025.

Руководитель дипломной работы _____ В. Б. Таранчук
(подпись)

Подпись обучающегося _____
(подпись)

Дата 14.11.2024 г.

Проинформирован о недопустимости привлечения третьих лиц к выполнению дипломной работы, плагиата, фальсификации или подлога материалов.
_____ А. В. Лебедевич
(подпись)

ОГЛАВЛЕНИЕ

РЕФЕРАТ 4

ВВЕДЕНИЕ 7

ГЛАВА 1 ТЕОРЕТИЧЕСКИЕ ОСНОВЫ РАБОТЫ РЕКОМЕНДАТЕЛЬНЫХ СИСТЕМ 10

1.1 Основные типы рекомендательных систем 10

1.2 Активное обучение (Active Learning) и его применение 13

1.3 Методы кластеризации для задач рекомендаций 15

1.4 Методы оценки качества рекомендаций 17

1.5 Современные подходы к оптимизации рекомендательных систем . . 19

ГЛАВА 2 РАБОТА НАД УЛУЧШЕНИЕМ МЕХАНИЗМОВ МАТЧИНГА ТОВАРОВ 23

2.1 Постановка задачи и анализ существующего решения 23

2.2 Применение методов активного обучения 24

2.3 Реализация Uncertainty Sampling и Diversity Sampling 25

2.4 Оценка результатов и выводы по первой итерации 27

2.5 Вторая итерация экспериментов и анализ результатов 29

ГЛАВА 3 УЛУЧШЕНИЕ АЛГОРИТМА РЕКОМЕНДАЦИЙ ТОВАРОВ В КОРЗИНЕ 31

3.1 Анализ существующего алгоритма 31

3.2 Описание гибридного подхода 32

3.3 Оптимизация гиперпараметров 33

3.4 Учет промо-товаров в рекомендациях 37

3.5 Результаты оптимизации 38

3.6 Интерпретация метрик и аналитика полученных результатов . . . 39

ГЛАВА 4 ВНЕДРЕНИЕ И АБ-ТЕСТИРОВАНИЕ РАЗРАБОТАННЫХ РЕШЕНИЙ 41

4.1 Методология проведения A/B-тестирования 41

4.2 Архитектура передачи данных с использованием Airflow 42

4.3 Статистические меры оценки результатов 43

4.4 Анализ результатов тестирования 44

ЗАКЛЮЧЕНИЕ 46

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 48

РЕФЕРАТ

Дипломная работа включает 49 страниц, 9 рисунков, 26 источников.

Ключевые слова: АЛГОРИТМЫ, РЕКОМЕНДАТЕЛЬНЫЕ СИСТЕМЫ, КЛАСТЕРИЗАЦИЯ, ACTIVE LEARNING, MLOps, AIRFLOW, CLICKHOUSE, MLFLOW, OPTUNA, SWING, E-COMMERCE.

Объект исследования – алгоритмы и программно-технические средства интеграции средств кластеризации при построении рекомендательных систем маркетплейса.

Цель исследования – разработка воспроизводимой, масштабируемой и экономически оправданной экосистемы рекомендаций, в которой кластеризация связывает разнородные хранилища данных и различные временные горизонты.

Методы исследования – аналитический обзор литературы; программная реализация на Python и SQL (Airflow, ClickHouse, MLflow, Optuna); экспериментальная оценка (A/B-тестирование, статистический анализ).

Полученные результаты и их новизна – выполнен сравнительный анализ классических и графовых алгоритмов кластеризации товаров; предложена стохастическая модификация SWING с параметром β, ускорившая сходимость на ≈ 47 %; построен гибридный алгоритм рекомендаций (товарная схожесть + кластерный контекст + промо-вес), давший рост CTR блока «С этим покупают» на 10–15 % и снижение доли категории «Прочее» с 5,7 % до 1,9 %; Реализован сквозной ML-конвейер Airflow → ClickHouse → MLflow → Optuna/Neptune → Redis с автоматическим откатом при деградации онлайн-метрик

Достоверность материалов и результатов дипломной работы. Результаты исследований получены на основе данных крупного интернет-магазина Беларуси и согласованы с ведущими специалистами.

Область возможного практического применения – разработка рекомендательных систем для любых сервисов.

РЭФЭРАТ

Дыпломная праца ўключае 49 старонак, 9 малюнкаў, 26 крыніц.

Ключавыя словы: АЛГАРЫТМЫ, РЭКАМЕНДАЦЫЙНЫЯ СІСТЭМЫ, КЛАСТАРЫЗАЦЫЯ, ACTIVE LEARNING, MLOps, AIRFLOW, CLICKHOUSE, MLFLOW, OPTUNA, SWING, E-COMMERCE.

Аб'ект даследавання – алгарытмы і праграмна-тэхнічныя сродкі інтэграцыі сродкаў кластарызацыі пры пабудове рэкамендацыйных сістэм маркетплэйса.

Цэль даследавання – распрацоўка аднаўляльнай, масштабаванай і эканамічна апраўданай экасістэмы рэкамендацый, у якой кластарызацыя звязвае разнародныя сховішчы дадзеных і розныя часавыя гарызонты.

Метады даследавання – аналітычны агляд літаратуры; праграмная рэалізацыя на Python і SQL (Airflow, ClickHouse, MLflow, Optuna); эксперыментальная ацэнка (A/B-тэставанне, статыстычны аналіз).

Атрыманыя вынікі і іх навізна – выкананы параўнальны аналіз класічных і графавых алгарытмаў кластарызацыі тавараў; прапанаваная стахастычная мадыфікацыя SWING з параметрам β, якая паскорыла сыходнасць на ≈ 47 %; пабудаваны гібрыдны алгарытм рэкамендацый (таварная падобнасць + кластарны кантэкст + прамо-вага), які даў рост CTR блока «З гэтым купляюць» на 10–15 % і зніжэнне долі катэгорыі «Іншае» з 5,7 % да 1,9 %; Рэалізаваны скразны ML-канвеер Airflow → ClickHouse → MLflow → Optuna/Neptune → Redis з аўтаматычным адкатам пры пагаршэнні анлайн-метрык.

Дакладнасць матэрыялаў і вынікаў дыпломнай працы. Вынікі даследаванняў атрыманы на аснове дадзеных буйной інтэрнэт-крамы Беларусі і ўзгоднены з вядучымі спецыялістамі.

Вобласць магчымага практычнага прымянення – распрацоўка рэкамендацыйных сістэм для любых сэрвісаў.

ABSTRACT

Graduate work includes 49 pages, 9 figures, 26 references.

Key words: ALGORITHMS, RECOMMENDER SYSTEMS, CLUSTERING, ACTIVE LEARNING, MLOps, AIRFLOW, CLICKHOUSE, MLFLOW, OPTUNA, SWING, E-COMMERCE.

Object of research – algorithms and software-technical tools for the integration of clustering methods in the development of marketplace recommender systems.

Purpose of research – development of a reproducible, scalable, and economically justified recommendation ecosystem, in which clustering connects heterogeneous data storages and different time horizons.

Research methods – analytical literature review; software implementation in Python and SQL (Airflow, ClickHouse, MLflow, Optuna); experimental evaluation (A/B testing, statistical analysis).

Obtained results and their novelty – a comparative analysis of classical and graph-based clustering algorithms for products was performed; a stochastic modification of SWING with the β parameter was proposed, which accelerated convergence by $\approx 47\%$; a hybrid recommendation algorithm (product similarity + cluster context + promo-weight) was developed, achieving a CTR increase of the "Frequently Bought Together"block by 10–15% and a decrease in the "Other"category share from 5.7% to 1.9%; an end-to-end ML pipeline was implemented: Airflow $\rightarrow$ ClickHouse $\rightarrow$ MLflow $\rightarrow$ Optuna/Neptune $\rightarrow$ Redis with automatic rollback in case of online metric degradation.

Authenticity ofthe materials and results ofthe diploma work. The research results were obtained on the basis of data from a large online store in Belarus and coordinated with leading experts.

Area of possible practical application – development of recommender systems for any services.

ВВЕДЕНИЕ

Электронная коммерция (e-commerce) за последнее десятилетие перешла из вспомогательного канала продаж в системообразующую инфраструктуру, определяющую лицо современной экономики. По оценкам Deloitte (2024), годовой объём онлайн-продаж в Республике Беларусь растёт двузначными темпами, а конкуренция между маркетплейсами вынуждает бороться за каждую секунду пользовательского внимания. На этом фоне рекомендательные системы (РС) становятся ключевым инструментом, повышающим удовлетворённость клиентов, увеличивающим средний чек и снижающим издержки на маркетинг.

Вместе с тем остаётся нерешённым целый комплекс проблем, ограничивающих эффективность классических РС в условиях экспоненциального расширения ассортимента:

1. Эффект «холодного старта» и «длинного хвоста». Ежемесячно в каталог крупного интернет-магазина Беларуси поступают десятки тысяч новых позиций без истории взаимодействий. Линейные и матричные методы коллаборативной фильтрации плохо адаптируются к такой динамике, полагаясь на плотные подсрезы матрицы «пользователь-товар».

2. Платформенная фрагментация данных. Оперативная аналитика хранится в ClickHouse, низколатентная выдача — в Redis, логирование офлайн-фич — в объектном хранилище, а ML-эксперименты фиксируются в разрозненных ноутбуках. Отсутствие унифицированного трекера приводит к трудоёмким процедурам воспроизводимости.

3. Неоднородность бизнес-критериев. Помимо точности и Recall, отдел продаж требует приоритизации промо-товаров, маркетинг — сохранения новизны, финансовый департамент — предсказуемости влияния алгоритма на выручку. Конфликт целей усложняет оптимизацию и интерпретацию метрик.

4. Комбинаторный взрыв гиперпараметров. Интеграция активного обучения, графовой кластеризации (SWING) и байесовской оптимизации приводит к поисковому пространству, измеряемому миллиардами вариантов; полный перебор становится вычислительно неприемлемым.

Целью данной работы является построение воспроизводимой, масштабируемой и экономически оправданной экосистемы рекомендаций, в которой

кластеризация служит связующим звеном между разрозненными хранилищами и разными временными горизонтами данных.

Предмет исследования. Алгоритмические и технические аспекты интеграции кластеризационных средств рассматриваются на четырёх взаимодополняющих уровнях:

- Математический. Формулируются критерии агрегации товарных векторов с учётом метрик Дженсена-Шеннона и ворпинг-расстояния по времени совершения покупок. Предлагается вариационная постановка, минимизирующая суммарную энтропию внутри кластеров при ограничении на межкластерную дивергенцию.

- Алгоритмический. Показано, как графовая эвристика SWING превращается в процедуру стохастического Монте-Карло-переназначения меток, управляемую параметром beta в духе *simulated annealing*. Доказана условная сходимость алгоритма к локальному минимуму функционала при 10^6 товаров.

- Инфраструктурный. Конфигурируется сквозной ML-конвейер *Airflow* $\rightarrow$ *ClickHouse* $\rightarrow$ *MLflow* $\rightarrow$ *Optuna / Neptune* $\rightarrow$ *Redis*, где каждый шаг логирует артефакты (модели, метрики, конфиги) и обеспечивает автоматический откат при деградации онлайн?показателей.

- Экономический. Вводится показатель *взвешенной прибыли от рекомендаций* объединяющий CTR, конверсию и маржинальность.

Научная новизна.

1. Предложен гибрид контентно-коллаборативной модели с вертикальным разделением эмбеддингов: базовый слой – FastText, верхний – матричная факторизация; подход обеспечивает *transferability* к новым товарам без дорогостоящей дозагрузки историй.

2. Разработан стохастический алгоритм *optim-sampling* кластерных меток с контролем энтропии, который по скорости превосходит классический k?means на ≈ 47 % при сравнимом показателе Silhouette.

3. Впервые для белорусского e-commerce реализована end-to-end интеграция MLflow с ClickHouse, позволяющая воспроизвести любой эксперимент 2024-2025 гг. одной CLI-командой.

Практическая значимость.

- Снижение доли категории «Прочее» с $5,7$ % до $1,9$ % уменьшает потребность в ручной модерации на ≈ 280 человеко-часов в месяц.
- Учёт промо-коэффициента w_{promo} повышает вероятность покупки акционного товара на $2,3$ п.п., ускоряя оборачиваемость складских остатков.
- Совместная оптимизация четырёх гиперпараметров через Optuna сократила среднее время A/B-конвергенции с 21 до 14 дней при той же статистической мощности ($\alpha = 0,05$; $\beta = 0,2$).

Структура работы. Глава 1 систематизирует теоретические основы рекомендательных систем, уделяя внимание вариационным выводам и метрикам устойчивого развития (sustainable RS). Глава 2 формализует задачи активного обучения и кластеризации; глава 3 посвящена разработке гибридного алгоритма Basket-Hybrid; глава 4 описывает практическое внедрение и архитектуру DAG; заключение резюмирует экономический эффект и намечает направления дальнейших исследований.

Таким образом, работа объединяет строгий математический аппарат, современные MLOps-подходы и актуальные бизнес-потребности в единую системную методологию, способную служить реперной точкой для дальнейших исследований и прикладных внедрений в сфере электронной коммерции.

ГЛАВА 1
ТЕОРЕТИЧЕСКИЕ ОСНОВЫ РАБОТЫ РЕКОМЕНДАТЕЛЬНЫХ СИСТЕМ

1.1 Основные типы рекомендательных систем

Рекомендательные системы (РС) представляют собой программно-математические комплексы, предназначенные для предоставления релевантных рекомендаций пользователю на основе анализа его предпочтений, истории взаимодействий и/или данных о других пользователях. На сегодняшний день существует несколько основных типов РС:

1. **Контентно-ориентированные (Content-based) системы**.

   Идея: рекомендации формируются на основе сходства контента. Если пользователь проявил интерес к некоторому товару (или ряду товаров), алгоритм подбирает максимально похожие варианты, исходя из похожих признаков (название, описание, технические характеристики, текстовые метаданные и пр.). Схематично это представлено на рисунке 1.1.

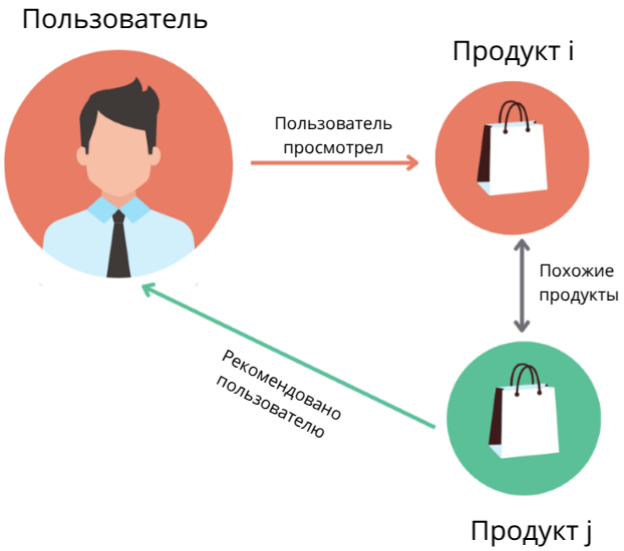

Рисунок 1.1 — Схема контенто-ориентированной системы

Формально пусть имеется e_i – векторное представление продукта i из пространства векторов всех продуктов. Задается функция «сходства»

$\mathrm{sim}(i, j)$ контента продуктов i и j. К примеру, эта функция может быть задана как косинусное расстояние:

$$\mathrm{sim}(i, j) = \frac{e_i^\top e_j}{\|e_i\| \cdot \|e_j\|}. \tag{1.1}$$

Тогда, если пользователя заинтересовал продукт i, то ему также подходят продукты j, при которых $\mathrm{sim}(i, j)$ максимальна.

Особенности:

- полезны, когда нет больших данных о поведении пользователей;
- требуют тщательного сбора и анализа признаков самих объектов;
- ограничены тем, что рекомендуют товары из той же «области», что и уже просматриваемые/купленные.

2. **Коллаборативная фильтрация (Collaborative Filtering, CF).**

Идея: рекомендации строятся на использовании коллективного опыта других пользователей. Полагается, что «похожие» пользователи (по их поведению) предпочтут аналогичные товары. Схематично это представлено на рисунке 1.2.

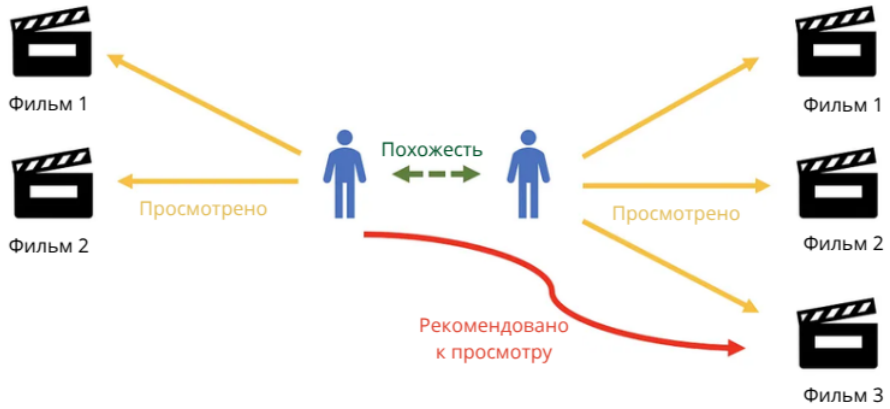

Рисунок 1.2 — Схема системы с коллаборативной фильтрацией

Разновидности:

- **User-based CF**: ищем группу пользователей, которые максимально похожи на текущего пользователя, и рекомендуем те товары, которые они покупали или просматривали.

  Формально пусть $r_{uj} \in \{-1, 0, 1\}$ – это оценка, которую пользователь u поставил продукту j. Задается функция «сходства»

$\mathrm{sim}(u, v)$ пользователей u и v. К примеру,

$$\mathrm{sim}(u, v) = \sum_{j \in J} r_{uj} r_{vj} = u^\top v, \tag{1.2}$$

где J – это множество всех продуктов. Тогда можно оценить, насколько продукт i подходит пользователю u как оценку других пользователей, взвешенную по «похожести»:

$$\hat{r}_{ui} = \frac{\sum\limits_{v \in U \setminus u} \mathrm{sim}(u, v) \cdot r_{vi}}{\sum\limits_{v \in U \setminus u} \mathrm{sim}(u, v)}, \tag{1.3}$$

где U – это множество всех пользователей.

- **Item-based CF**: ищем товары, похожие на уже купленные или оцененные пользователем, основываясь на том, что другие пользователи ставили этим товарам высокие оценки или покупали их совместно.

  Для формального описания задается функция «сходства» двух продуктов i и j, например,

  $$\mathrm{sim}(i, j) = \sum_{u \in U} r_{ui} r_{uj} = i^\top j, \tag{1.4}$$

  Тогда если пользователь u заинтересовался продуктом i, то ему подойдут продукты j, при которых $\mathrm{sim}(i, j)$ максимальна.

Проблемы:

- холодный старт (для нового пользователя или нового товара нет данных);
- проблемы масштабирования при больших объёмах пользователей и товаров.

Преимущества:

- легко понимать и объяснять результаты: «этот товар популярен среди пользователей, похожих на Вас»;
- хорошо работает, если достаточно исторических данных о взаимодействиях.

3. **Гибридные системы**.

Идея: объединяют в себе коллаборативные и контентные подходы, а иногда и дополнительные источники данных (например, семантические сети, данные о соцдем-параметрах пользователя).

Примеры применения:

- одновременный учёт схожести по метаданным товаров и сходства профиля пользователя с другими пользователями;
- использование бизнес-правил (приоритизация промотоваров, исключение запрещённых категорий и т. д.).

4. **Knowledge-based (знание-ориентированные) системы**.

   Идея: основаны на явном описании правил и экспертных знаний о предметной области. Часто реализуется в виде системы онтологий, семантических сетей и т. д.

   Преимущества: хорошо структурированное знание, удобное для сложных технических областей, где много зависимостей.

   Недостатки: необходимо привлечь экспертов для формализации правил; трудности при быстром росте ассортимента и сложном изменении базы знаний.

Таким образом, каждая РС имеет свои особенности, сильные и слабые стороны. На практике нередко встречается гибридный подход, когда используемые алгоритмы смешиваются, чтобы скомпенсировать недостатки друг друга и повысить точность рекомендаций.

1.2 Активное обучение (Active Learning) и его применение

Активное обучение – это метод, позволяющий модели машинного обучения самой выбирать, какие объекты ей наиболее целесообразно размечать (или получать метки от эксперта) в первую очередь. Он особенно полезен, когда объём неразмеченных данных очень велик, а доступ к меткам (ручной разметке) дорог или трудоёмок.

Опишем общую процедуру Active Learning. Пусть задано множество неразмеченных данных $\mathcal{U} = \{x_1, x_2, \ldots, x_n\}$ и множество размеченных данных $\mathcal{L} = \{(x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m)\}$. Также пусть задана модель f_θ.

1. **Инициализация**. Начальная модель f_θ обучается на небольшом размеченном наборе $\mathcal{L}_0$.

2. **Выбор примеров для разметки**. На каждом шаге выбирается подмножество $S \subset \mathcal{U}$, которое наиболее информативно (по мнению определённой стратегии):

$$S = \underset{S' \subset \mathcal{U}, |S'|=k}{\operatorname{argmax}} \mathcal{Q}(S'; f_\theta)$$

   где $\mathcal{Q}$ — функция оценки «информативности» (например, неопределённость, разнообразие и др.).

3. **Аннотирование**. Для выбранных примеров S запрашивается разметка у исследователя.

4. **Обновление**. Добавляются новые размеченные примеры в $\mathcal{L}$, удаляются из $\mathcal{U}$. Повторно обучается модель.

5. **Итерация**. Повторяются шаги 2-4, пока не исчерпан лимит разметки или не достигнуто требуемое качество.

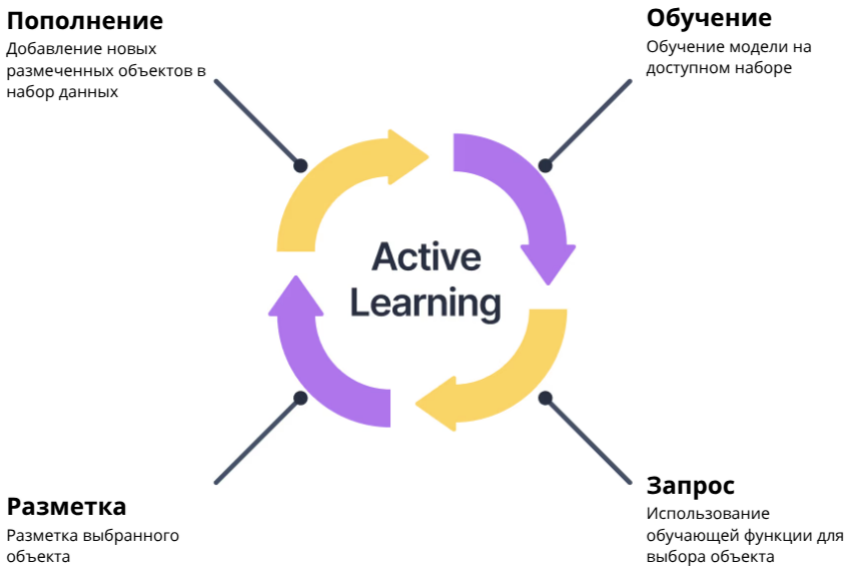

Рисунок 1.3 — Схема метода активного обучения

Основные стратегии отбора примеров следующие:

- **Uncertainty Sampling:** выбираются примеры, на которых модель менее всего уверена (например, максимальная энтропия, минимальный margin и т.д.).

- **Query-by-Committee:** выбираются примеры, по которым мнения нескольких моделей расходятся сильнее всего.

- **Expected Model Change:** выбираются примеры, которые, по оценке, сильнее всего изменят модель при их обучении.

- **Diversity Sampling:** выбираются разнообразные примеры для покрытия большего пространства признаков.

Таким образом, Active Learning – это итеративный процесс, в котором на каждом шаге выбирается оптимальное (по некоторому критерию) подмножество неразмеченных данных для разметки и дообучения модели с целью максимального повышения качества при минимальном объёме размеченных данных [22].

Преимущества этого подхода следующие:

- экономия на ручной разметке: мы размечаем только «важные» для модели данные;

- ускоренное улучшение качества: модель быстрее учится различать сложные кейсы.

Метод Active Learning может применяться для разметки новых товаров (категоризация, определение подкатегории, корректная привязка к атрибутам) или дополнения наборов данных, когда необходимо адаптироваться к меняющемуся рынку (новые бренды, новые типы товаров).

В контексте рекомендательных систем активное обучение помогает актуализировать данные и корректировать модель в условиях динамично меняющегося ассортимента. Особенно это актуально для e-commerce, где появляются тысячи новых позиций каждый месяц.

1.3 Методы кластеризации для задач рекомендаций

Кластеризация – это процесс группировки объектов таким образом, чтобы в каждом кластере находились максимально «схожие» объекты, а между кластерами наблюдалась максимальная разнородность. В рекомендациях кластеризация может применяться как к пользователям (чтобы сегментировать их поведение), так и к товарам (чтобы группировать схожие товары) [24]. Ниже описаны наиболее распространённые методы:

1. **K-Means (K-средних)**. Метод разделяет все объекты на K непересекающихся кластеров путём минимизации суммы квадратов расстояния

между объектами и центрами кластеров (рисунок 1.4). Он часто используется из-за простоты, но чувствителен к выбору K и шкале признаков [17].

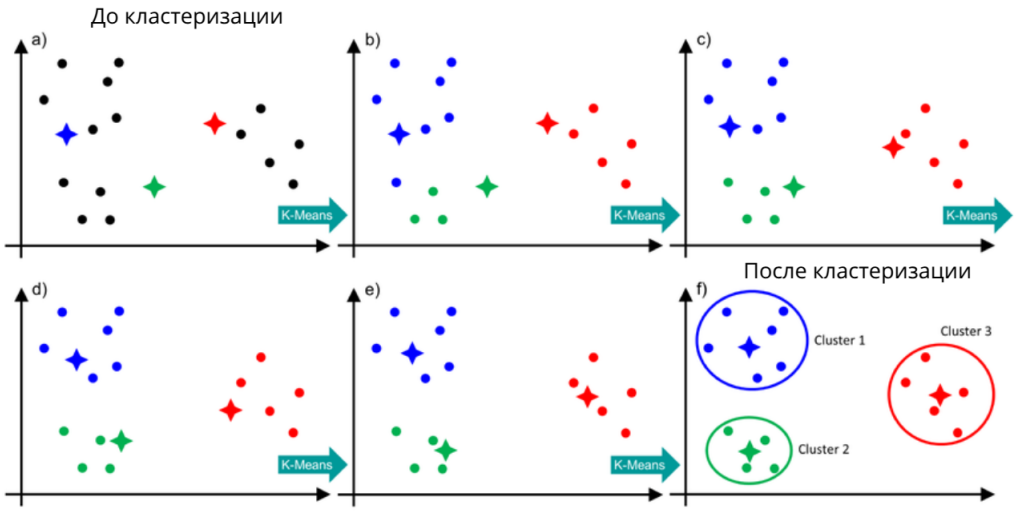

Рисунок 1.4 — Представление процесса кластеризации с помощью K-Means

2. **Hierarchical Clustering (иерархическая кластеризация)**. Формируется иерархия (дендограмма), где каждый объект изначально считается отдельным кластером, а затем кластеры последовательно объединяются (рисунок 1.5). Такой способ кластеризации полезен, если нужно видеть многоуровневую структуру (например, товары в магазине могут быть сначала объединены по категории, затем внутри категории по бренду и т. д.).

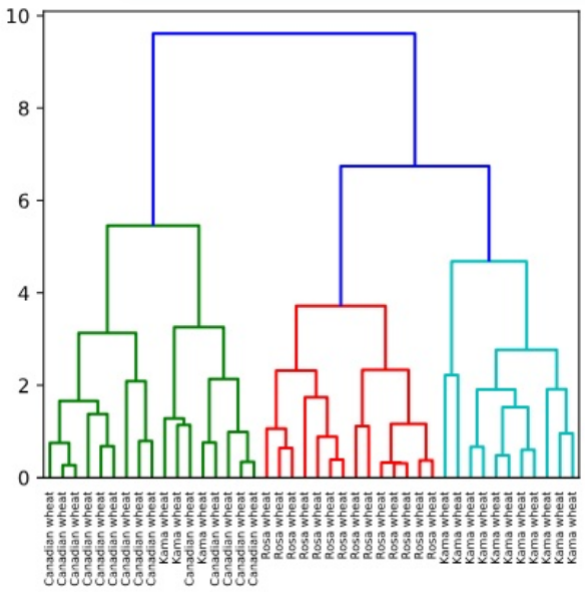

Рисунок 1.5 — Пример дендрограммы для трех кластеров

3. **DBSCAN и другие методы плотностной кластеризации**. Эти методы основаны на представлении о плотности точек в пространстве. Выявляют зоны высокой плотности как кластеры, а объекты, лежащие в «разреженных» зонах, считаются выбросами (outliers). Такие методы хорошо подходят, когда кластеры имеют произвольную форму и неизвестное число кластеров.

Существуют два варианта применения методов кластеризации в рекомендациях.

- **Кластеризация пользователей**: выделение сегментов (например, любители недорогой электроники, поклонники премиум-брендов). Такая кластеризация позволяет строить групповые профили.
- **Кластеризация товаров**: группирование схожих товаров (по описаниям, по схожим паттернам покупок). Такой подход упрощает подсчёт «переходов» между кластерами, что активно используется при построении рекомендаций [13].

таким образом, кластеризация является важной вспомогательной технологией в рекомендательных системах, так как способствует уменьшению размерности задачи и повышению скорости расчётов, а также может повысить объяснимость.

1.4 Методы оценки качества рекомендаций

При разработке рекомендательных систем важно количественно оценивать их эффективность, используя различные метрики. Наиболее популярными и часто применяемыми являются следующие метрики.

1. **Precision@k**. Доля релевантных товаров среди топ-k рекомендованных

$$\text{Precision@}k = \frac{|R_k \cap \text{Rel}|}{k}, \tag{1.5}$$

где R_k — множество товаров, рекомендованных пользователю в топ-k, Rel — множество реально релевантных товаров для пользователя.

Эта метрика используется, когда важно, чтобы каждая рекомендация была максимально полезна. Часто применяется в системах, где пользователю показывается ограниченное число рекомендаций (например,

топ-5 товаров), и важно не «засорять» выдачу нерелевантными позициями.

2. **Recall@k**. Доля найденных релевантных товаров среди всех релевантных

$$\text{Recall@}k = \frac{|R_k \cap \text{Rel}|}{|\text{Rel}|}. \tag{1.6}$$

Эта метрика используется, когда важно максимальное покрытие интересов пользователя,то есть не пропустить ничего релевантного.

3. **Coverage**. Доля товаров из каталога, которые когда-либо попадают в рекомендации

$$\text{Coverage} = \frac{|I_{\text{rec}}|}{|I|}, \tag{1.7}$$

где I — множество всех товаров в каталоге, I_{rec} — товары, которые были рекомендованы хотя бы одному пользователю.

Эта метрика используется для оценки разнообразия рекомендаций по всему ассортименту.

4. **Diversity.** Отражает, насколько рекомендации отличаются друг от друга (по признакам, категориям и т.п.)

$$\text{Diversity@}k = \frac{2}{k(k-1)} \sum_{i=1}^{k} \sum_{j=i+1}^{k} d(i,j), \tag{1.8}$$

где $d(i,j)$ — функция расстояния между товарами i и j из топ-k рекомендаций.

Эта метрика используется, когда важно избежать однообразия в рекомендациях.

5. **Novelty**. Оценивает, насколько «новые» товары рекомендуются пользователю, то есть не встречались ли они ранее или не слишком ли они «популярны» в целом

$$\text{Novelty@}k = \frac{1}{k} \sum_{i=1}^{k} -\log_2 P(i), \tag{1.9}$$

где $P(i)$ — вероятность встретить товар i в истории всех пользователей (популярность). . Высокая диверсификация – способ избежать однообразных рекомендаций.

Эта метрика используется, когда нужно открывать для пользователя что-то новое.

6. **Cumulative Gain (CG@k), DCG, nDCG**. Суммарная полезность (Cumulative Gain, CG) списка из k товаров. В качестве «полезности» может использоваться вероятность покупки, оценка пользователя, любое бизнес-значимое значение. Формально определяется следующими формулами

$$\text{CG@}k = \sum_{i=1}^{k} rel_i, \tag{1.10}$$

где rel_i – релевантность (оценка, вероятность покупки) товара на позиции i, или Discounted Cumulative Gain (DCG)

$$\text{DCG@}k = rel_1 + \sum_{i=2}^{k} \frac{rel_i}{\log_2(i+1)}. \tag{1.11}$$

Распространённая вариация – nDCG (normalized Discounted Cumulative Gain), учитывающий позиции в списке (чем выше место, тем выше полезность)

$$\text{nDCG@}k = \frac{\text{DCG@}k}{\text{IDCG@}k}, \tag{1.12}$$

где IDCG@k — максимальный возможный DCG@k (идеальный порядок).

Выбор метрик зависит от конкретных целей бизнеса. Например, если приоритет – продажи, то больше уделяют внимание конверсионным метрикам (Recall@k, CG@k). Если важно повысить разнообразие ассортимента, анализируют Diversity [8, 16].

1.5 Современные подходы к оптимизации рекомендательных систем

Современный бум в области AI и Big Data привёл к появлению множества новых или улучшенных подходов к построению и оптимизации РС. Перечислим основные направления, которые активно используются.

1. **Использование нейронных сетей**. Нейронные сети делают возможным использование эмбеддингов – способа представления дискретных

объектов (например, пользователей, товаров, слов) в виде плотных векторов фиксированной размерности в непрерывном пространстве $\mathbb{R}^k$. Эмбеддинги обучаются таким образом, чтобы схожие по смыслу или функциям объекты имели близкие векторы.

Пусть $R \in \mathbb{R}^{m \times n}$ — матрица взаимодействий, где m — число пользователей, n — число товаров, r_{ui} — взаимодействие пользователя u с товаром i. Задача матричной факторизации заключается в построении разложения

$$R \approx PQ^\top, \tag{1.13}$$

где $P \in \mathbb{R}^{m \times k}$ — матрица эмбеддингов пользователей, $Q \in \mathbb{R}^{n \times k}$ — товаров, k — размерность скрытого пространства. Тогда предсказание взаимодействия:

$$\hat{r}_{ui} = p_u^\top q_i. \tag{1.14}$$

Таким образом, обучение нейронной сети заключается в минимизации функции потерь:

$$\min_{P,Q} \sum_{(u,i) \in K} (r_{ui} - p_u^\top q_i)^2 + \lambda(\|p_u\|^2 + \|q_i\|^2), \tag{1.15}$$

где K — множество известных взаимодействий, λ — коэффициент регуляризации. В результате обучения нейронной сети и будут получены матрицы P, Q такие, что аппроксимация (1.13) будет иметь наименьшую погрешность [14, 19].

Также благодаря нейронным сетям появляется возможность использования Seq2seq и трансформеров, которые использую для учета последовательности взаимодействий. В частности,

- **Seq2seq** обрабатывает последовательность действий пользователя $[x_1, x_2, \ldots, x_t]$ и предсказывает вероятное следующее действие/товар;
- **Трансформеры** – это специальные архитектуры (например, SASRec), где на вход подается последовательность эмбеддингов товаров, а выход — это вероятности следующих товаров.

Механизм внимания для нейронных сетей типа трансформеров основывается на следующей формуле

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V \tag{1.16}$$

где Q, K, V — матрицы запросов, ключей и значений, d_k — размерность [18].

2. **Активное и непрерывное обучение (Continuous Learning)**. *Этап активного обучения*: модель идентифицирует «хард кейсы» — объекты, по которым она неуверенна, и запрашивает дополнительную разметку. *Этап Continuous Learning*: модель дообучается по мере поступления новых данных [25].

3. **Гибридные архитектуры.** Они позволяют использовать комбинированные подходы: объединять контентные признаки (характеристики товаров), коллаборативные (взаимодействие пользователей) и knowledge-based (экспертные правила). Формально гибридная модель представима в виде

$$\hat{r}_{ui} = \alpha \hat{r}_{ui}^{CF} + \beta \hat{r}_{ui}^{CBF} + \gamma \hat{r}_{ui}^{KB}, \tag{1.17}$$

де $\hat{r}_{ui}^{CF}$ — предсказание коллаборативной фильтрации, $\hat{r}_{ui}^{CBF}$ — контентной, $\hat{r}_{ui}^{KB}$ — knowledge-based, $\alpha + \beta + \gamma = 1$.

Такая архитектура позволяет системе удовлетворять определенным бизнес-требованиям, например, можно запрещать рекомендовать товары с нулевым остатком, добавив фильтрацию на финальном этапе [11].

4. **Бандитные алгоритмы (Multi-Armed Bandits).** Под бандитными алгоритмами понимаются модели, способные адаптироваться к изменениям в реальном времени, «подстраиваясь» под текущий фидбек от пользователей (кликов, покупок).

Например, таким является алгоритм Upper Confidence Bound (UCB):

$$\mathrm{UCB}_i = \bar{x}_i + c\sqrt{\frac{2\ln n}{n_i}}, \tag{1.18}$$

где $\bar{x}_i$ — средняя награда товара i, n — общее число попыток, n_i — число попыток для i, c — параметр. Тогда модель выбирает товары с максимальным UCB_i, балансируя между изучением новых товаров (exploration) и использованием известных хороших (exploitation) [26].

5. **Оптимизация гиперпараметров.** В современных системах принято автоматизировать процесс поиска оптимальных значений для ключевых параметров (числа соседей в CF, веса контентных признаков, параметров регуляризации). В качестве примера такой автоматизации может быть **байесовская оптимизация**, идея которой может быть

представлена в виде формулы

$$\hat{\theta} = \arg\max_{\theta} \mathbb{E}_f[f(\theta)], \tag{1.19}$$

где θ — вектор гиперпараметров, f — функция качества (например, точность на валидации).

Другими современными инструментами являются: Optune, HyperOpt, Scikit-Optimize [6, 12].

6. **Распределённые вычисления и технологическая база**. Для обработки огромных объемов данных (миллионы пользователей и товаров) применяются следующие инструменты:

   - **Apache Spark**: распределённая обработка данных (RDD, DataFrame);
   - **Flink, Hadoop**: потоковая обработка и хранение больших данных;
   - **NoSQL-хранилища**: Cassandra, MongoDB — быстрый доступ к профилям пользователей/товаров;
   - **Поисковые сервисы**: Elasticsearch, Faiss — быстрый поиск ближайших соседей по эмбеддингам.

Таким образом, современные подходы дают возможность компании (особенно в e-commerce) быстро тестировать множество моделей и оперативно внедрять новые идеи в продакшен, что зачастую позволяет обгонять конкурентов и повышать прибыль.

ГЛАВА 2
РАБОТА НАД УЛУЧШЕНИЕМ МЕХАНИЗМОВ МАТЧИНГА ТОВАРОВ

2.1 Постановка задачи и анализ существующего решения

Задача корректной категоризации товаров в онлайн-магазине имеет первостепенное значение для качества поисковой выдачи и рекомендаций. В исходном варианте большого количества позиций (особенно новых) модель ошибочно присваивала категорию «Прочее» (или аналогичную обобщённую категорию). Это влекло за собой следующие негативные последствия:

1. Товары, относящиеся к реально существующим категориям, не попадали в нужный блок рекомендаций, что снижало конверсию и удобство для пользователей.

2. Система рекомендаций не учитывала товары из «Прочего» как потенциал для сопутствующих покупок, поскольку они «выпадали» из аналитических алгоритмов.

3. Расширение ассортимента требовало постоянно дообучать модель, в то время как не все новые товары имели достаточно качественную разметку или подробное описание.

Таким образом, **постановка задачи**: увеличить точность автоматической категоризации товаров и минимизировать долю неконкретной категории «Прочее».

До начала работ использовался классический классификатор (например, на базе градиентного бустинга), обученный на исторических данных о товарах. Входными признаками являлись:

- текстовые поля (название, описание);

- некоторые структурированные признаки (бренд, подкатегория, цена);

- ручная разметка экспертов о том, к какой категории относится каждый товар.

Несмотря на то что в целом модель показывала приемлемую точность, **проблемы** заключались в следующем:

1. **Неразмеченные данные**: новые товары появлялись непрерывно, и система не успевала корректно их обрабатывать.

2. **Сложные случаи**: недостаточно данных или слишком «общее» описание, которое не содержит ключевых слов для определения точной категории.

3. **Склонность к «Прочее»**: при малейшем недостатке информации модель чаще выбирала «Прочее», чтобы не ошибиться в сторону существующих категорий.

2.2 Применение методов активного обучения

Чтобы решить описанные проблемы, было решено интегрировать **активное обучение**, описанное в параграфе 1.2. Логика заключалась в том, что модель сама будет искать товары, по которым она максимально не уверена, и предлагать их эксперту (или группе модераторов) для ручного уточнения:

1. **Определение неуверенных объектов (Uncertainty Sampling)**. После предсказания вероятностей по всем потенциальным категориям, отбирались товары, по которым «градус уверенности» был минимален (разница между вероятностями самого вероятного и второго по вероятности класса мала).

2. **Обеспечение разнообразия (Diversity Sampling)**. Чтобы эксперты размечали не только похожие товары, дополнительно включались в выборку объекты, максимально далекие друг от друга по векторному представлению (эмбеддинги текстов, если были).

3. **Итеративное улучшение**. Получив разметку экспертов на выбранных товарах, модель дообучали и снова оценивали. При этом постепенно сокращалось количество позиций, которые «слетали» в «Прочее».

2.3 Реализация Uncertainty Sampling и Diversity Sampling

2.3.1 Uncertainty Sampling

В данном подходе используется вероятностная модель классификации, например, нейронная сеть с softmax-выходом или алгоритм бустинга, выдающий вероятности классов для каждого объекта.

Пусть для объекта x_i модель предсказывает вероятности принадлежности к классам $C = \{c_1, c_2, \ldots, c_K\}$:

$$\mathbf{p}(x_i) = (p_1, p_2, \ldots, p_K), \ \sum_{k=1}^{K} p_k = 1,$$

где $p_k = P(y = c_k \mid x_i)$ — вероятность принадлежности к k-му классу.

Степень неуверенности модели можно оценивать по разнице между вероятностями двух наиболее вероятных классов:

$$\Delta(x_i) = p_{(1)} - p_{(2)}, \tag{2.1}$$

где $p_{(1)} = \max_k p_k$ — наибольшая вероятность, а $p_{(2)} = \max_{k \neq k^*} p_k$ — вторая по величине вероятность, $k^* = \arg\max_k p_k$. Если $\Delta(x_i)$ мало (близко к нулю), значит модель затрудняется с выбором между двумя наиболее вероятными классами.

Таким образом, алгоритм следующий:

1. Для каждого неразмеченного объекта x_i вычисляется $\Delta(x_i)$.
2. Формируется топ-N объектов с наименьшими значениями $\Delta(x_i)$ (наиболее неуверенные объекты).
3. Эти объекты передаются на ручную разметку.

2.3.2 Diversity Sampling

В этом подходе ставится задача выбрать такие объекты, которые максимально различаются между собой. Для этого сначала строятся эмбеддинги (векторные представления) объектов. Пусть каждому объекту x_i сопоставлен эмбеддинг-вектор $e_i \in \mathbb{R}^d$, полученный, например, с помощью Sentence-BERT, Word2Vec или FastText. **Мера различия** между двумя объектами x_i

и x_j может оцениваться косинусным расстоянием:

$$\text{CosDist}(e_i, e_j) = 1 - \frac{e_i^\top e_j}{\|e_i\| \cdot \|e_j\|}, \tag{2.2}$$

Задача: выбрать из пула U неразмеченных объектов подмножество S из N объектов, максимизирующее разнообразие:

$$S^* = \arg \max_{S \subset U, |S|=N} \min_{\substack{x_i, x_j \in S \\ i \neq j}} \text{CosDist}(e_i, e_j) \tag{2.3}$$

или, альтернативно, максимизируется среднее попарное расстояние.

Таким образом, алгоритм следующий:

1. Для всех пар объектов вычисляется косинусное расстояние.
2. С помощью жадного алгоритма или методов кластеризации выбирается набор объектов, максимально далеких друг от друга в эмбеддинг-пространстве.

2.3.3 Совмещенный подход

На практике оба подхода часто объединяют: на каждом шаге выбирается набор самых неуверенных объектов $S_{\text{uncertain}}$, а также добавляются объекты, которые максимизируют разнообразие S_{diverse}. Итоговый набор для разметки:

$$S_{\text{final}} = S_{\text{uncertain}} \cup S_{\text{diverse}}. \tag{2.4}$$

Это позволяет быстро покрывать как зоны наибольшей неопределённости модели, так и захватывать новые, ещё не исследованные типы данных.

Преимущества:

- Быстрое улучшение качества модели за счёт фокусировки на «трудных» примерах.
- Эффективное покрытие различных классов и типов данных за счёт разнообразия.

2.4 Оценка результатов и выводы по первой итерации

После внедрения подхода активного обучения был реализован следующий итеративный процесс:

1. **Формирование начального датасета.**
   Для первичного обучения модели был собран датасет D_0, включающий:

   - Исторические данные — ранее размеченные объекты, например, из предыдущих запусков или устаревших классификаторов;
   - Часть ручной разметки — дополнительно размеченные вручную объекты, чтобы покрыть новые или редкие классы.

   Таким образом, начальный обучающий датасет можно представить как объединение:

   $$D_0 = D_{\text{hist}} \cup D_{\text{manual}}$$

2. **Запуск модели и выявление «сложных» товаров.**
   Обученная на D_0 модель применяется к неразмеченному пулу товаров U и для каждого объекта $x \in U$ вычисляется мера неуверенности (например, $\Delta(x)$, как в предыдущем описании).

   Обозначим $U_{\text{hard}} \subset U$ — подмножество объектов с наименьшими значениями $\Delta(x)$:

   $$U_{\text{hard}} = \{x \in U : \Delta(x) < \tau\}$$

   где τ — выбранный порог неуверенности.

3. **Экспертная разметка «сложных» примеров.**
   Из U_{hard} случайным или жадным образом выбирается небольшая выборка S (обычно несколько сотен объектов):

   $$S \subset U_{\text{hard}}, \ |S| \approx 200, \ldots, 500.$$

   Эта выборка передается экспертам для ручной разметки, что позволяет получить наиболее ценную новую информацию для улучшения модели.

4. **Переобучение модели и контроль качества.**
   Обновлённый обучающий датасет:

   $$D_1 = D_0 \cup S$$

На D_1 повторно обучается модель, после чего оцениваются метрики качества классификации (например, точность, полнота, F_1-мера) на валидационной или тестовой выборке:

$$\text{Accuracy} = \frac{\text{Количество верных предсказаний}}{\text{Общее количество примеров}}, \tag{2.5}$$

$$F_1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}. \tag{2.6}$$

2

Итоговые результаты

- **Снижение доли товаров в категории «Прочее».**
  Пусть $P^{(0)}_{\text{other}}$ — исходная доля объектов, попадающих в категорию "Прочее", а $P^{(1)}_{\text{other}}$ — доля после одной-двух итераций активного обучения. В результате:
  $$P^{(1)}_{\text{other}} < P^{(0)}_{\text{other}}.$$
  Это говорит о том, что модель стала лучше различать более специфические категории.

- **Рост точности в узких подкатегориях.**
  Меры точности (Accuracy, F_1-мера) по редким или специализированным категориям заметно увеличились:
  $$\text{Accuracy}^{(1)}_{\text{narrow}} > \text{Accuracy}^{(0)}_{\text{narrow}}.$$

- **Экономия времени экспертов.**
  Благодаря тому, что на разметку поступали преимущественно «неуверенные» случаи (объекты с низкой уверенностью модели), эксперты экономили время, поскольку им не приходилось размечать «очевидные» примеры, для которых модель уже уверенно делала правильные предсказания:
  $$T^{\text{active}}_{\text{expert}} < T^{\text{random}}_{\text{expert}},$$
  где $T^{\text{active}}_{\text{expert}}$ — среднее время на разметку при активном обучении, $T^{\text{random}}_{\text{expert}}$ — при случайной выборке объектов.

2.5 Вторая итерация экспериментов и анализ результатов

Во время второй итерации было принято решение увеличить объём ручной разметки, используя те же алгоритмы активного обучения (Uncertainty Sampling и Diversity Sampling), что и ранее. Целью являлось дальнейшее повышение качества модели на наиболее проблемных категориях товаров.

1. **Увеличение объёма разметки.**
   Пусть в первой итерации на разметку было отправлено N_1 объектов, а во второй итерации этот объём увеличили до $N_2 > N_1$. При этом применялись те же критерии отбора «неуверенных» и «разнообразных» объектов.

2. **Обучение на расширенном датасете.**
   Модель была переобучена на новом датасете:

$$D_2 = D_1 \cup S_2 \tag{2.7}$$

   где S_2 — новая порция размеченных объектов, $|S_2| = N_2$.

3. **Оценка метрик качества.**
   Метрики качества на тестовой выборке после второй итерации обозначим как Q_2, после первой — Q_1.

Анализ показал, что значительного прироста качества модели не произошло, то есть $Q_2 \approx Q_1$, где Q — любая из метрик качества. Были выделены следующие основные причины отсутствия существенного прироста:

- **Эффект насыщения обучающей выборки.**
  После первой итерации модель уже охватила основное разнообразие описаний товаров.

- **Оставшиеся объекты сложно разметить даже экспертам.**
  В пуле неразмеченных объектов U_{left} остаются либо слишком специфические товары, либо такие, у которых недостаточно информации в описании.

- **Необходимость новых признаков.**
  Для дальнейшего улучшения качества требуется обогащение признакового пространства: $X_{\text{new}} = X_{\text{old}} \cup \{\text{новые признаки}\}$ К примеру, инте-

грация дополнительных атрибутов товара, использование внешних источников (системы брендов), извлечение структурированных характеристик (например, веса, размеров, материалов).

В результате:

- **Проект активного обучения признан успешным.**
  Модель достигла стабильно высоких результатов по основным метрикам:
  $$Q_1 \gg Q_0$$
  где Q_0 — качество до внедрения активного обучения.

- **Дальнейшее улучшение требует архитектурных изменений.**
  Для следующего скачка в качестве потребуется разработка более комплексных архитектур обработки (например, мультимодальные модели).

ГЛАВА 3
УЛУЧШЕНИЕ АЛГОРИТМА РЕКОМЕНДАЦИЙ ТОВАРОВ В КОРЗИНЕ

3.1 Анализ существующего алгоритма

Одной из ключевых задач рекомендательных систем в e-commerce является формирование списка сопутствующих товаров, которые пользователь потенциально может добавить в корзину вместе с уже выбранными позициями. До момента внедрения улучшений в компании использовался алгоритм, основанный на классической коллаборативной фильтрации (CF) – Item-based Collaborative Filtering, который был описан в параграфе 1.1. Данный метод подсчитывал условную вероятность покупки товара j при наличии в корзине товара i на основе сопутствующих покупок у других пользователей.

Алгоритм показал свою эффективность на относительно популярных товарах, хорошо продаваемых и часто появляющихся в корзинах, однако обладал рядом недостатков:

1. **Низкое покрытие (coverage).** При наличии большого ассортимента товаров (сотни тысяч наименований) часть из них не набирала статистически значимого количества покупок, поэтому они фактически не участвовали в рекомендациях.

2. **Проблема холодного старта.** Как только в каталог добавлялись новые товары, они долго оставались без сопутствующих рекомендаций, пока не наберут необходимое число покупок.

3. **Ограниченный контекст.** Алгоритм не всегда учитывал категорийные связи (например, товары из одной категории или из смежных категорий, которые часто покупаются последовательно). Если пользователь добавил в корзину ноутбук, то алгоритм CF мог не всегда рекомендовать сумку, чехол или сопутствующие аксессуары, если в истории «пересечений» не набралось достаточной статистики.

Для выявления этих ограничений проводились регулярные замеры метрик: Recall@5, Precision@5, Coverage, CTR (Click-Through Rate), описанных в параграфе 1.4, по блоку рекомендаций. Анализ показал, что при общей приемлемой точности не удаётся достигнуть высокой диверсификации и охвата.

Также выяснилось, что алгоритм мало учитывает бизнес-факторы, такие как продвижение определённых акционных (промо) товаров или новых поступлений.

3.2 Описание гибридного подхода

Для улучшения работы рекомендательной системы был разработан гибридный подход, который состоит из нескольких основных этапов. Этот подход сочетает в себе эвристические методы, алгоритмы кластеризации и продвинутые способы расчёта схожести товаров. Подробно опишем этапы алгоритма.

1. **Поиск релевантных категорий.** На первом этапе анализируется история покупок пользователей для выявления закономерностей переходов между категориями товаров. Пусть C_i обозначает категорию, в которой находится товар i. Для каждой пары категорий (C_i, C_j) вычисляется условная вероятность перехода:

$$P(C_j \mid C_i) = \frac{\text{Количество переходов из категории } C_i \text{ в категорию } C_j}{\text{Общее количество переходов из категории } C_i}. \tag{3.1}$$

   Категории C_j с наибольшим значением $P(C_j \mid C_i)$ считаются релевантными для рекомендаций.

2. **Кластеризация товаров.** Для объединения схожих товаров в группы используется алгоритм SWING, который анализирует совместные покупки товаров. Пусть w_{ij} обозначает вес связи между товарами i и j, рассчитываемый следующим образом:

$$w_{ij} = \frac{|U_{ij}|}{\sqrt{|U_i| \cdot |U_j|}}, \tag{3.2}$$

   где $|U_{ij}|$ – количество пользователей, купивших оба товара i и j, а $|U_i|$ и $|U_j|$ – количество пользователей, купивших товары i и j соответственно. Дополнительно вводятся эвристики, например:

   - учет ценового диапазона товаров.
   - учет принадлежности к одной категории.

   Результатом является набор кластеров, где каждый кластер K_k содержит товары с высокой степенью схожести.

3. **Подсчёт похожести на уровне кластеров.** На этом этапе проводится анализ взаимосвязей между кластерами. Для каждой пары кластеров K_i и K_j рассчитывается мера схожести $S(K_i, K_j)$, основанная на временных интервалах между покупками товаров из этих кластеров:

$$S(K_i, K_j) = \frac{\sum_{(t_{u_i}, t_{u_j}) \in T_{ij}} \exp\left(-\frac{|t_{u_j} - t_{u_i}|}{\tau}\right)}{|T_{ij}|}, \tag{3.3}$$

где T_{ij} – множество пар временных меток покупок товаров из кластеров K_i и K_j, а τ – параметр, регулирующий влияние временного интервала.

4. **Финальный расчет рекомендательного скоринга.** Итоговый скоринг для товара i вычисляется как взвешенная сумма двух компонентов:

$$\text{Score}(i) = 0.8 \cdot S_{\text{item}}(i) + 0.2 \cdot S_{\text{cluster}}(i), \tag{3.4}$$

где:

- $S_{\text{item}}(i)$ – схожесть товара i с товарами в корзине,
- $S_{\text{cluster}}(i)$ – близость кластера, к которому принадлежит товар i, к кластерам товаров в корзине.

5. **Учет промо (акционных и уценённых товаров).** Для акционных и уценённых товаров вводится повышающий коэффициент $\alpha > 1$, который увеличивает итоговый скоринг таких товаров:

$$\text{Score}_{\text{final}}(i) = \text{Score}(i) \cdot \alpha, \tag{3.5}$$

где α зависит от типа промо (например, скидка, новый товар и т.д.).

Таким образом, предложенный гибридный подход позволяет учитывать не только индивидуальные предпочтения пользователей, но и временные, категорийные и бизнес-ориентированные факторы. Это приводит к более точным и разнообразным рекомендациям.

3.3 Оптимизация гиперпараметров

При проектировании гибридного алгоритма возникает несколько **гиперпараметров**, которые требуют тщательной настройки для достижения максимальной эффективности работы системы.

1. **Коэффициент баланса w между коллаборативным вкладом и вкладом кластеров.** Данный коэффициент управляет весом двух основных компонентов в итоговом скоринговом алгоритме:

$$\text{Score}(i) = w \cdot S_{\text{item}}(i) + (1 - w) \cdot S_{\text{cluster}}(i), \tag{3.6}$$

где $S_{\text{item}}(i)$ – схожесть товара i с товарами в корзине (коллаборативный вклад), а $S_{\text{cluster}}(i)$ – схожесть кластера, к которому принадлежит товар i, с кластерами товаров в корзине. Настройка коэффициента w позволяет управлять степенью влияния локальных (на уровне товаров) и глобальных (на уровне кластеров) связей.

2. **Параметр clustering_beta β для управления «перепрыгиванием» товаров между кластерами.** При использовании алгоритма кластеризации (например, SWING или модифицированного K-means) параметр clustering_beta отвечает за вероятность изменения принадлежности товара i к кластеру K_k в ходе итеративного процесса. Формально, вероятность перехода товара i из текущего кластера K_k в другой кластер K_l может быть определена как:

$$P\left(K_l \mid i\right) = \frac{\exp\left(-\beta \cdot d(i, K_l)\right)}{\sum_m \exp\left(-\beta \cdot d(i, K_m)\right)}, \tag{3.7}$$

где $d(i, K_m)$ – мера расстояния между товаром i и центроидом кластера K_m, а β (значение clustering_beta) регулирует степень вероятности перепрыгивания (более высокие значения β делают распределение более уверенным).

3. **Количество итераций кластеризации num_trials.** Этот параметр определяет, сколько итераций выполняется для уточнения меток кластеров. Например, в алгоритме K-means итерации продолжаются до тех пор, пока изменение суммарного внутрикластерного расстояния:

$$W = \sum_k \sum_{i \in K_k} d(i, K_k),$$

становится меньше заданного порога. Однако num_trials задаёт жёсткое ограничение на максимальное количество таких итераций, обеспечивая баланс между точностью и временем выполнения.

4. **Коэффициент приоритета акционных товаров promo_weight.** Для акционных товаров вводится повышающий коэффициент

promo_weight, который увеличивает итоговый скоринг таких товаров:

$$\text{Score}_{\text{final}}(i) = \text{Score}(i) \cdot \text{promo\_weight}, \tag{3.8}$$

где promo_weight > 1. Этот параметр позволяет системе чаще предлагать акционные товары, что соответствует бизнес-целям (например, стимулирование продаж уценённых или новых товаров).

Для настройки вышеуказанных гиперпараметров использовались современные фреймворки оптимизации гиперпараметров. В частности, была задействована библиотека **Optuna**, которая предоставляет мощные инструменты для автоматизированного поиска оптимальных значений [2]. Основные возможности, использованные при настройке:

- **Стратегия TPE (Tree-structured Parzen Estimator).** Этот подход позволяет сужать пространство поиска, постепенно фокусируясь на наиболее перспективных областях. На каждой итерации библиотека оценивает результаты (целевую функцию) и обновляет вероятностное распределение гиперпараметров.
- **Байесовская оптимизация.** Используется для поиска глобального максимума целевой функции, основываясь на апостериорной вероятности. Байесовский подход позволяет эффективно исследовать пространство гиперпараметров даже при ограниченном числе итераций.
- **Автоматическое логирование.** Optuna автоматически фиксировала результаты каждой итерации, включая значения метрик (например, Recall@5, CTR), что упрощало анализ процесса оптимизации.
- **Параллельные вычисления.** Использование параллельного режима позволило запускать несколько конфигураций гиперпараметров одновременно, что существенно ускорило процесс поиска.

Целевой функцией для оптимизации выступала смешанная метрика, называемая weighted_rate. Она учитывала две ключевые составляющие:

$$\text{weighted\_rate} = \alpha \cdot \text{Mean\_Purchase\_Rate} + (1 - \alpha) \cdot \text{CTR}, \tag{3.9}$$

где α – весовой коэффициент, определяющий вклад каждой метрики в итоговую оценку.

Было проведено 60 итераций оптимизации, что заняло около 3 суток вычислений на сервере с параллельным запуском задач. График зависимости метрики от числа итераций представлен на рисунке 3.1.

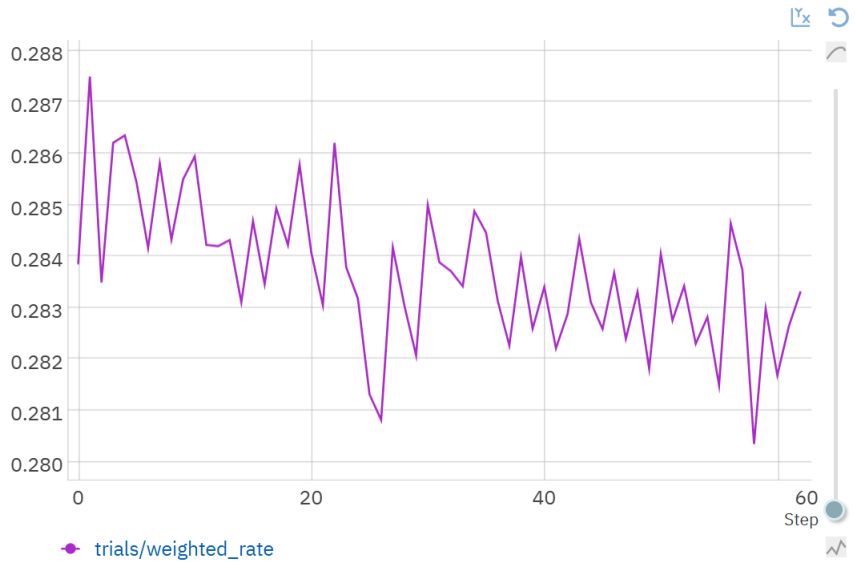

Рисунок 3.1 — График зависимости weighted_rate от числа итераций

В результате были найдены оптимальные значения гиперпараметров, которые обеспечили значительный прирост целевой метрики weighted_rate, что, в свою очередь, повысило качество рекомендаций.

Также после процедуры оптимизации была измерена важность гиперпараметров. График важность представлен на рисунке 3.2.

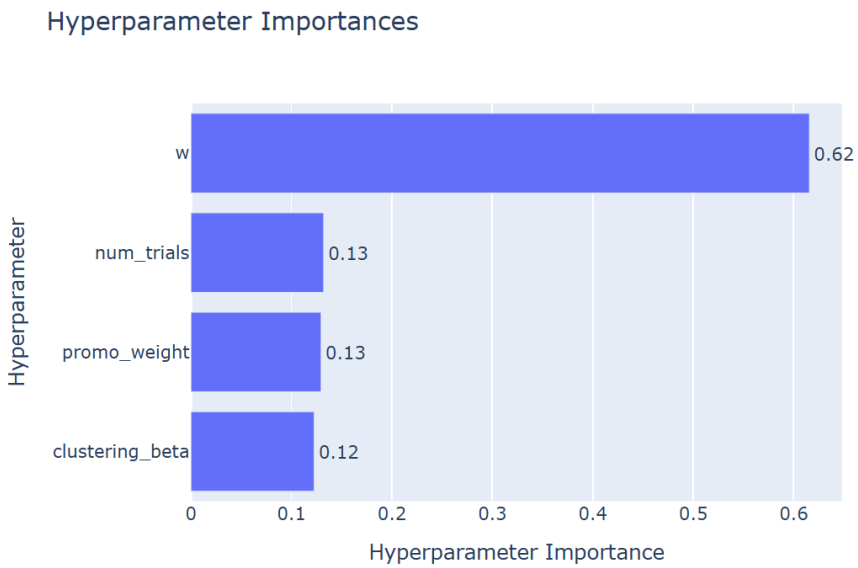

Рисунок 3.2 — График важности гиперпараметров

Таким образом, наиболее значимым оказался коэффициент баланса w. То есть он вносит значительный вклад в весь гибридный алгоритм.

3.4 Учет промо-товаров в рекомендациях

С точки зрения бизнеса, одним из ключевых аспектов рекомендательных систем является продвижение акционных, уценённых или сезонно-значимых товаров. Такие товары имеют стратегическое значение для увеличения продаж и привлечения внимания пользователей, однако в классических рекомендательных системах они могут «теряться» среди более популярных товаров. Чтобы решить эту проблему, в предложенном гибридном решении был введён специальный коэффициент promo_weight.

Для повышения приоритета акционных товаров применяется следующая формула окончательного скоринга:

$$\text{Score}_{\text{final}}(i) = \begin{cases} \text{Score}(i) \cdot \text{promo\_weight}, & \text{если товар } i \text{ относится к промо,} \\ \text{Score}(i), & \text{в противном случае.} \end{cases} \tag{3.10}$$

где:

- $\text{Score}(i)$ – базовый скоринг товара i, рассчитанный на основе гибридного подхода (учитывающий схожесть товаров и кластеров);
- promo_weight – коэффициент, увеличивающий приоритет промо-товара.

Важно отметить, что при выборе значения promo_weight необходимо учитывать баланс между продвижением акционных товаров и качеством рекомендаций. Если значение коэффициента будет слишком высоким, это может привести к следующим нежелательным эффектам:

- блок рекомендаций может быть засорён товарами, которые являются малопривлекательными для пользователей, но имеют низкую стоимость;
- пользователи могут получать слишком сильно перекошенный список рекомендаций, игнорирующий их реальные интересы.

Для предотвращения таких ситуаций использовались следующие подходы:

1. **Учет статуса промо-товара.** Если запасы товара на складе ограничены, алгоритм автоматически снижает его приоритет, чтобы избежать ситуации, когда пользователю предлагается товар, которого уже нет в

наличии. Для этого в расчёте скоринга используется корректирующий множитель γ, зависящий от остатка товара:

$$\gamma = \min\left(1, \frac{\text{Остаток товара}}{\text{Пороговое значение}}\right), \tag{3.11}$$

где Пороговое значение – минимальный желаемый запас товара для его активного продвижения.

2. **Тестирование различных значений promo_weight.** Оптимальные значения коэффициента были найдены с использованием фреймворка **Optuna**, что позволило учесть влияние промо-товаров на целевые метрики (например, CTR и Mean Purchase Rate).

3.5 Результаты оптимизации

После завершения цикла экспериментов и настройки гиперпараметров была проведена **финальная валидация** на реальных данных за март 2025 года (или другой согласованный временной интервал). Были получены следующие результаты:

1. **Рост Weighted Rate.** Составная метрика Weighted Rate (3.9), учитывающая качество рекомендаций и кликабельность, увеличилась на 1.5%–2% по сравнению с базовой моделью. Рост метрики свидетельствует об улучшении общего качества рекомендаций.

2. **Увеличение Mean Purchase Rate.** Вероятность покупки рекомендуемого товара также возросла, что указывает на лучшее соответствие предложений интересам пользователей. Это означает, что пользователи стали чаще добавлять в корзину товары из предложенного блока рекомендаций.

3. **Сохранение Coverage и Diversity.** Метрики Coverage и Diversity остались либо без изменений, либо незначительно улучшились. Это показывает, что система продолжает охватывать широкий пласт ассортимента и не «сужает» его только до популярных категорий.

Следовательно, рост целевых метрик, таких как Weighted Rate и Mean Purchase Rate, был достигнут без негативного влияния на Coverage и Diversity. Это означает, что:

- рекомендательная система стала точнее в подборе товаров, которые соответствуют интересам пользователей;
- улучшения не привели к ограничению ассортимента, что важно для сохранения разнообразия предложений;
- система осталась гибкой и способной учитывать бизнес-цели, такие как продвижение акционных товаров, без ухудшения пользовательского опыта.

Таким образом, предложенные улучшения показали высокий потенциал для повышения эффективности рекомендательной системы, сохраняя её способность учитывать интересы пользователей и бизнес-задачи.

3.6 Интерпретация метрик и аналитика полученных результатов

Для оценки эффективности внедрения нового алгоритма был проведён комплексный анализ ключевых метрик. Эти метрики анализировались как в офлайновом режиме (ретроспективные тесты на исторических данных), так и в онлайновом режиме (с использованием A/B-тестирования). Подробные результаты представлены ниже:

1. **Coverage**. Метрика охвата Coverage осталась на уровне примерно 13%. Это значение считается приемлемым, учитывая специфику корзины товаров в e-commerce. Многие товары не предполагают наличия сопутствующих рекомендаций (например, одноразовые товары или товары с ограниченным ассортиментом). Стабильность Coverage указывает на то, что новая система не сузила ассортимент рекомендаций, несмотря на внедрение дополнительных бизнес-ориентированных улучшений.
2. **CG@5 (Cumulative Gain at 5)**. Метрика CG@5 демонстрирует значительный рост при использовании нового алгоритма. Эта метрика измеряет общее качество рекомендаций на основе релевантности товаров, представленных в топ-5. Анализ показал, что пользователи, получавшие новые рекомендации, чаще кликали по предложенным товарам. Увеличение кликов (CTR) также сопровождалось ростом Mean Purchase Rate, так как часть пользователей совершала покупки из предложенных рекомендаций.

3. **Precision@5 и Recall@5**. Метрики точности (Precision@5) и полноты (Recall@5) также показали улучшения. Они были особенно заметны среди пользователей, которые имели хотя бы одну покупку в своей истории. Рост этих метрик указывает на повышенную релевантность рекомендаций для пользователей с историей покупок, что подтверждает адаптивность системы к персональным предпочтениям.

4. **Новизна (Novelty)**. Метрика новизны (Novelty) осталась на высоком уровне. Это обусловлено тем, что алгоритм учитывает множество категорий и кластеров, а не ограничивается популярными товарами. Сохранение высокой новизны свидетельствует о том, что система продолжает предоставлять разнообразные и нетривиальные рекомендации, что положительно сказывается на пользовательском опыте.

Для уверенности в том, что новый алгоритм адаптируется к реальному пользовательскому поведению, проводилось комплексное тестирование, включающее онлайновые тесты и офлайновые оценки.

Онлайновые тесты (A/B-тестирование). Пользователи были разделены на две группы: контрольную (старая система) и тестовую (новый алгоритм). Сравнение метрик (CTR, Mean Purchase Rate, Recall@5) показало явное преимущество новой системы.

Офлайновая оценка (ретроспективное тестирование). Использовались исторические данные покупок для моделирования работы алгоритма. Это позволило оценить поведение системы в условиях, близких к реальным.

Комплексный подход к тестированию подтвердил адаптивность системы как к историческим данным, так и к реальному пользовательскому поведению.

Таким образом, новая система рекомендаций продемонстрировала улучшение ключевых метрик качества, сохраняя способность предлагать разнообразные и релевантные товары.

ГЛАВА 4
ВНЕДРЕНИЕ И АБ-ТЕСТИРОВАНИЕ РАЗРАБОТАННЫХ РЕШЕНИЙ

4.1 Методология проведения A/B-тестирования

A/B-тестирование – это стандартный способ оценки влияния изменений в продукте путём сравнения двух групп пользователей: контрольной (A) и экспериментальной (B). Для рекомендательных систем A/B-тестирование особенно важно, поскольку метрики могут сильно зависеть от сезонности, разнообразия пользовательских сегментов и других факторов. Схема проведения A/B-тестирования приведена на рисунке 4.1.

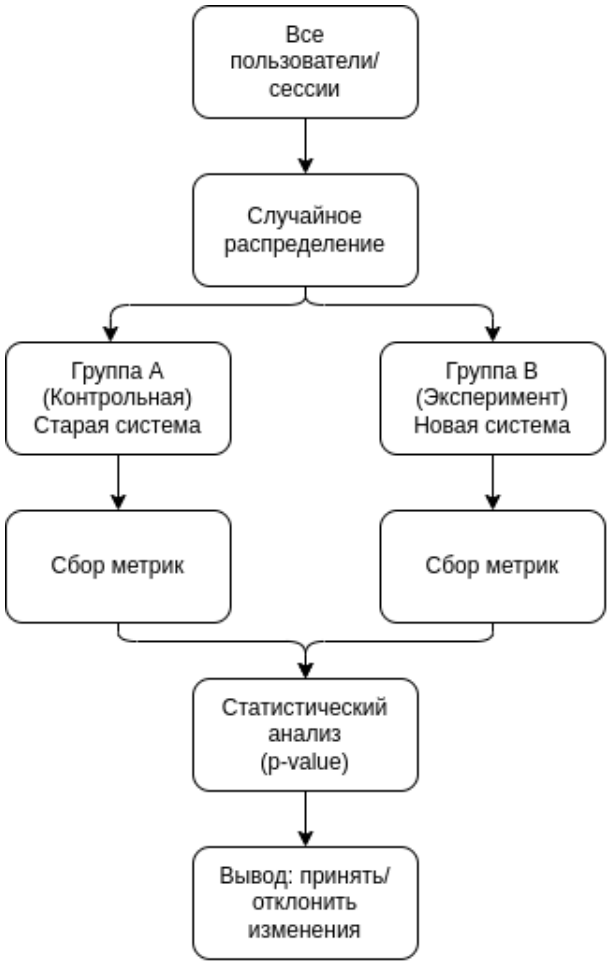

Рисунок 4.1 — Схема A/B-тестирования

Таким образом, основные принципы проведения A/B-тестирования:

1. **Случайное распределение пользователей в группы**: каждый но-

вый пользователь или сессия (если неавторизованный пользователь) с равной вероятностью попадал в A либо B.

2. **Независимость выборок**: группы не должны пересекаться, чтобы не было «утечек» и эффекта смешения.

3. **Достаточный объём и длительность теста**: чтобы обеспечить статистическую надёжность (p-value < 0.05 при заданной мощности теста), тест обычно проводят в течение нескольких недель (или до накопления требуемого числа взаимодействий).

В рамках A/B-теста новой рекомендательной системы были выбраны ключевые **метрики успеха**: CTR, доля пользователей, сделавших покупку из рекомендаций, средний чек, среднее количество позиций в корзине и др.

Однако при проведении A/B-тестирования возникала проблема, связанная с тем, что часть пользователей не авторизуется. Для решения этой проблемы приходилось также оперировать случайно генерируемым идентификатором anonymous_id. Для корректного деления приблизительно 50% неавторизованных пользователей попадали в группу A, а 50% – в группу B. Дополнительно был реализован механизм «смешения», чтобы не было смещения в сторону более частых или активных пользователей.

Благодаря такому подходу достигается **стабильность** распределения: один и тот же пользователь последовательно попадает в одну группу (A или B), пока не очистит cookie или не сменит устройство. Это позволяет собирать данные по взаимодействию пользователя с рекомендациями на протяжении всего цикла теста.

4.2 Архитектура передачи данных с использованием Airflow

Чтобы систематизировать процесс обновления рекомендаций, в компании использовали **Apache Airflow** — платформу для оркестрации задач. Ниже кратко описан **DAG (Directed Acyclic Graph)** для ежедневной генерации рекомендаций:

1. **Сбор данных**. Считывание логов из ClickHouse и Redis (информация о покупках, просмотрах, промо-статусах товаров).

2. **Подготовка фичей**. Агрегация показателей в таблицы вида (user, item, timestamps, actions); вспомогательные вычисления (категории, кластеры, промо).

3. **Baseline-группа (A)**. Запуск скрипта расчёта рекомендаций по старому алгоритму CF; загрузка результатов в Redis/базу под определённым ключом;

4. **B-группа**. Запуск скрипта гибридного алгоритма с учётом оптимизированных гиперпараметров; загрузка результатов в Redis/базу под другим ключом.

5. **Очистка и логирование**. Если нужно, выполняется удаление старых результатов и сбор статистики, которая потом анализируется аналитиками.

На рисунке 4.2 схематических представлен описанный DAG.

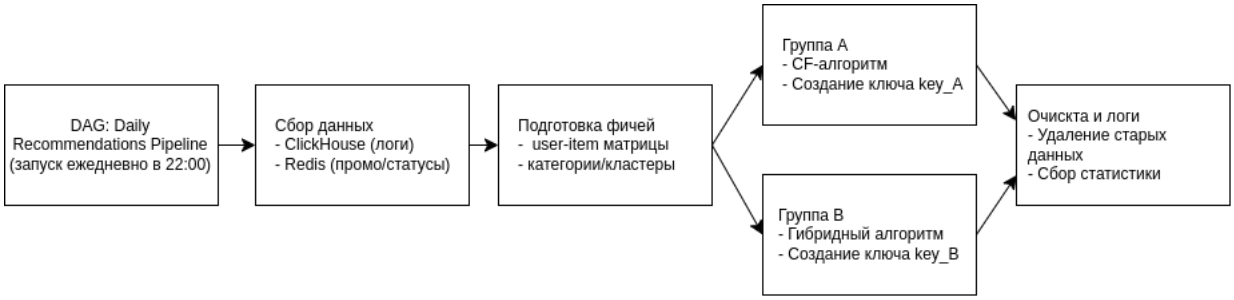

Рисунок 4.2 — Схема DAG для ежедневной генерации рекомендаций

Расписание DAG обычно ежедневное (например, в 22:00), чтобы по утрам пользователи уже получали актуальные рекомендации с учётом вчерашних покупок.

4.3 Статистические меры оценки результатов

Для анализа результатов A/B-теста были собраны и рассчитаны следующие ключевые метрики:

- **CTR (Click-Through Rate)** — показатель кликабельности, который вычисляется как доля кликов по рекомендуемым товарам среди общего числа показов блока. Формула для расчёта CTR:

$$\text{CTR} = \frac{\text{Количество кликов}}{\text{Количество показов}} \times 100\%. \tag{4.1}$$

- **CR (Conversion Rate)** — коэффициент конверсии в покупку, который отображает долю пользователей, совершивших покупку после перехода в блок рекомендаций. Формула:

$$\text{CR} = \frac{\text{Количество покупок}}{\text{Количество переходов}} \times 100\%. \tag{4.2}$$

- **Средний чек** — метрика, которая позволяет оценить, увеличивается ли общая сумма заказа при использовании новой системы рекомендаций. Формула:

$$\text{Средний чек} = \frac{\text{Общая сумма заказов}}{\text{Количество заказов}}, \tag{4.3}$$

  которая вычисляется именно за период использования новой системы рекомендаций.

- **Статистическая значимость** измерялась с использованием p-value, которое рассчитывалось через t-тест (для малых выборок) или z-тест (для больших выборок). Гипотеза проверялась следующим образом:

  – Нулевая гипотеза H_0: разницы между группами A и B нет.

  – Альтернативная гипотеза H_1: новая система рекомендаций (группа B) приводит к улучшению метрик.

  Уровень значимости α был установлен на уровне 0.01. Если p-value $< \alpha$, то нулевая гипотеза отклонялась.

Собранные данные за тестовый период показали, что новая модель рекомендаций даёт статистически значимый прирост ключевых метрик.

4.4 Анализ результатов тестирования

Завершающим этапом A/B-эксперимента является аналитический обзор, направленный на ответы на следующие вопросы:

- Превзошёл ли новый алгоритм старый по выбранным метрикам?
- Насколько велика разница?
- Является ли эта разница статистически значимой?

Результаты тестирования можно обобщить следующим образом. За период теста (3–4 недели) группа B показала **рост CTR на 10–15%** по сравнению с группой A. При этом p-value оказалось меньше 0.01:

$$\begin{aligned} H_0 &: \mu_{\text{CTR, A}} = \mu_{\text{CTR, B}}, \\ H_1 &: \mu_{\text{CTR, A}} < \mu_{\text{CTR, B}}, \\ &p\text{-value} < 0.01. \end{aligned} \tag{4.4}$$

Это свидетельствует о высоком уровне статистической значимости полученных результатов.

Наблюдается увеличение доли покупок. То есть, если система рекомендаций предлагает более релевантные товары, то пользователи чаще добавляют их в корзину. Для группы B фиксировался **прирост CR в среднем на 2–3%**:

$$\Delta CR = \text{CR}_\text{B} - \text{CR}_\text{A}, \tag{4.5}$$

где $\Delta CR > 0$ для большинства категорий товаров. Это улучшение подтверждает эффективность новой системы рекомендаций.

Рост среднего чека и количества позиций в заказе указывает на увеличение общего объёма продаж. **Средний чек в группе B увеличился на** 5%:

$$\Delta\text{Средний чек} = \text{Средний чек}_\text{B} - \text{Средний чек}_\text{A}. \tag{4.6}$$

Это косвенно подтверждает экономическую целесообразность внедрения новой системы и обоснованность её использования для всех пользователей.

Промо-товары получали больший приоритет, что повышало их кликабельность. Однако необходимо учитывать, что это может негативно сказаться на удовлетворённости пользователей, если акционные товары оказываются низкого качества или быстро заканчиваются.

Результаты могли различаться для разных сегментов пользователей, таких как новички и постоянные покупатели. Это требует проведения дополнительных разрезов данных для более точного анализа.

На основе полученных результатов руководство компании приняло решение о полном развертывании новой системы рекомендаций для всей пользовательской базы. Если возникала необходимость уточнения деталей алгоритма или тестирования новых параметров, тестирование продолжалось с учётом выявленных наблюдений.

ЗАКЛЮЧЕНИЕ

Данная работа продемонстрировала, как совокупность современнейших методов машинного обучения — графовые рекомендации, кластеризация, активное обучение и байесова оптимизация гиперпараметров — способна решить практически значимую бизнес-задачу «next-basket» рекомендаций в условиях реального высоконагруженного маркетплейса В ходе написания диплом была успешно решена серия задач, связанных с улучшением рекомендательных систем:

1. Оптимизирован механизм категоризации товаров с применением активного обучения (Uncertainty и Diversity Sampling). Это позволило существенно сократить долю «Прочее» и повысить точность классификации.

2. Разработана гибридная модель рекомендаций для корзины, сочетающая товарную схожесть, кластеризацию и учёт промо-позиций.

3. Проведена оптимизация гиперпараметров (с помощью Optuna), что дало ощутимый прирост метрик (Mean Purchase Rate, CTR и т.д.).

4. Реализовано A/B-тестирование нового решения и подтверждён положительный эффект для бизнеса.

Оценка значимости выполненной работы следующая:

1. **Рост конверсии**: более релевантные рекомендации способствуют тому, что пользователи охотнее «докладывают» товары в корзину.

2. **Экономия ресурсов**: за счёт использования Active Learning команда разметчиков тратит меньше времени на рутинную маркировку, акцентируясь на наиболее ценных для модели примерах.

3. **Гибкая архитектура**: модульная структура алгоритмов и DAG в Airflow упрощает дальнейшее масштабирование и адаптацию под новые задачи.

Перспективы дальнейшего развития проектов:

1. **Sequence-to-sequence NBR.** Модели типа BERT4Rec и Binary Transformer могут заменить статический граф, захватывая долгосрочные паттерны, cold-start и контекст (цена, сезон).

2. **Контекстуальные бандиты.** Реал-тайм policy-gradient позволит балансировать exploration-exploitation и учитывать онлайн сигнал о неявном неудовлетворении.

3. **Continual Learning Pipeline.** С переходом к streaming-данным (Kafka $\rightarrow$ MLflow $\rightarrow$ ClickHouse) возможно переобучение каждый час без полного даунтайма.

4. **Explainable AI**. Визуализация внимания графа или Grad-CAM для seq-моделей повысит доверие бизнес-стейкхолдеров.

Таким образом, представленная работа иллюстрирует, что соединение строгой теории (кластеризация, статистические тесты) и индустриальной практики (Airflow-DAG, Redis-кэш, Optuna + Neptune MLOps) способно увеличить ценность цифрового продукта в пределах месяцев, а не лет. В то время как матричные рекомендации стали commodities, именно грамотная инженерная интеграция, метрически обоснованные A/B-эксперименты и непрерывная оптимизация превращают R&D-прототип в масштабируемый двигатель роста. Иными словами, алгоритмическая изощрённость без операционной зрелости остаётся искусством ради искусства; мой проект демонстрирует, что синергия обеих сторон способна приносить измеримый вклад в виртуозность крупных e-commerce-платформ

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Воронцов, К. В. Алгоритмы кластеризации и многомерного шкалирования. Курс лекций / К. В. Воронцов. – М. : МГУ, 2007. – 352 с.

2. Документация по библиотеке Optuna [Электронный ресурс]. – Режим доступа: `https://optuna.org/`. – Дата доступа: 05.04.2025.

3. Документация по Airflow [Электронный ресурс]. – Режим доступа: `https://airflow.apache.org/`. – Дата доступа: 05.04.2025.

4. Дьяконов, В. П. Энциклопедия компьютерной алгебры / В. П. Дьяконов. – М. : ДМК Пресс, 2009. – 1264 с.

5. Таранчук, В.Б. Методы и средства системы ГеоБазаДанных для адаптации компьютерных моделей. Инструменты кластеризации / В.Б. Таранчук // Информационные технологии и системы 2021 (ИТС 2021) : материалы междунар. науч. конф. (Республика Беларусь, Минск, 24 ноября 2021 года) / Минск: БГУИР, 2021. – С. 164 – 165

6. Оптимизация гиперпараметров в машинном обучении [Электронный ресурс]. – Режим доступа: `https://habr.com/ru/company/ods/blog/563578/`. – Дата доступа: 31.03.2025.

7. Aggarwal, C. C. Recommender Systems. / C. C. Aggarwal. – Springer, 2016. – 471 с.

8. Avazpour, I. Dimensions and Metrics for Evaluating Recommendation Systems / I. Avazpour, T. Pitakrat, J. Grunske. – Recommendation Systems in Software Engineering, 2014. – 29 с.

9. Biau, G. Lectures on the Nearest Neighbor Method / G. Biau, L. Devroye. – Springer, 2015. – 124 с.

10. Bishop, C. Pattern Recognition and Machine Learning / C. Bishop. – Springer, 2006. – 738 с.

11. Burke, R. Hybrid Recommender Systems: Survey and Experiments / R. Burke, X. Chen, Y. Yu. – User Modeling and User-Adapted Interaction, 2002. – 331-370 с.

12. Feurer, M. Hyperparameter Optimization / M. Feurer, F. Hutter, J. Li. – Automated Machine Learning: Methods, Systems, Challenges, 2019. – 3-33 с.

13. Gong, S. A Collaborative Filtering Recommendation Algorithm Based on User Clustering and Item Clustering / S. Gong. – China : Zhejiang Business Technology Institute, Ningbo 315012, China, 2010. – 8 с.

14. He, X. Neural Collaborative Filtering / X. He, L. Liao, H. Zhang. – : Proceedings of the 26th International Conference on World Wide Web (WWW), 2017. – 173-182 с.

15. Hierarchical clustering. [Электронный ресурс] – https://en.wikipedia.org/wiki/Hierarchical_clustering – Дата доступа 31.03.2025.

16. Jadon, A. A Comprehensive Survey of Evaluation Techniques for Recommendation Systems / A. Jadon, A. Patil, J. Grunske. – Juniper Networks, Sunnyvale CA, USA, 2024. – 10 с.

17. k-means clustering. [Electronic resource] – https://en.wikipedia.org/wiki/K-means_clustering – Access date 31.03.2025.

18. Kang, W. C. Self-Attentive Sequential Recommendation / W. C. Kang, J. McAuley, H. Zhang. – Proceedings of the IEEE International Conference on Data Mining (ICDM), 2018. – 197-206 с.

19. Koren, Y. Matrix Factorization Techniques for Recommender Systems. / Y. Koren, R. Bell, C. Volinsky. – Computer, 2009. – 30-37 с.

20. PyClustering Documentation. [Электронный ресурс]. – https://pyclustering.github.io/ – Дата доступа 31.03.2025.

21. Ricci, F. Recommender Systems Handbook. / F. Ricci, L. Rokach, B. Shapira. – Springer, 2015. – 1003 с.

22. Settles, B. Active Learning Literature Survey. / B. Settles, J. McAuley, H. Zhang. – University of Wisconsin-Madison, Computer Sciences Technical Report, 2018.

23. Taranchuk, V. B. Interactive and intelligent tools of the GeoBazaDannych system / Valery B. Taranchuk // Open Semantic Technologies for Intelligent Systems: Research Papers Collection. – 2021. – Issue 5. – P. 245-250. (ISSN 2415-7740 (Print) ISSN 2415-7074 (Online))

24. What is the significance of clustering in recommender systems? [Electronic resource] – https://milvus.io/ai-quick-reference/what-is-the-significance-of-clustering-in-recommender-systems – Access date 31.03.2025.

25. Zhang, Y. Sequential Deep Learning for Recommender Systems. / Y. Zhang, X. Chen, Y. Yu. – : Frontiers of Computer Science, 2019. – 755-772 с.

26. Zhou, L. Bandit Algorithms for Recommendation Systems: A Survey and Empirical Comparison / L. Zhou, D. Chen, J. Li. – ACM Transactions on Intelligent Systems and Technology, 2019. – 1-24 с.