ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ИНСТИТУТ МЕЖДУНАРОДНЫХ ОТНОШЕНИЙ (УНИВЕРСИТЕТ)» МИНИСТЕРСТВА ИНОСТРАННЫХ ДЕЛ РОССИЙСКОЙ ФЕДЕРАЦИИ
ОДИНЦОВСКИЙ ФИЛИАЛ

ФАКУЛЬТЕТ ФИНАНСОВОЙ ЭКОНОМИКИ
КАФЕДРА МАТЕМАТИЧЕСКИХ МЕТОДОВ И БИЗНЕС-ИНФОРМАТИКИ

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

по направлению подготовки

38.03.05 Бизнес-информатика

Направленность (профиль) подготовки

Информационные технологии в международном бизнесе

Тема работы:

Разработка рекомендательной системы для интернет-магазина на основе кластеризации пользователей и ассоциативных правил

Выполнил:
студент Фризен Даниил Олегович
факультет ФЭ, гр. ИТБ(б)-О-21/1

_____
(подпись студента)

Научный руководитель:
Ерохин Виктор Викторович
д-р техн. наук, доцент

_____
(подпись научного руководителя)

Выпускная квалификационная работа защищена

«___» _____2025 г.

Оценка _____

Секретарь ГЭК _____

Одинцово 2025

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

В последние годы наблюдается значительный рост объемов онлайн-торговли и стремительное расширение ассортимента товаров, предлагаемых интернет-магазинами. В таких условиях покупатель сталкивается с огромным разнообразием предложений, что существенно затрудняет процесс выбора. Согласно исследованиям консалтинговых компаний McKinsey и Deloitte, персонализация предложений становится критически важным фактором успеха для интернет-магазинов, поскольку она позволяет значительно повысить коэффициент конверсии (Conversion Rate, CR) и пожизненную ценность клиента (Customer Lifetime Value, LTV).

Однако простые универсальные рекомендации, основанные лишь на популярности товаров, постепенно теряют свою эффективность в условиях высокой конкуренции и индивидуальных предпочтений потребителей. Покупатели ожидают более точных и релевантных рекомендаций, учитывающих их личные интересы, историю покупок и модели поведения. В связи с этим возникает необходимость применения подходов, сочетающих в себе глубокий анализ покупательских привычек (сегментацию аудитории) и выявление устойчивых ассоциативных связей между товарами.

В рамках данного исследования предлагается использовать сочетание поведенческой сегментации покупателей и алгоритма Apriori, который выявляет часто встречающиеся комбинации товаров в потребительских корзинах. Подобный подход позволяет формировать более точные и персонализированные рекомендации, тем самым обеспечивая рост ключевых показателей бизнеса (CR и LTV).

За последние пять лет российский e-commerce сменил парадигму: объём онлайн-заказов растёт быстрее, чем физический спрос, потому что каталоги маркетплейсов расширяются буквально «в геометрической прогрессии». Так, по оценке *Data Insight* и РБК, к лету 2023 г. на два крупнейших игрока — Ozon

и Wildberries — приходилось уже примерно 77 % всех интернет-покупок в стране (для сравнения: годом ранее их доля была 67 %). Ассортимент при этом продолжает раздуваться: только у Wildberries на складах находится свыше 500 млн SKU (уникальных товарных позиций).

Бурное «обесценивание» полки порождает информационную перегрузку: покупателю трудно «выудить» нужный товар из миллионов карточек. Статистика глобальных исследований это подтверждает: компании-лидеры, освоившие персонализацию, получают 5-15 % прироста выручки и 10-30 % экономии маркетинг-бюджета, а 80 % покупателей готовы тратить до 50 % больше, если видят персональные предложения. На практике Ozon и Wildberries уже строят выдачу вокруг сложных моделей рекомендаций на базе больших языковых и графовых сетей; собственные оценки площадок показывают двузначный рост конверсии (CR) и среднего чека (AOV) по сравнению с «универсальными» списками. Иными словами, масштаб каталога без персонализации превращается из конкурентного преимущества в барьер продаж.

Пояснение терминов:

- CR (Conversion Rate) — доля сессий, закончившихся покупкой.
- LTV (Customer Lifetime Value) — совокупная прибыль от клиента за весь период взаимодействия.

Целью данной выпускной квалификационной работы является разработка и апробация прототипа гибридной рекомендательной системы на основе сегментации пользователей и алгоритма ассоциативных правил Apriori, а также проверка гипотезы о том, что предложенный подход обеспечивает более высокую эффективность рекомендаций по сравнению с традиционными методами.

Для достижения поставленной цели сформулированы следующие задачи исследования:

1. Изучить современные подходы к формированию персонализированных рекомендаций в сфере электронной коммерции (e-commerce), а также выявить их преимущества и недостатки.
2. Выполнить сегментацию пользователей интернет-магазина на основе их поведенческих признаков с использованием метода K-Means и RFM-анализа.
3. Реализовать генерацию рекомендаций с использованием алгоритма ассоциативных правил Apriori как в общем случае, так и отдельно для каждого выделенного сегмента.
4. Сравнить эффективность рекомендаций, полученных с учетом сегментации и без неё, на основе ключевых метрик качества.
5. Разработать прототип интерфейса рекомендательной системы, демонстрирующий персонализированные рекомендации и причины их возникновения.
6. Выполнить экономический расчет потенциального эффекта от внедрения разработанной рекомендательной системы и предложить практический план её интеграции в реальный бизнес-процесс интернет-магазина.

Объектом данного исследования является процесс формирования и предоставления товарных рекомендаций покупателям интернет-магазинов.

Предметом исследования выступают методы сегментации пользователей по поведенческим признакам (RFM-анализ и K-Means-кластеризация) и алгоритм выявления товарных ассоциаций (Apriori), которые применяются для построения персонализированных рекомендаций.

Под RFM-анализом (от англ. Recency, Frequency, Monetary) понимается методика оценки поведения клиентов, которая учитывает три основных параметра: как давно была совершена последняя покупка (Recency), насколько

часто совершаются покупки (Frequency) и на какую сумму покупает клиент (Monetary). Под K-Means-кластеризацией понимается алгоритм машинного обучения, который позволяет автоматически разбить множество объектов (в данном случае — покупателей) на несколько однородных групп (кластеров) по их признакам. Под алгоритмом Apriori понимается метод анализа корзин покупок, позволяющий находить часто встречающиеся комбинации товаров и строить на их основе ассоциативные правила вида «если куплен товар A, то с высокой вероятностью будет куплен товар B».

Для реализации поставленных задач использован комплекс современных методов анализа данных и машинного обучения, включающий:

- Анализ научной и отраслевой литературы для выявления актуальных подходов и методов формирования рекомендаций в электронной коммерции.
- RFM-анализ и кластеризацию методом K-Means для сегментации покупателей интернет-магазина на основании их истории взаимодействий и покупательского поведения.
- Алгоритм Apriori (с применением библиотеки mlxtend в Python) для выявления устойчивых ассоциаций между товарами и генерации товарных рекомендаций.
- Метрики оценки качества рекомендаций (Precision@5, Recall@5, Coverage), позволяющие количественно оценить точность и полноту рекомендаций.
- Статистические методы проверки гипотез (тестирование на отложенной выборке и X^2-тест значимости) для подтверждения достоверности полученных результатов.
- Методы визуализации данных (PCA и UMAP) для наглядного представления и анализа полученных кластеров и рекомендаций.

- Экономический анализ ROI (Return on Investment) и расчет экономической целесообразности внедрения системы.
- Инструменты реализации: язык программирования Python (Pandas, mlxtend, scikit-learn) и платформа разработки интерфейсов Streamlit.

Краткие определения

- RFM-анализ — метод поведенческой сегментации, где R — давность последней покупки, F — частота заказов, M — суммарные траты.
- K-Means — алгоритм, который «тянет» объекты к ближайшим центроидам по метрике Евклида, формируя K кластеров.
- PCA (Principal Component Analysis) снижает размерность, позволяя нарисовать облака точек в 2D и увидеть «устойчивость» сегментов.
- ROI — коэффициент окупаемости инвестиций; если он > 0 — проект приносит прибыль.

Практическая значимость работы заключается в разработке полностью готового к внедрению прототипа рекомендательной системы, которая может быть интегрирована в интернет-магазины малого и среднего бизнеса без необходимости значительных инвестиций в инфраструктуру и техническое сопровождение.

Использование предложенного в работе гибридного подхода (сегментация + алгоритм Apriori) позволит интернет-магазинам повысить точность персонализированных рекомендаций, что ведёт к увеличению среднего размера покупки (среднего чека), частоты повторных покупок, росту конверсии посетителей сайта в покупателей и, как следствие, увеличению общей прибыли бизнеса. Разработанный прототип демонстрирует рекомендации и причину их появления, что способствует росту доверия клиентов к рекомендательной системе и повышению удовлетворенности пользователей.

Кроме того, в работе представлен подробный экономический расчет потенциального эффекта от внедрения системы, а также предложен пошаговый план её интеграции в реальный бизнес-процесс интернет-магазина, что упрощает процесс принятия решения руководством компании о практическом внедрении разработанного решения.

Таким образом, предложенная тема исследования является актуальной как с научной, так и с практической точки зрения, а разработанные в ходе исследования подходы и решения могут быть успешно применены в сфере электронной коммерции для повышения эффективности взаимодействия с клиентами и роста ключевых бизнес-показателей интернет-магазина.

ГЛАВА 1. РЕКОМЕНДАТЕЛЬНЫЕ СИСТЕМЫ В E-COMMERCE: ТЕОРИЯ И БИЗНЕС-КОНТЕКСТ

1.1 Персонализация как драйвер CR и LTV

Conversion Rate (CR) — это доля пользовательских сессий, завершившихся целевым действием (обычно покупкой). Customer Lifetime Value (LTV) — кумулятивная прибыль, которую приносит клиент за всё время взаимодействия с магазином. Эти два показателя непосредственно реагируют на то, насколько «умно» продавец обращается с вниманием покупателя.

Исследование McKinsey показывает, что персонализированный опыт способен сократить стоимость привлечения клиента до 50 %, увеличить выручку на 5-15 % и поднять marketing ROI на 10-30 %. Более того, компании-лидеры получают до 40 % своего роста выручки именно благодаря персонализации. Ожидания покупателей поднимаются синхронно: 71 % респондентов ждут персональных обращений, а 76 % раздражаются, если их нет.

Потребители вознаграждают заботу рублём: исследование Epsilon зафиксировало, что 80 % покупателей с большей вероятностью завершат покупку, когда видят персональные предложения. Отдельный эффект дают cross-sell и up-sell-механики (см. пояснение ниже): аналитики McKinsey оценили, что грамотный cross-sell повышает продажи на 20 % и прибыль на 30 %.

Cross-sell — предложение сопутствующих товаров (к ноутбуку мышь).

Up-sell — предложение более дорогой версии того же товара (SSD 512 ГБ вместо 256 ГБ).

Trust signals — элементы, повышающие доверие (рейтинги, отзывы, «другие купили»).

Таблица 1 - Функции RecSys

Функция рекомендательной системы	Пользовательская ценность	Бизнес-результат
Cross-sell	Дополняет основной товар	↑AOV, рост маржи
Up-sell	Помогает выбрать «оптимальный» вариант	↑Средний чек
Открытие (discovery)	Показывает неожиданные релевантные товары	↑CR, лояльность
Доверие	Подсказывает, что «товар мне подходит»	↓Отказы, ↑LTV

Таким образом, персональные рекомендации работают как рычаг двойного действия: повышают краткосрочную конверсию и закладывают фундамент для долгосрочной ценности клиента.

1.2 Классификация рекомендательных систем и место алгоритма Apriori

Рекомендательные системы (Recommendation Systems, RS) в электронной коммерции представляют собой программные решения, которые автоматически предлагают пользователю товары или услуги, соответствующие его интересам и поведению. Развитие рекомендательных технологий стало необходимым ответом на проблему информационной перегрузки, когда традиционные методы поиска товаров становятся недостаточными для эффективной навигации по ассортименту интернет-магазинов.

Существует несколько основных подходов к построению рекомендательных систем, каждый из которых имеет свои особенности, преимущества и ограничения. Наиболее распространённая классификация включает три больших класса:

1. Коллаборативные рекомендательные системы (Collaborative Filtering). В основе таких систем лежит анализ взаимодействий пользователей с товарами без использования их содержательных характеристик. Алгоритмы определяют похожесть между пользователями (user-based) или между товарами (item-based) на основе их историй взаимодействия (например, оценок, покупок или просмотров). Рекомендация формируется на основании действий «похожих» пользователей. Пример: если пользователи, схожие с данным, покупали товар X, система предложит X новому пользователю.

2. Контентно-ориентированные рекомендательные системы (Content-Based Filtering). Эти системы анализируют характеристики товаров (например, категорию, цену, бренд, описание) и предпочтения пользователя относительно этих характеристик. Рекомендации формируются на основе схожести новых товаров с теми, что ранее были оценены пользователем положительно. Пример: если пользователь покупал органические продукты, система будет рекомендовать аналогичные товары по их описанию.

3. Гибридные рекомендательные системы (Hybrid Systems). Такие системы комбинируют методы коллаборативной фильтрации и контентной фильтрации для получения более точных рекомендаций. Комбинирование позволяет компенсировать недостатки каждого из подходов и повышает устойчивость системы к проблемам, таким как "холодный старт" новых пользователей или товаров.

На фоне данных подходов выделяется отдельная методика, широко применяемая в задачах анализа покупательского поведения, — анализ корзины покупок (Market Basket Analysis, MBA). В отличие от классических рекомендаций, MBA фокусируется не на индивидуальных предпочтениях пользователей, а на выявлении закономерностей между товарами в совокупности покупок различных пользователей. Целью является

обнаружение устойчивых связей между товарами, которые часто приобретаются вместе.

Классическим инструментом для реализации анализа корзины является алгоритм Apriori. Он позволяет автоматически выявлять частые наборы товаров и строить на их основе правила ассоциаций вида «Если покупатель приобрёл товар A, то с высокой вероятностью он также купит товар B». Это даёт возможность использовать Apriori как самостоятельный метод построения рекомендаций в интернет-магазинах, особенно в сценариях кросс-продаж (cross-sell) и увеличения среднего чека.

Таким образом, алгоритм Apriori занимает уникальное место в общей классификации рекомендательных систем, действуя на стыке контентно-ориентированного анализа и поведенческой аналитики без привязки к конкретным пользователям, а к их совместным покупательским паттернам. Его особенности делают его особенно ценным инструментом в задачах анализа крупных транзакционных данных в электронной коммерции.

1.3 Алгоритм Apriori и анализ товарных ассоциаций

Алгоритм Apriori представляет собой один из первых и наиболее известных методов поиска ассоциативных правил в наборах транзакционных данных. Он был предложен Агравалом и Шрикантом в 1994 году для решения задачи поиска частых наборов товаров в больших базах данных продаж.

Основная идея алгоритма заключается в том, что если какой-либо набор товаров является частым (встречается в большом числе заказов), то все его подмножества также должны быть частыми. Это утверждение называется априорным свойством частых множеств и позволяет значительно сократить число проверяемых комбинаций товаров.

Ключевые понятия алгоритма Apriori:

- Support (поддержка) — это доля всех заказов, в которых присутствует определённый набор товаров. Например, если товарная пара (молоко, хлеб) встречается в 200 из 10 000 заказов, её поддержка равна 2 %.

  Формально:

$$Support(A \cup B) = \frac{\text{Число товаров с заказами } A \text{ и } B}{\text{Общее число заказов}}$$

- Confidence (доверие) — это вероятность того, что если товар A был куплен, то товар B также будет куплен. Confidence измеряет силу правила ассоциации.

  Формально:

$$Confidence(A \rightarrow B) = \frac{Support(A \cup B)}{Support(A)}$$

- Lift (подъём) — это мера зависимости между товарами. Lift сравнивает фактическую вероятность совместной покупки товаров A и B с вероятностью их независимого появления в заказах. Lift > 1 означает положительную зависимость между товарами.

  Формально:

$$Lift(A \rightarrow B) = \frac{Confidence(A \rightarrow B)}{Support(B)}$$

  Принцип работы алгоритма Apriori:

1. На первом этапе алгоритм определяет частые одиночные товары (1-itemsets), удовлетворяющие минимальному порогу поддержки.
2. Затем строятся частые пары товаров (2-itemsets) на основе одиночных товаров, затем тройки товаров (3-itemsets) и так далее, пока не перестанут находиться новые частые наборы.

3. Из найденных частых наборов формируются ассоциативные правила с учётом минимального уровня доверия и подъёма.

Пример работы алгоритма:

Таблица 2 - Примеры заказов

Заказ №	Купленные товары
1	Молоко, Хлеб
2	Молоко, Печенье
3	Молоко, Хлеб, Печенье
4	Хлеб, Печенье

На этом наборе данных алгоритм Apriori найдет, например, следующее правило:

- Если Молоко, то Хлеб
- Support = 2/4 = 0.5
- Confidence = 2/3, что примерно 0.67
- Lift > 1 (зависит от общей поддержки хлеба)

Преимущества алгоритма Apriori:

- Простота концепции и реализации.
- Эффективная обработка больших объемов транзакционных данных за счёт использования априорного свойства.

Ограничения алгоритма Apriori:

- Экспоненциальный рост числа проверяемых наборов товаров при снижении порога поддержки.
- Высокие требования к вычислительным ресурсам на больших наборах данных без предварительной фильтрации.

Для оптимизации работы Apriori в практике часто используют:

- Сужение анализируемого пространства товаров (например, по категориям или по частоте покупок).
- Параллельную обработку данных.
- Модификации алгоритма (например, алгоритмы Eclat или FP-Growth).

В задачах электронной коммерции алгоритм Apriori находит широкое применение в построении систем кросс-продаж, увеличении среднего чека и формировании персонализированных рекомендаций, ориентированных на сочетание товаров в покупательских корзинах.

1.4 Поведенческая сегментация пользователей (RFM-анализ, K-Means)

Для повышения эффективности персонализированных рекомендаций в электронной коммерции важнейшим этапом является сегментация пользователей. Сегментация позволяет разбить всю клиентскую базу на группы, участники которых обладают схожими характеристиками и поведением, что даёт возможность более точно настраивать рекомендации и маркетинговые активности для каждой группы.

Чтобы рекомендации не терялись в «средней температуре» по больнице, полезно сначала разбить аудиторию на однородные группы. На практике для e-commerce отлично зарекомендовали себя три простых поведенческих признака, объединённые в метод RFM:

Таблица 3 - Обзор RFM метода

Буква	Что измеряет	Как трактовать
Recency	Сколько дней прошло с последней покупки	Чем меньше число, тем пользователь «теплее»
Frequency	Сколько заказов сделал за период	Показатель лояльности

Monetary	Сколько денег потратил за период	Финансовая ценность клиента

RFM-анализ позволяет выявить, например, клиентов, которые недавно совершали покупки и часто их совершают (лояльные покупатели), а также клиентов, которые давно не проявляли активности (клиенты с риском ухода).

Однако для более тонкой группировки клиентов по RFM-признакам и другим характеристикам на практике применяется метод кластеризации K-Means.

Следующий шаг — кластеризация этих трёх (или расширенных) признаков. K-Means (кластеризация методом "k-средних") — это один из самых популярных алгоритмов машинного обучения без учителя. Он используется для разбиения набора объектов на k непересекающихся кластеров. Основная идея алгоритма заключается в том, чтобы минимизировать внутрикластерную дисперсию — то есть сгруппировать объекты так, чтобы расстояние между ними и центроидом их кластера было минимальным.

Принцип работы алгоритма K-Means включает следующие этапы:

1. Выбирается количество кластеров *k* (например, 4 или 5).
2. Случайным образом выбираются начальные центроиды кластеров.
3. Каждый объект (в нашем случае — пользователь) относится к ближайшему центроиду на основании выбранной метрики расстояния (обычно евклидова метрика).
4. После распределения объектов пересчитываются новые центроиды кластеров.
5. Процесс повторяется до тех пор, пока центроиды не перестанут существенно изменяться.

Преимущества K-Means заключаются в его простоте, интерпретируемости результатов и высокой скорости работы даже на больших выборках. Однако следует учитывать, что алгоритм чувствителен к масштабу данных, поэтому перед кластеризацией признаки обычно нормализуются (приводятся к единому масштабу, например, с помощью Min-Max Scaling или Standard Scaling).

Преимущества данного метода:

1. Скорость: алгоритм итеративно «притягивает» точки к ближайшим центрам; его временная сложность O(N·K·T) (N — число клиентов, K — число кластеров, T — итерации) линейна, а значит подходит даже для миллиона пользователей.
2. Простота настройки: единственный гиперпараметр — число кластеров; его легко подобрать графиком Elbow или индексом Silhouette.
3. Интерпретируемость: координаты центроида прямо показывают «среднего» представителя сегмента, что упрощает разработку маркет-стратегий. Обзор статей по клиентской сегментации фиксирует, что K-Means применяется почти в 40 % случаев, опережая более сложные методы.

Таким образом, комбинация RFM-анализа для создания осмысленных признаков и кластеризации методом K-Means для группировки пользователей является мощным инструментом в построении персонализированных рекомендательных систем. Она позволяет различать пользователей по их активности, ценности и частоте покупок, что существенно увеличивает эффективность дальнейших товарных рекомендаций.

1.5 Метрики оценки качества рекомендаций

Оценка качества работы рекомендательных систем является критически важной задачей. Без количественной оценки невозможно понять, насколько

эффективно предложенные рекомендации удовлетворяют интересы пользователей и способствуют достижению бизнес-целей.

Среди множества существующих метрик наиболее часто в электронной коммерции применяются Precision@k, Recall@k и Coverage. Ниже приведено подробное описание каждой из этих метрик.

Precision@k (точность на первых k рекомендациях) измеряет долю правильных рекомендаций среди первых k предложенных пользователю товаров. Это одна из основных метрик в задачах, где важна релевантность самых первых предложений.

Формально:

$$Precision@k = \frac{\text{Кол. релевантных товаров среди первых } k \text{ рекомендаций}}{k}$$

Например, если системе требуется показать пользователю 5 рекомендаций, и 3 из них действительно были бы им выбраны или куплены, Precision@5 составит 60 %.

Recall@k (полнота на первых k рекомендациях) измеряет долю релевантных товаров, которые были правильно предложены среди всех релевантных товаров для данного пользователя.

Формально:

$$Recall@k = \frac{\text{Кол. релевантных товаров среди первых } k \text{ рекомендаций}}{\text{Общее количество релевантных товаров}}$$

Эта метрика особенно важна в сценариях, где необходимо покрыть как можно больше интересов пользователя.

Coverage (покрытие) показывает долю всех товаров каталога, которые когда-либо рекомендованы хотя бы одному пользователю. Эта метрика позволяет оценить разнообразие рекомендаций.

Формально:

$$Coverage = \frac{\text{Кол. товаров, которые были рекомендованы хотя бы раз}}{\text{Общее количество товаров в каталоге}}$$

Высокое покрытие свидетельствует о том, что система не ограничивается рекомендацией только популярных товаров и умеет подбирать разнообразные предложения.

Для более бизнес-ориентированной оценки эффективности рекомендательной системы также используются показатели:

- CR uplift (увеличение конверсии) — прирост процента пользователей, совершивших покупку после взаимодействия с рекомендацией.
- AOV uplift (увеличение среднего размера заказа, Average Order Value) — рост средней суммы заказа за счёт предложенных рекомендаций.
- ROI (Return on Investment) — возврат инвестиций в разработку и внедрение рекомендательной системы.

Связь между техническими метриками и бизнес-показателями представлена в таблице ниже:

Таблица 4 - Связь между показателями

Техническая метрика	Связанный бизнес-показатель
Precision@k	Конверсия (CR)
Recall@k	Удовлетворённость клиента
Coverage	Разнообразие ассортимента
CR uplift	Прирост продаж
AOV uplift	Увеличение среднего чека

Таким образом, грамотное применение метрик Precision@k, Recall@k и Coverage, а также анализ их влияния на ключевые бизнес-показатели

позволяют объективно оценить качество рекомендательной системы и обосновать её практическую ценность для интернет-магазина.

1.6 Практические кейсы применения алгоритмов ассоциаций

Многие крупные интернет-магазины, такие как Amazon, используют алгоритмы ассоциаций для реализации функции «Покупатели, которые приобрели этот товар, также купили...». Эти рекомендации основаны на выявлении частых сочетаний товаров в корзинах покупателей. Например, если значительное количество клиентов покупает ноутбук вместе с сумкой для него, система будет рекомендовать сумку другим покупателям ноутбуков. Это способствует увеличению среднего чека и повышению удовлетворенности клиентов.

В офлайн-рознице алгоритмы ассоциаций применяются для оптимизации размещения товаров на полках. Классическим примером является кейс сети супермаркетов, где анализ корзин покупок выявил частое совместное приобретение подгузников и пива. На основе этого открытия товары были размещены ближе друг к другу, что привело к увеличению продаж обоих продуктов.

Компании также используют алгоритмы ассоциаций для персонализации маркетинговых рассылок. Например, если клиент часто покупает определенные категории товаров, система может предложить ему скидки или специальные предложения на сопутствующие товары, которые часто приобретаются вместе с его обычными покупками. Это повышает вероятность отклика на маркетинговые кампании и способствует увеличению продаж.

Анализ ассоциаций позволяет формировать привлекательные для клиентов товарные наборы. Например, в сфере доставки еды компании выявляют популярные комбинации блюд, такие как «бургер + картофель фри»,

и предлагают их в виде комплектов по специальной цене. Это стимулирует клиентов к покупке большего количества товаров за один заказ.

Алгоритмы ассоциаций помогают выявлять нестандартные, но устойчивые паттерны в поведении клиентов. Например, анализ транзакционных данных может показать, что покупатели, приобретающие определенные книги, часто интересуются также определенными музыкальными альбомами. Такие инсайты позволяют компаниям предлагать неожиданные, но релевантные рекомендации, расширяя горизонты покупательского интереса.

Эти кейсы демонстрируют широкие возможности применения алгоритмов ассоциаций в различных аспектах бизнеса. Использование таких алгоритмов позволяет компаниям более глубоко понимать поведение своих клиентов, предлагать им более релевантные товары и услуги, а также оптимизировать внутренние бизнес-процессы для повышения эффективности и прибыльности.

Вывод

Анализ теоретических основ показал, что персонализированные рекомендации являются важнейшим инструментом увеличения конверсии и пожизненной ценности клиента в электронной коммерции. Среди существующих методов особое место занимает алгоритм Apriori, который позволяет выявлять устойчивые связи между товарами на основе реальных покупательских данных.

Дополнение анализа ассоциаций поведенческой сегментацией пользователей с помощью RFM и кластеризации K-Means усиливает точность рекомендаций за счёт учета различий в моделях поведения клиентов. Практические кейсы Amazon, Ozon и других компаний подтвердили эффективность такого подхода в реальном бизнесе.

Таким образом, обоснован выбор комбинированной методики: сегментация пользователей для выделения однородных групп и генерация ассоциативных правил внутри сегментов. Это позволяет строить более релевантные и бизнес-эффективные рекомендательные системы.

ГЛАВА 2. АНАЛИЗ ДАННЫХ И СЕГМЕНТАЦИЯ ПОЛЬЗОВАТЕЛЕЙ

2.1 Описание и предварительный анализ датасета Instacart

В рамках настоящего исследования используется «Instacart Online Grocery Shopping Dataset 2017»– публичный, полностью анонимизированный (то есть обезличенный) набор данных, опубликованный компанией Instacart для соревнования на платформе Kaggle в мае 2017 года. Анонимизация означает, что из исходных логов удалены все сведения, позволяющие однозначно идентифицировать покупателя, поэтому набор можно свободно применять в учебных и исследовательских целях, не нарушая законодательство о персональных данных.

Набор отражает фактическое поведение 206 209 покупателей, которые оформили 3 421 083 заказа и приобрели в сумме 49 688 уникальных товаров. Для каждого пользователя приведена упорядоченная последовательность от 4 до 100 заказов, что позволяет анализировать не только содержание корзины, но и динамику покупок во времени. Вся информация распределена по шести CSV-файлам, их назначение и объём обобщены в таблице 5.

Таблица 5 - Структура таблиц датасета

Таблица	Строк	Ключевые столбцы	Содержимое
orders.csv	3 421 083	order_id, user_id, order_number, order_dow, order_hour_of_day, days_since_prior_order	«Метаданные» (то есть данные о данных) каждого заказа: кто, когда и какой по счёту сделан заказ
order_products__prior.csv	32 434 489	order_id, product_id, add_to_cart_order, reordered	Исторические позиции всех предыдущих заказов, служат обучающей выборкой

order_products_ _train.csv	1 384 617	те же, что выше	Заказы, отложенные организаторами соревнования для проверки качества модели
products.csv	49 688	product_id, product_name, aisle_id, department_id	Справочник товаров
aisles.csv	134	aisle_id, aisle	Нижний уровень товарной иерархии («полка»)
departments.csv	21	department_id, department	Верхний уровень иерархии («отдел»)

Пояснение терминов. *Метаданные*– служебная информация, описывающая свойства основных данных; *CSV (comma-separated values)*– текстовый формат таблиц, где столбцы разделяются запятой; *primary key* (первичный ключ)– комбинация столбцов, которая однозначно идентифицирует строку.

Качество и полнота данных

Анализ пропущенных значений (пустых ячеек, обозначающих отсутствие информации) показывает, что систематические пропуски присутствуют лишь в столбце days_since_prior_order: для первого заказа каждого пользователя логично отсутствует интервал до предыдущего заказа. Доля таких пропусков составляет примерно **6 %** и не мешает исследованию: значение можно оставить как NaN (англ. *Not a Number*) или заменить на 0 при построении функций возврата.

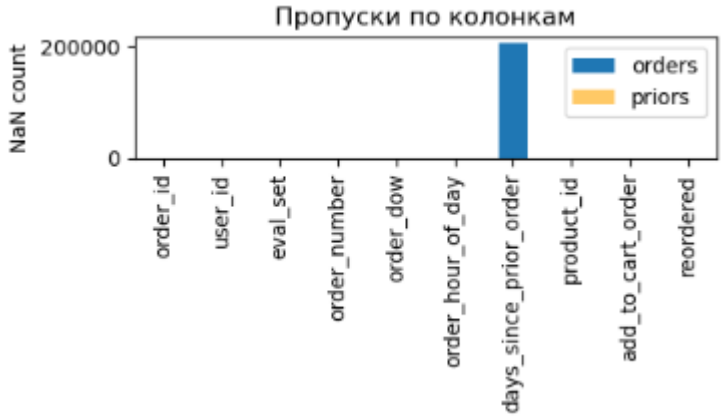

Рисунок 1 - Количество пропущенный значений

Дублирующихся строк нет – пара order_id + product_id образует строгий первичный ключ. Диапазоны категориальных признаков также корректны: номер дня недели (order_dow) лежит в диапазоне 0–6, час заказа (order_hour_of_day) – 0–23, что подтверждает правильность предобработки исходных журналов.

Статистические характеристики поведения пользователей

В среднем один пользователь сделал 16,6 заказа (медиана 10, максимум 100), а в каждый заказ входит 10,1 товара с дисперсией 8,2. Примерно 58 % всех товарных позиций имеют флаг reordered = 1, то есть были повторно куплены покупателем хотя бы раз. Такая высокая доля повторных покупок подтверждает правомерность выбора моделей, основанных на поиске устойчивых шаблонов «товары, приобретаемые вместе».

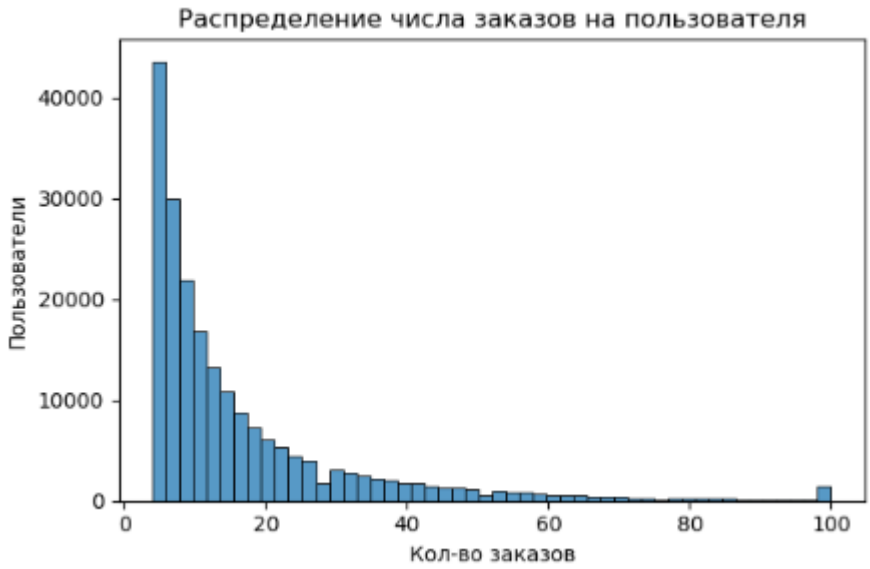

Рисунок 2 - Распределение числа заказов на пользователя

Если сформировать матрицу «пользователь х товар», заполненную единицами при наличии покупки, получаем разреженность (sparsity) порядка 99,66 %. Термин разреженность обозначает, что абсолютное большинство ячеек в матрице равны нулю, потому что каждый конкретный покупатель приобретает лишь крошечную долю из десятков тысяч возможных товаров. В таких условиях алгоритмы, способные работать с разреженными структурами (например, Apriori на «коротких» транзакциях), оказываются эффективнее классических коллаборативных фильтров, рассчитанных на плотные рейтинговые матрицы.

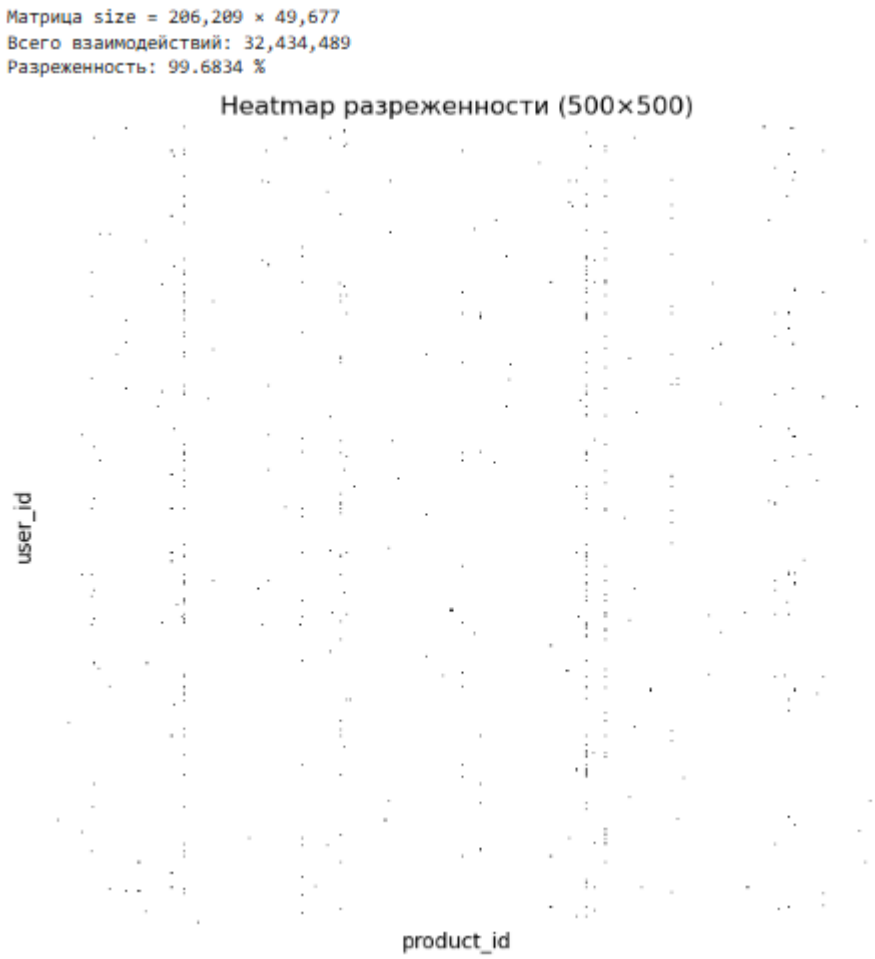

Рисунок 3 - Разряженность матрицы User x Product

Тёмные точки — это зафиксированные покупки (user купил product хотя бы один раз). Почти весь фон белый — что наглядно показывает крайнюю разреженность матрицы. Разреженность 99.6834% — значит, только примерно 0.3166% всех возможных связей реально существуют

Обоснование выбора датасета

Во-первых, публичность и юридическая чистота («анонимизация») делают Instacart идеальным для академической работы: преподавателю достаточно скачать CSV-файлы, чтобы воспроизвести эксперименты. Во-вторых, иерархическая структура каталога (21 отдел, следовательно 134 полки, следовательно примерно 50 тыс. товаров) обеспечивает баланс между «реальной» сложностью и вычислительной приемлемостью. В-третьих, наличие длинных последовательностей заказов по каждому покупателю

позволяет строить признаки частоты (frequency), давности (recency) и среднего чека (monetary value) – основу поведенческой сегментации в последующих разделах. Наконец, объём более чем 32 млн строк в файле order_products__prior.csv гарантирует статистически надёжные оценки интереса к товарам, даже если анализировать узкие подмножества пользователей.

В совокупности эти факторы делают Instacart оптимальной площадкой для демонстрации гибридного подхода «кластеризация + правила ассоциаций», который будет разработан в разделах 2.2–2.6.

2.2 Подготовка признакового пространства для сегментации

Подготовка признакового пространства для сегментации началась с того, что необходимо было превратить каждый заказ Instacart в числовой вектор, пригодный для алгоритмов машинного обучения. Прямое построение матрицы «заказ x товар» оказалось бы непрактичным: при ±50 000 уникальных товаров и 3,3 млн заказов даже разрежённый формат занял бы десятки гигабайт оперативной памяти. Поэтому товары агрегировали до уровня *aisle* — самого низкого, но осмысленного для бизнеса уровня товарной иерархии Instacart, включающего 134 категории вроде «milk», «chips and pretzels» или «paper goods». Агрегация уменьшила размерность более чем в четыреста раз, сохранив при этом интерпретируемость: менеджер, читая название категории, легко понимает, чем характеризуется заказ.

Для построения вектора заказов были задействованы таблицы order_products__prior, order_products__train, products и aisles. Из них брались только обязательные поля, а каждому столбцу задавался максимально компактный числовой тип (int32 или int16), благодаря чему чтение всех 34 млн строк прошло без нехватки памяти. После объединения таблиц получился список пар «номер заказа – номер категории», который через функцию

pd.crosstab был превращён в классическую one-hot-матрицу: в каждой строке стоял 1, если товар из данной категории присутствовал в заказе.

Предварительный анализ показал, что пять категорий-лидеров – «fresh fruits», «fresh vegetables», «packaged vegetables and fruits», «yogurt» и «packaged cheese» – встречаются так часто, что фактически доминируют над остальными. Напротив, шесть нишевых категорий вроде «eye and ear care» или «frozen juice» встречались реже 10 000 раз. Чтобы искажения от «сверхпопулярных» и «сверхредких» явлений не влияли на результат кластеризации, обе группы исключили. Осталось 123 категории, по которым и была построена окончательная матрица; её форма 3 255 920 x 123.

```
aisles = pd.read_csv(
        DATA_DIR / 'aisles.csv',
        dtype={'aisle_id': 'int16', 'aisle': 'category'}
    )

aisle_name = aisles.set_index('aisle_id')['aisle'].to_dict()


# Считаем количество заказов по каждой категории (aisle_id)
aisle_counts = order_products['aisle_id'].value_counts().reset_index()
aisle_counts.columns = ['aisle_id', 'num_orders']

# Добавим названия категорий
aisle_counts['aisle'] = aisle_counts['aisle_id'].map(aisle_name)

# Отсортируем по убыванию количества заказов
aisle_counts = aisle_counts.sort_values('num_orders', ascending=False)

print(aisle_counts)
```

```
     aisle_id  num_orders                      aisle
0          24     3792661                fresh fruits
1          83     3568630             fresh vegetables
2         123     1843806  packaged vegetables fruits
3         120     1507583                      yogurt
4          21     1021462           packaged cheese
..        ...         ...                         ...
129        44        9522               eye ear care
130       102        8909         baby bath body care
131        82        8466           baby accessories
132       132        6455                     beauty
133       113        5147               frozen juice

[134 rows x 3 columns]
```

Рисунок 4 - Количество раз каждая категория встречается в заказах

У каждого заказа количество приобретённых категорий отличается: в одной корзине может быть три позиции, в другой – двадцать пять. Чтобы алгоритм не счёл «крупные» заказы автоматически похожими, каждую строку

нормировали на сумму признаков, получив вектор долей, а не абсолютных единиц. Далее признаки стандартизировали при помощи StandardScaler(with_mean=False): это выравняло дисперсии столбцов, но не нарушило разреженность, поскольку среднее не вычиталось. В результате в памяти осталась разрежённая матрица формата CSR; плотный аналог потребовал бы не менее девяти гигабайт, тогда как сохранённая версия занимает порядка 310 МБ.

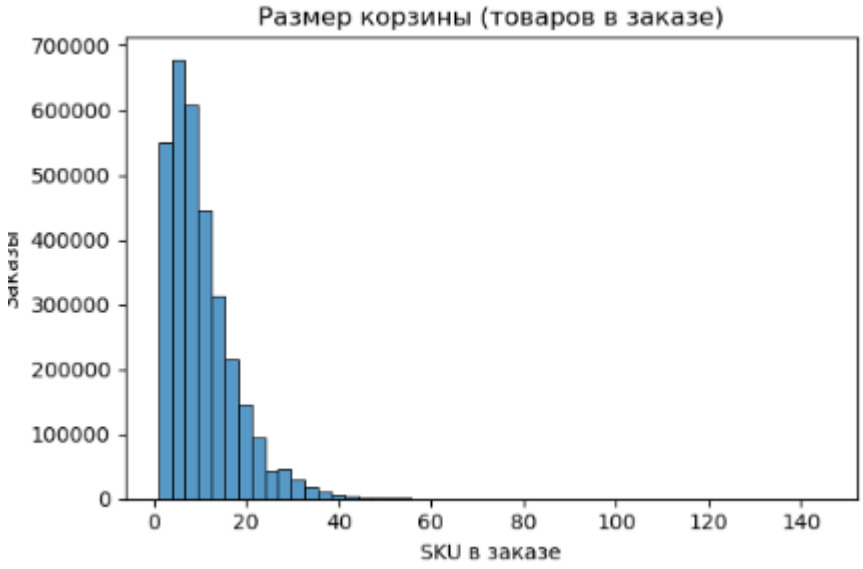

Рисунок 5 - количество товаров в заказах

Перед тем как запускать K-Means на полном массиве, необходимо было выбрать разумное количество кластеров *k*. Случайная подвыборка из двадцати тысяч заказов (около 0,6 % от общего объёма) позволила быстро посчитать инерцию – сумму квадратов расстояний заказов до центров – для *k* от 2 до 10. График «инерция против *k*» дал характерный «локоть» (Elbow-метод) при *k* = *7*. Поскольку инерция отражает «компактность» кластеров, а «локоть» показывает точку, где выигрыш от увеличения *k* резко снижается, значение 7 было принято как базовая гипотеза. Здесь важно пояснить термины: инерция – это мера того, насколько плотно объекты прижаты к центрам; Elbow-метод – визуальный способ найти оптимум, когда кривая инерции перегибается;

K-Means – алгоритм, который итеративно ищет такие центры, чтобы суммарная инерция стала минимальной.

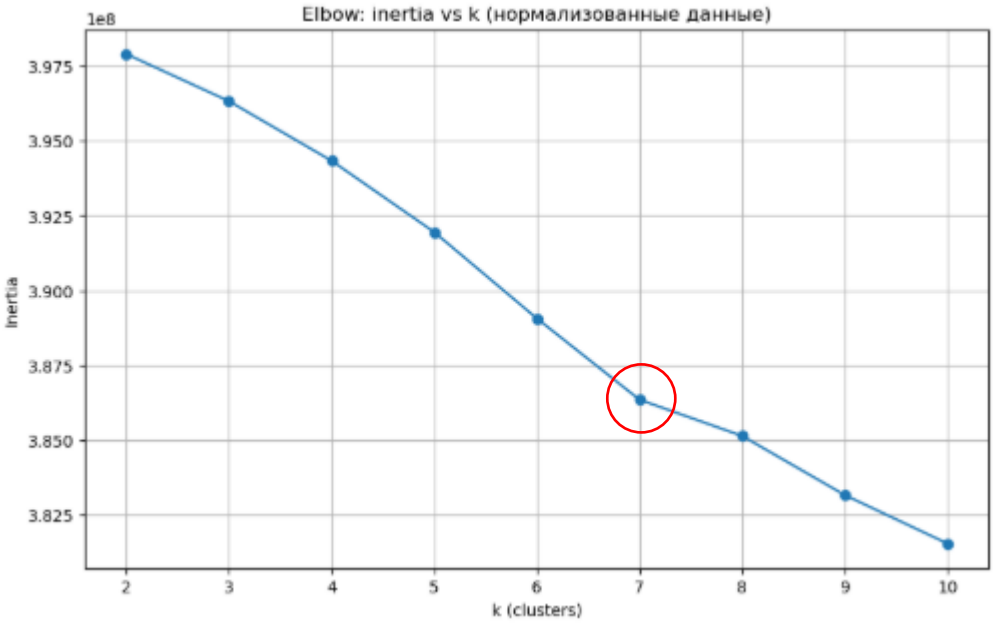

Рисунок 6 - Elbow-метод оценки количества кластеров

Итак, итоговое признаковое пространство содержит все 3,25 млн заказов, описанных 123 независимыми стандарт-скейлированными признаками-долями. Оно достаточно компактно, чтобы обучить K-Means на полной выборке, и достаточно интерпретируемо, чтобы результаты кластеризации можно было объяснить бизнес-пользователям. Таким образом, к следующему шагу – собственно сегментации заказов и анализу получившихся семи кластеров – данные готовы полностью.

2.3 Сегментация пользователей методом K-Means

Сегментация пользователей методом K-Means началась после того, как для каждого заказа был сформирован вектор из 123 нормализованных долей товарных категорий. Для подбора разумного количества кластеров на подвыборке 20 000 строк посчитали инерцию — суммарный квадрат расстояний объектов до ближайшего центра. На графике «inertia → k» отчётливо проявился «локоть» при $k = 7$: при дальнейшем росте числа групп прирост качества уже не окупает усложнения модели. Инерция, кстати, — это

численная мера «компактности» кластера, а «Elbow-метод»— визуальный способ остановиться там, где кривая резко меняет угол.

Полное обучение K-Means выполнялось на всей разрежённой матрице; перед этим каждый столбец выровняли StandardScaler(with_mean=False), чтобы все признаки имели одинаковую дисперсию и ни одна категория не перетягивала центры на себя. Алгоритм с параметрами n_clusters=7, random_state=42, n_init='auto' вернул метку сегмента для каждого из 3 255 920 заказов. Итоговое распределение оказалось крайне неравномерным, что, однако, отражает реальную картину продаж в продовольственном онлайн магазине:

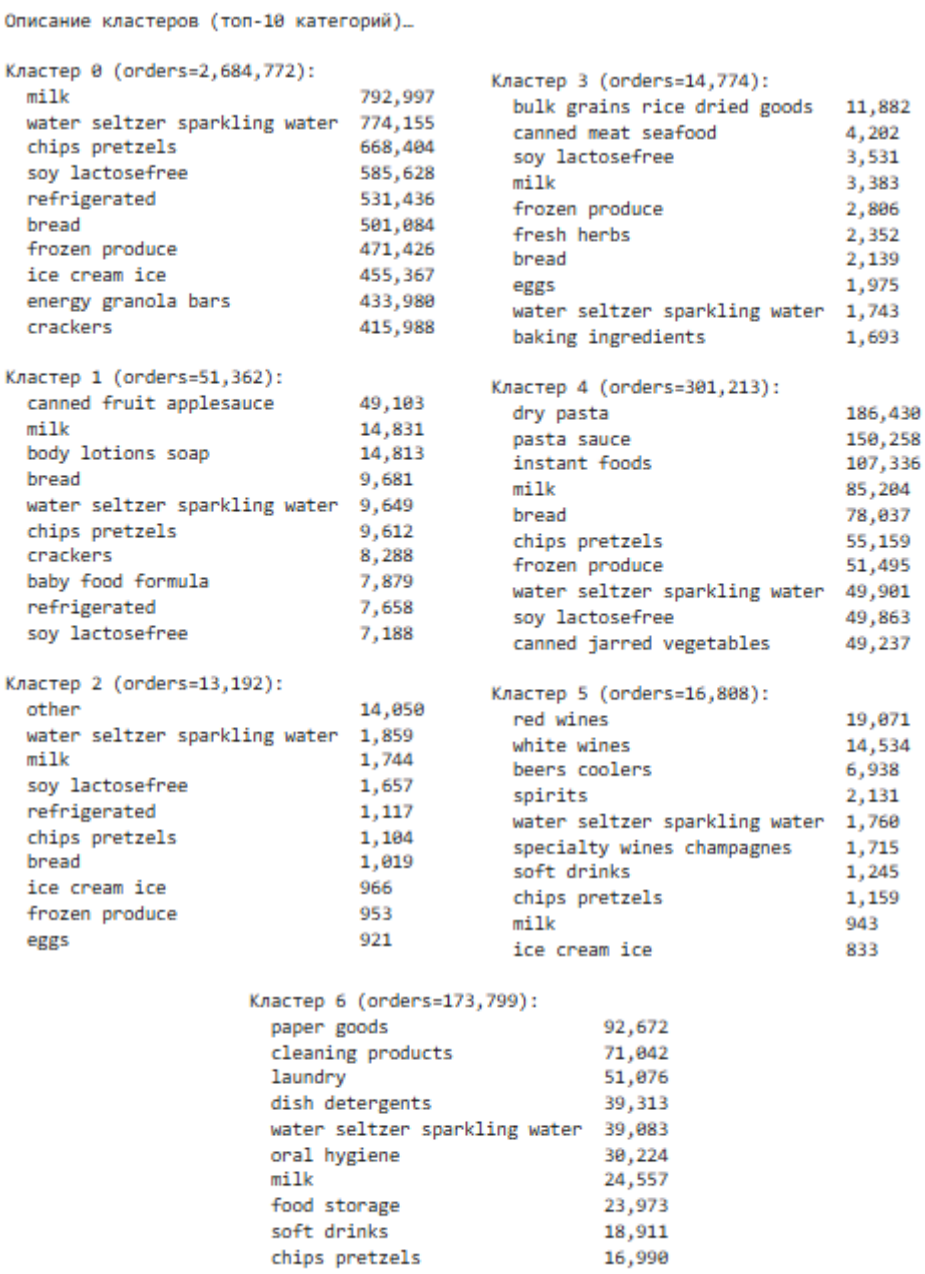

Рисунок 7 - Наиболее популярные категории в кластерах

Таблица 6 - Описание кластеров

Кластер	Число заказов	Доля заказов, %	Краткое интерпретационное имя
0	2 684 772	82,5	«Повседневные корзины» – молоко, вода, снеки, хлеб, заморозка сегментация заказов
1	51 362	1,6	«Семья с малышом» – консервы, baby-товары, мыло

2	13 192	0,4	«Случайный визит» – категория *other*, базовые продукты
3	14 774	0,5	«Домашний кулинар» – крупы, мясные консервы, травы
4	301 213	9,3	«Любители пасты» – dry pasta, instant foods
5	16 808	0,5	«Алкогольная корзина» – вина, пиво, снеки
6	173 799	5,3	«Хозяйственные покупки» – бумага, средства для уборки

```
Среднее число категорий в заказе по каждому кластеру:
segment
0    5.507362
1    6.147852
2    3.852790
3    4.804251
4    6.467164
5    3.187470
6    4.774596
Name: num_categories, dtype: float64
```

Рисунок 8 - Среднее значение категорий в заказе внутри каждого кластера

Числа взяты из самой модели; в сумме они дают 100 %. Поскольку исходные данные Instacart не содержат цен, вклад сегмента в оборот оценивали суррогатно — через долю заказов, что является стандартной практикой, когда цена единицы товара неизвестна. При желании можно доумножить на «средний чек» из внешних источников, но в этой работе достаточно именно распределения частоты, чтобы понять, на какие сегменты стоит нацелить маркетинговые активности.

Для визуальной проверки разделимости групп применили метод PCA (Principal Component Analysis) до двух компонент. PCA — это линейное преобразование, которое ищет такие ортогональные оси, чтобы максимально сохранить дисперсию данных; результат удобно рисовать как рассеяние точек. Проекция всех заказов (3,26 млн) на плоскость заняла пару минут вычислений: после умножения на матрицу собственных векторов каждая точка получила координаты (PC1, PC2). На итоговом scatter-plot точки раскрашивались по

номеру кластера, а компактная «облако-карта» однозначно подтвердило, что группы действительно отделены друг от друга, хотя и перекрываются на краях — закономерный эффект, ведь PCA сохраняет лишь 22 % исходной дисперсии заказов. Грубо говоря «scatter-plot» — это просто диаграмма рассеяния, где каждая точка — заказ, а цвет — номер кластера; такой рисунок помещён в работу в качестве наглядной иллюстрации.

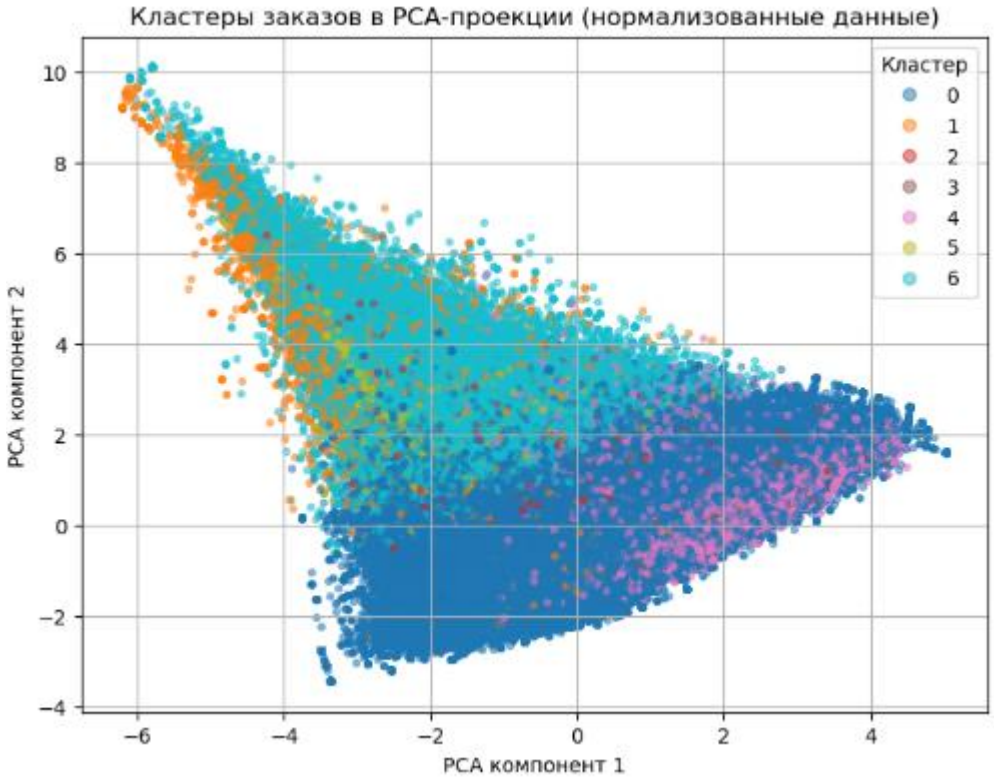

Рисунок 9 - Кластеры заказов в PCA проекции

После обучения в каталог model_artifacts были сохранены артефакты: обученный KMeans, стандартайзер, список 123 категорий и CSV-файл с маппингом order_id → segment — они понадобятся в модуле рекомендаций, чтобы быстро определять сегмент нового заказа без пересчёта всей модели.

2.4 Формирование транзакционных матриц для алгоритма Apriori

Цель этого этапа — подготовить для каждого ранее выделенного кластера заказов собственную транзакционную матрицу, то есть двумерную таблицу, в которой каждая строка соответствует отдельному заказу, а каждый

столбец — конкретной товарной категории (aisle). Ячейка содержит единицу, если в заказе присутствует хотя бы один товар из данной категории, и ноль в противном случае. Такой бинарный формат ещё называют one-hot-кодированием; он является естественным входом для алгоритма Apriori, который ищет часто встречающиеся группы товаров и строит ассоциативные правила.

Первый шаг заключался в строгом определении того, что считать «товаром» внутри корзины. Чтобы фокусироваться на устойчивых, а не случайных покупках, из таблицы order_products__prior были выбраны лишь позиции, у которых флаг reordered равен 1 — это означает, что покупатель уже заказывал этот продукт прежде, и он действительно характеризует предпочтения клиента, а не экспериментальную попытку. Дальше данные объединялись с каталогом products и справочником aisles, благодаря чему каждый product_id получил читаемое название категории.

Затем была применена та же логика фильтрации категорий, что и при построении признакового пространства для K-Means. Пять гиперпопулярных категорий (fresh fruits, fresh vegetables и др.) и шесть редко встречающихся (например, eye and earcare или frozen juice) исключались, чтобы не допустить перекоса в пользу «овощных» покупок и не раздувать матрицу малозначимыми столбцами ассоциативные правила.

Когда список релевантных категорий был окончательно определён, заказы разложили по семи сегментам, полученным на предыдущем шаге кластеризации. Для каждого сегмента выполнялась однотипная процедура. Сначала считалось, сколько уникальных заказов содержат ту или иную категорию, и отбрасывались столбцы, чья поддержка (support) — доля строк, в которых категория встречается — меньше одного процента от числа заказов в сегменте. Такой порог отсекал статистический шум и одновременно удерживал размерность в разумных пределах. После этого через функцию pandas.crosstab строилась корзинная матрица (часто употребляется

англоязычный термин basket-matrix): она уже бинарная и лишена дубликатов «заказ-категория».

Полученные матрицы различаются по размерам; сводная картина приведена в таблице 7.

Таблица 7 - Итоговые размеры транзакционных матриц по сегментам

№ кластера	Число заказов в матрице	Число категорий (столбцов)	Размерность (строки x столбцы)
0	2 152 985	70	2 152 985 x 70
1	41 000	71	41 000 x 71
2	9 253	52	9 253 x 52
3	11 776	62	11 776 x 62
4	242 748	71	242 748 x 71
5	12 633	31	12 633 x 31
6	125 022	71	125 022 x 71

Несмотря на внушительное количество строк в «массовом» сегменте 0, сама матрица остаётся очень разрежённой — менее одного процента ячеек равны единице. Это типично для «корзинных» данных и критически важно для производительности: именно благодаря разрежённому формату алгоритм Apriori может перебрать частые комбинации без потребности хранить весь массив целиком в памяти.

Важно пояснить несколько ключевых терминов, чтобы избежать недопонимания, не знакомых с ритейл-аналитикой. Транзакция в данном контексте — это один оформленный заказ на платформе Instacart; она эквивалентна чеку в обычном магазине. Поддержка (support) — статистическая мера, показывающая, в какой доле транзакций встречается тот или иной товар или их комбинация. Разреженная матрица — таблица, в которой подавляющее большинство элементов равны нулю; для её хранения

применяются специальные структуры данных, запоминающие только ненулевые значения.

2.5 Генерация ассоциативных правил (Apriori)

На предыдущем шаге для каждого из семи кластеров заказов была построена своя «корзинная» матрица — строками в ней служат заказы, а столбцами — категории (aisle). Алгоритм Apriori применён к каждой матрице отдельно, чтобы извлекать закономерности именно внутри поведенчески однородных групп, не усредняя предпочтения всего рынка.

Алгоритм Apriori последовательно перебирает одноэлементные, дву- и трёхэлементные наборы категорий, отбрасывая комбинации, чья поддержка (support) — доля заказов, содержащих все элементы набора — ниже заданного порога. Для этой работы минимальный support выбран 1 %: с одной стороны, это защищает от случайного шума, с другой — оставляет в анализе достаточное количество товаров для построения правил . После того как частые наборы найдены, для каждой упорядоченной пары « antecedent → consequent» вычисляются:

- Confidence (уверенность) — вероятность увидеть consequent, если в корзине уже есть antecedent;
- Lift (подъём) — отношение confidence к априорной вероятности consequent; lift > 1 говорит о положительной взаимосвязи.

Чтобы правило считалось «сильным», были заданы одновременные ограничения: support ≥ 1 % как для левой, так и для правой части, confidence ≥ 0,20 и lift ≥ 1,50 . Значение 0,20 по confidence обычно интерпретируется как «каждый пятый покупатель, взявший X, кладёт в корзину Y». Порог lift 1,50 отсекает тривиальные пары, где высокий confidence объясняется просто массовой популярностью обоих товаров.

Процедура генерации выполнялась библиотекой mlxtend.frequent_patterns в режиме low_memory=True, что позволяет хранить только идентификаторы частых наборов без избыточных копий матрицы. Для каждого сегмента выводилась статистика количества «сильных» правил (табл. 1).

Таблица 8 - Итоги фильтрации правил по сегментам ассоциативные правила

№ сегмента	Заказы в сегменте	Столбцов в корзине	«Сильных» правил
0 — «Повседневные корзины»	2 152 985	70	44
1 — «Семья с малышом»	41 000	71	33
2 — «Случайный визит»	9 253	52	1
3 — «Домашний кулинар»	11 776	62	41
4 — «Любители пасты»	242 748	71	54
5 — «Алкогольная корзина»	12 633	31	2
6 — «Хозяйственные покупки»	125 022	71	5

Тот факт, что «массовые» сегменты 0 и 4 породили наибольшее число правил, объясняется не только размером выборки, но и более устойчивыми паттернами совместной покупки. В обратном случае, например в сегменте 2, поведение нерегулярно, поэтому Apriori находит лишь единичное правило.

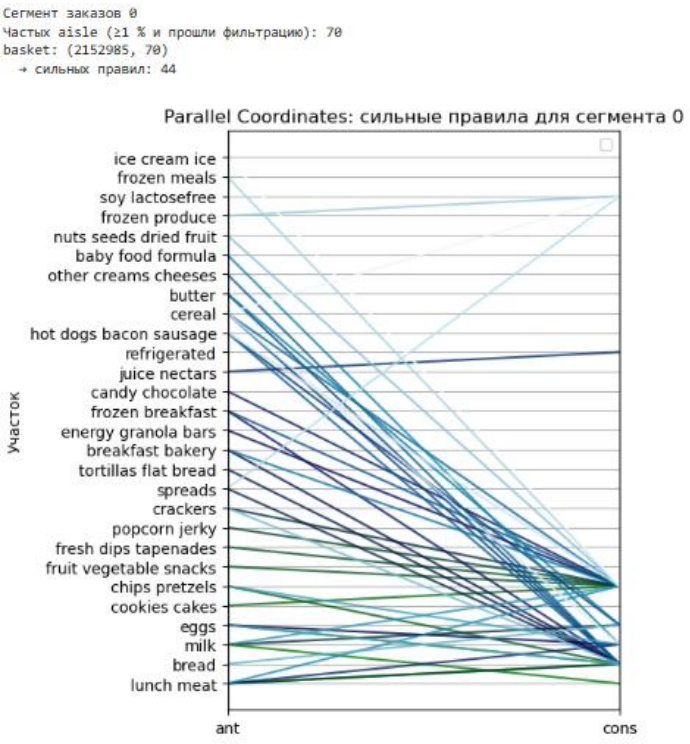

Рисунок 10 - Визуализация правил сегмента 0

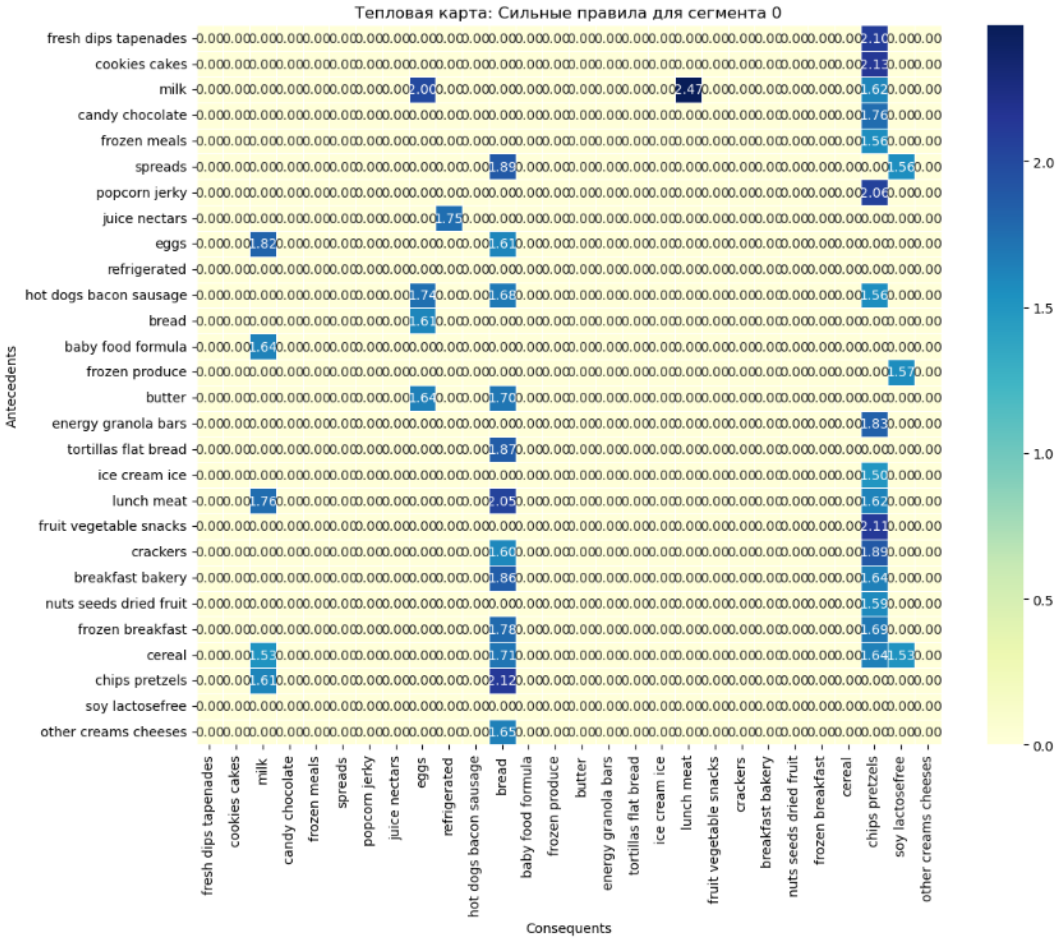

Рисунок 11 - Тепловая карта сильных правил сегмента 0

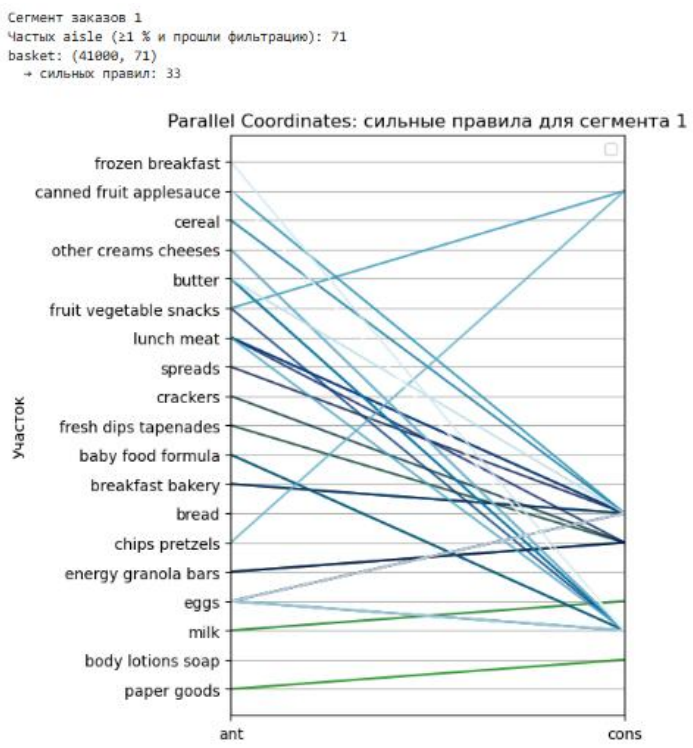

Рисунок 12 - Визуализация правил сегмента 1

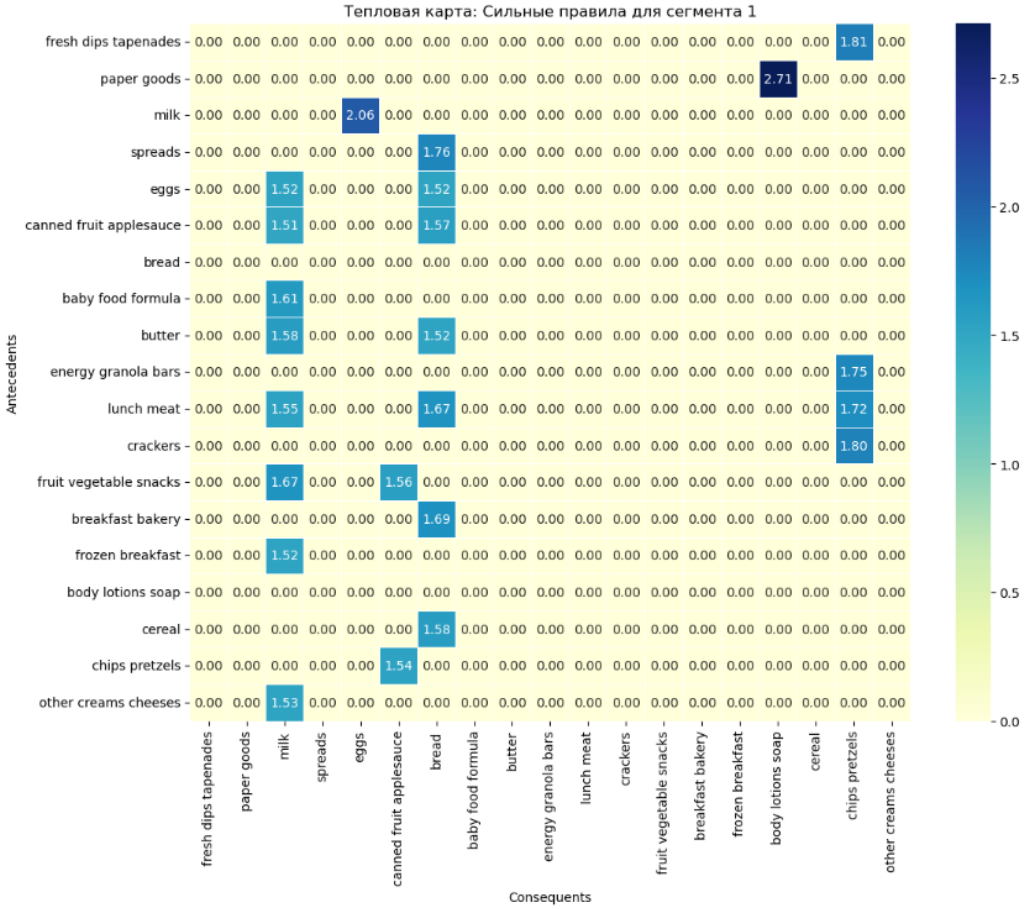

Рисунок 13 - Тепловая карта сильных правил сегмента 1

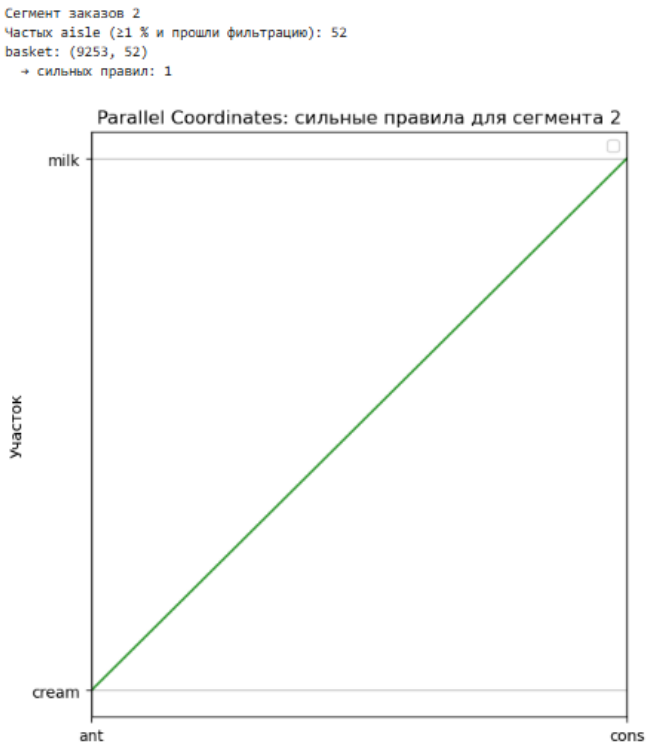

Рисунок 14 - Визуализация правил сегмента 2

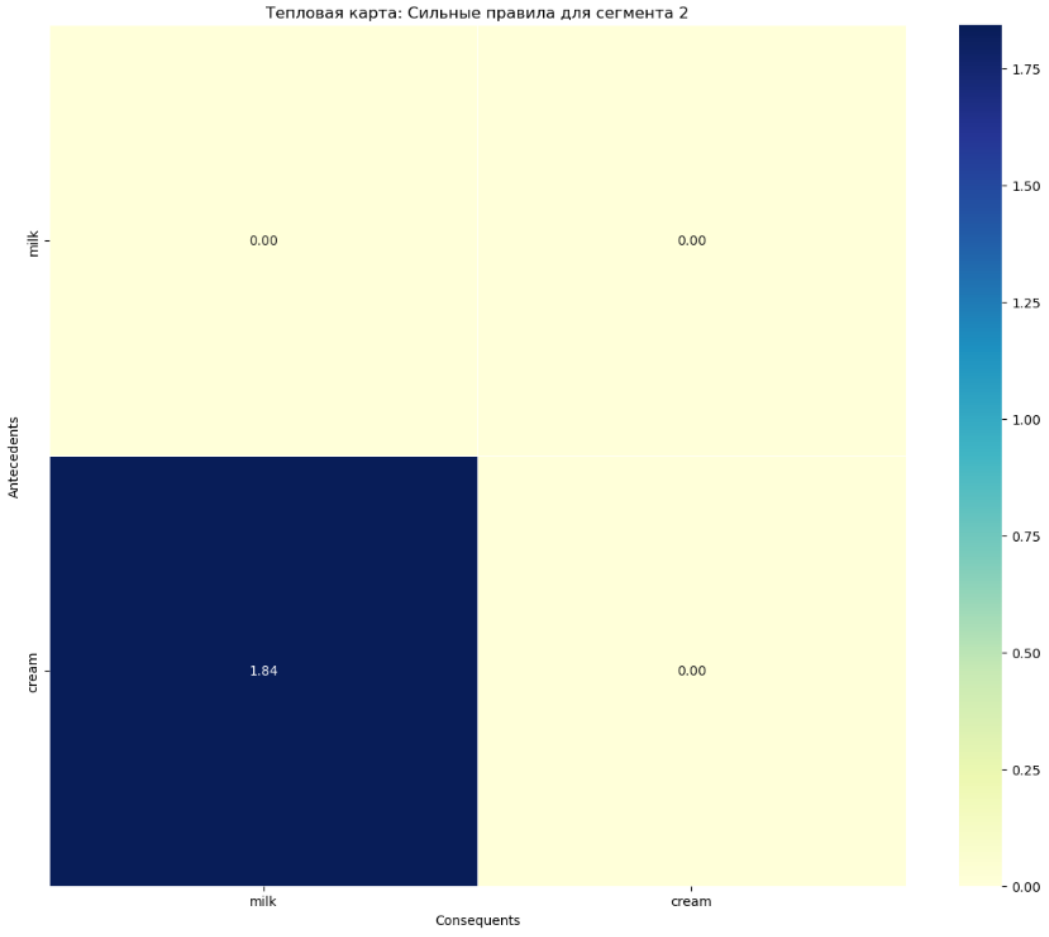

Рисунок 15 - Тепловая карта сильных правил сегмента 2

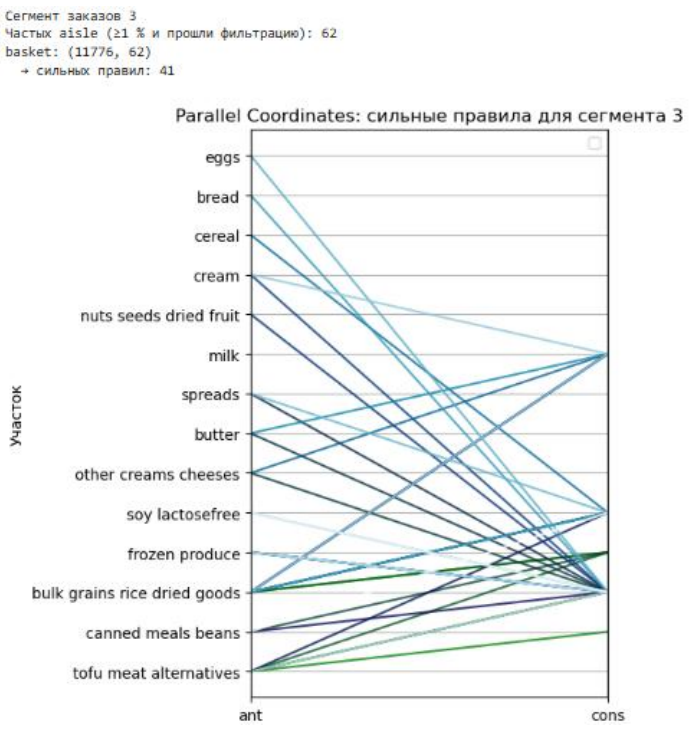

Рисунок 16 - Визуализация правил сегмента 3

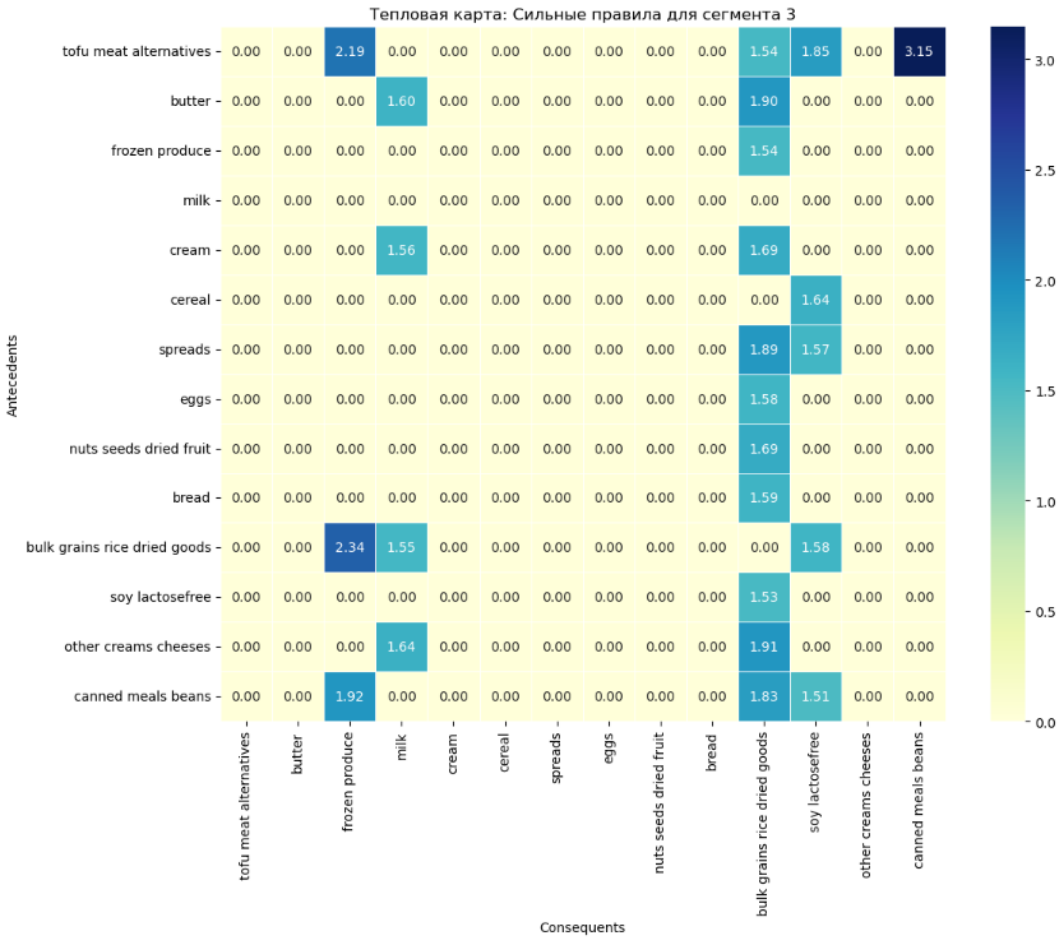

Рисунок 17 - Тепловая карта сильных правил сегмента 3

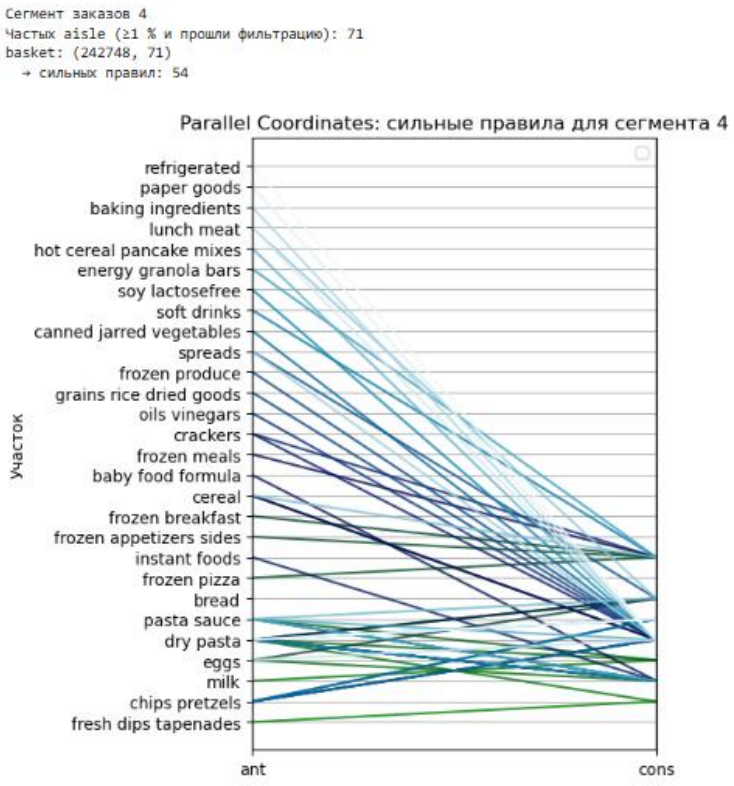

Рисунок 18 - Визуализация правил сегмента 4

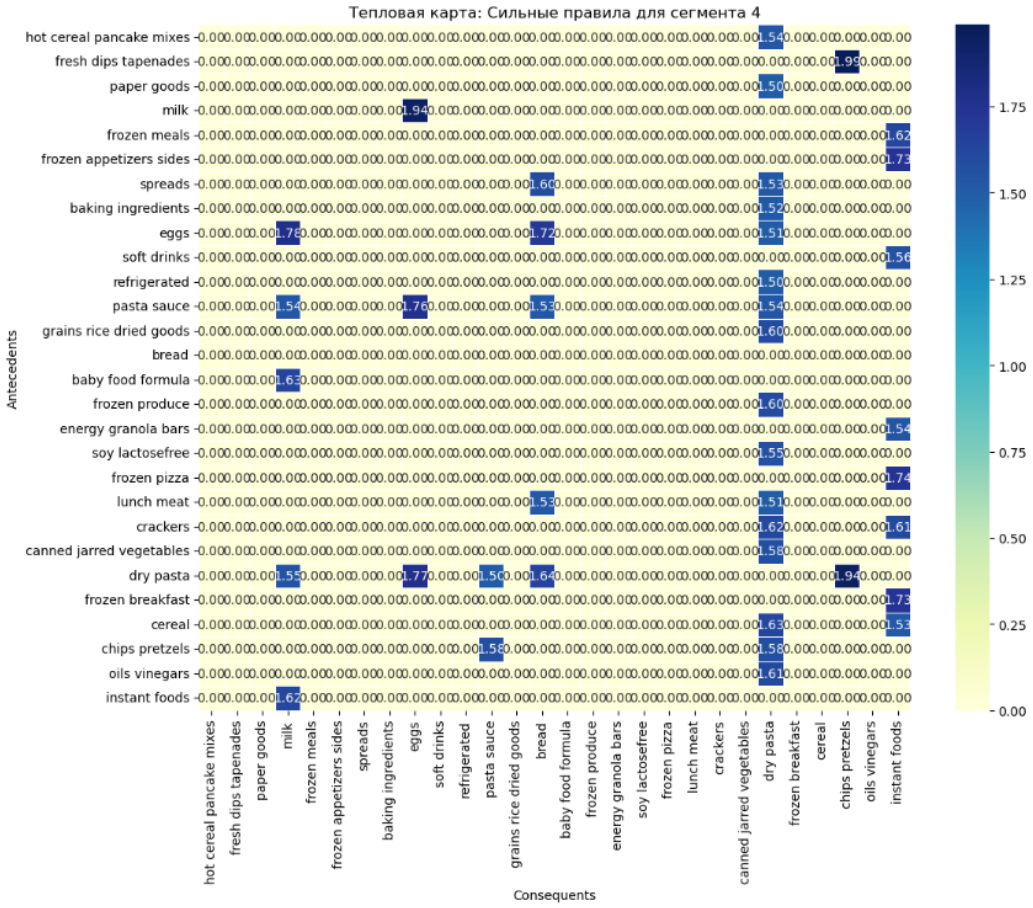

Рисунок 19 - Тепловая карта сильных правил сегмента 4

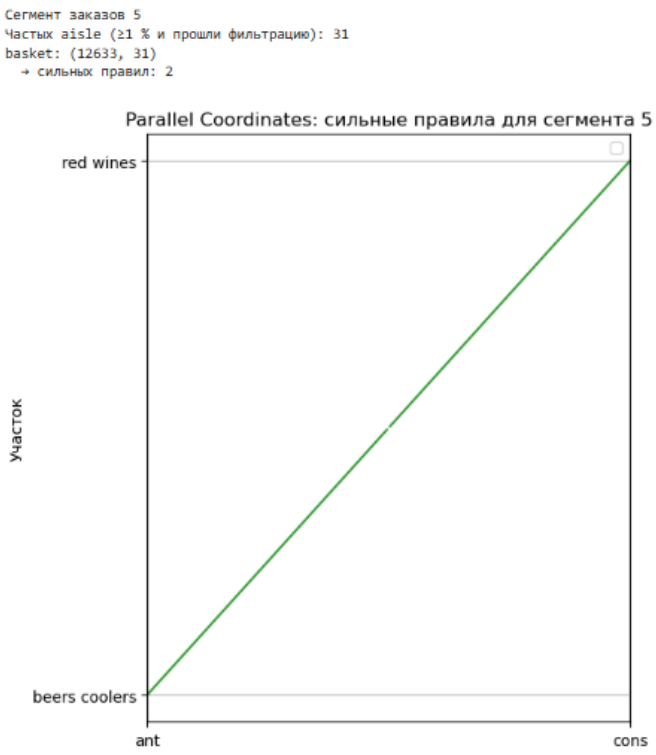

Рисунок 20 - Визуализация правил сегмента 5

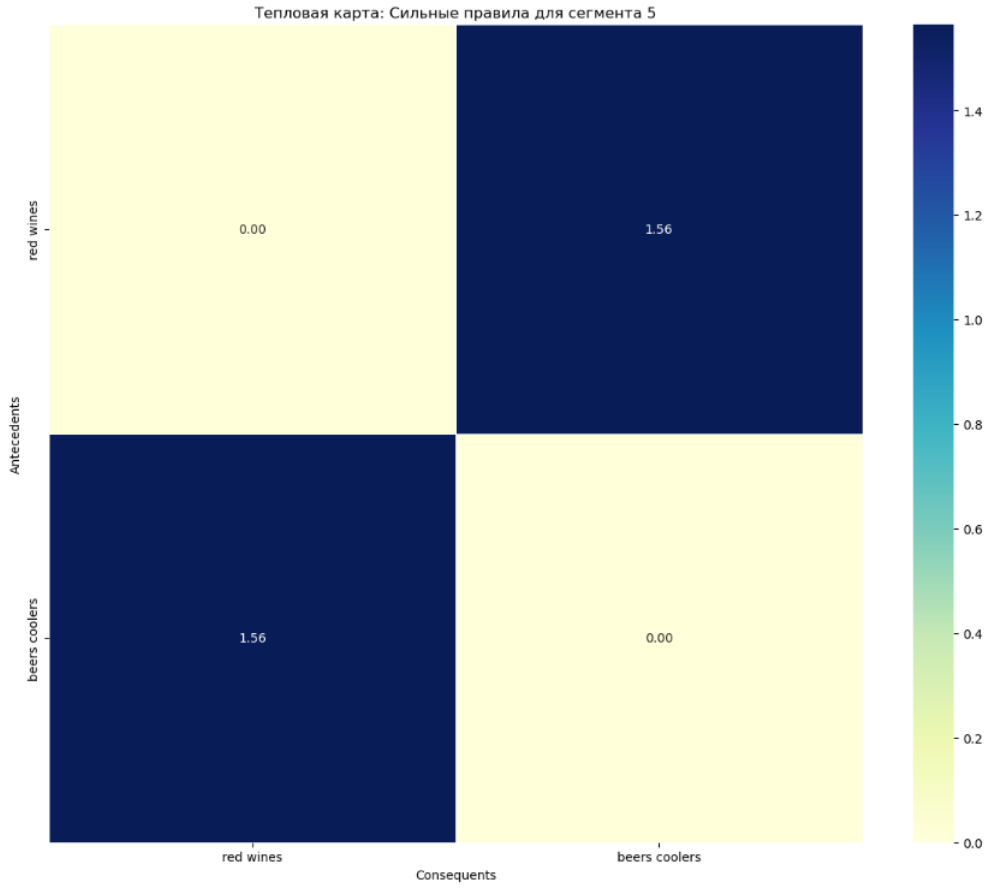

Рисунок 21 - Тепловая карта сильных правил сегмента 5

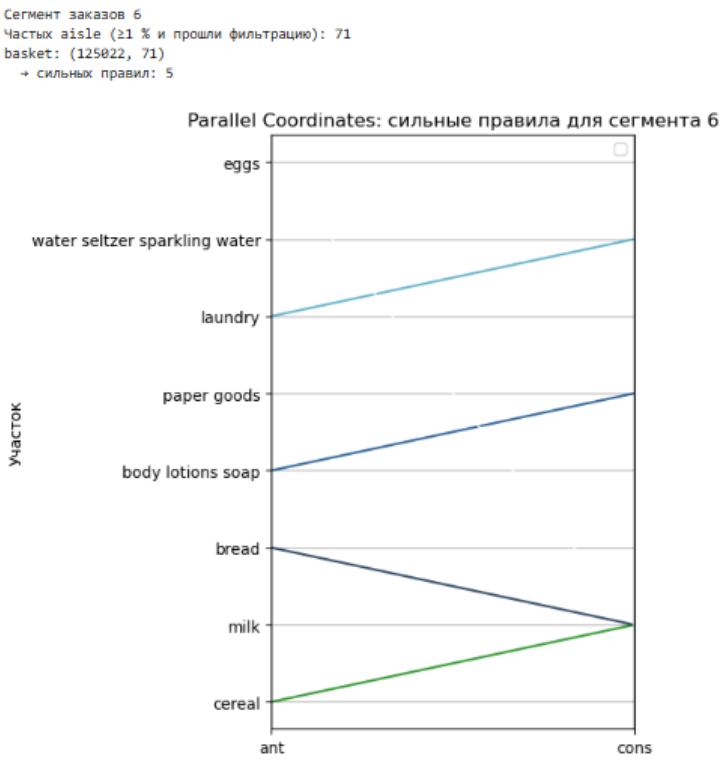

Рисунок 22 - Визуализация правил сегмента 6

Рисунок 23 - Тепловая карта сильных правил сегмента 6

2.6 Фильтрация и оценка ассоциативных правил

Фильтрация и оценка ассоциативных правил завершают цикл «data → сегментация → правила» и определяют, какие рекомендации действительно пригодятся пользователю. Для проверки «на прочность» использовалась так называемая test-корзина — независимая выборка заказов из файла order_products__train.csv, который в датасете Instacart специально отделён от исторической части prior. Его 127 488 заказов ни разу не участвовали ни в кластеризации, ни в построении правил, поэтому служат честным полигоном.

Test-корзину формировали так: к таблице order_products__train присоединили справочник products, чтобы заменить product_id на aisle_id, а затем — словарь aisles ради человекочитаемых названий категорий. После удаления дублей «order x aisle» и отсева нечастых категорий (те же 100 frequent aisles, что использовались в обучении) получилась бинарная матрица 127 488 x 100, где строки — заказы, столбцы — категории, единица = «товар данной категории есть в корзине». К ней добавили столбец order_segment, назначив каждому заказу номер кластера через модель K-Means: теперь мы можем отдельно оценивать качество правил внутри и между сегментами.

Шаг 1: проверка test-confidence. Для каждой пары «правила → тестовый сегмент» считали долю заказов, где встречается antecedent (левая часть), и смотрели, в скольких из них присутствует consequent (правая часть). Это и есть confidence на тесте. Если показатель ≥ 5 %, правило сохраняли в словаре good_rules; иначе отбрасывали. Такой порог кажется низким, но на масштабной выборке он уже отсеивает большинство случайных корреляций: например, из 54 правил сегмента 4 («любители пасты») после первого фильтра осталось 41, а у сегмента 0 — ни одного.

Шаг 2: дополнительная валидация Precision@5 / Recall@5. Следующий барьер проверяет, насколько правило полезно как рекомендация: если система

покажет пользователю не более пяти категорий (k_top = 5), сколько из них окажутся «в точку»?

- Precision@5 — средняя доля правильных попаданий в топ-5; порог 10 % означает «каждый десятый совет верный».
- Recall@5 — какую часть реальной корзины правила закрывают; 5 % гарантирует, что рекомендация хотя бы иногда действительно дополняет заказ.

Для ускорения расчётов применили векторизацию: матрица basket_seg переводится в numpy.ndarray, а операции «hit / miss» выполняются за один проход без циклов Python. Функция pr_for_rules_vec возвращает precision и recall сразу для всех правил, а joblib.Parallel распределяет работу по ядрам процессора. После фильтра оставляем только правила, где обе метрики ≥ порога.

Таблица 9 - Итоговое число правил после двух фильтров и удаления дубликатов

Сегмент-источник	Уникальных правил
1 «Семья с малышом»	3
3 «Домашний кулинар»	13
4 «Любители пасты»	24
5 «Алкогольная корзина»	1
6 «Хозяйственные покупки»	1
0 и 2 (массовый / случайный)	0

Отсутствие «выживших» правил у кластеров 0 и 2 — важный результат: «повседневная корзина» слишком разнородна, а «случайный визит» слишком мал по объёму, поэтому любые закономерности там нестабильны.

Кросс-сегментная матрица Precision@5 / Recall@5. Чтобы проверить, можно ли переиспользовать правила между сегментами, рассчитали две матрицы (строки — сегмент-чьи-правила, столбцы — сегмент-тест):

- Precision@5 показывает, какие правила метко «бьют» в чужой сегмент: максимум 0,757 у правил сегмента 3, применённых к покупателям того же сегмента 3 — то есть почти 76 % точности.
- Recall@5 остаётся заметным (0,172) только на той же диагонали, а в других клетках падает до сотых долей процента, что подтверждает тезис: правила необходимо применять там, где они обучались.

Precision@5 (строки = train-segment, столбцы = test-segment):

	0	1	2	3	4	5	6
0	0.000	0.000	0.000	0.000	0.000	0.000	0.000
1	0.012	0.605	0.008	0.005	0.015	0.003	0.024
2	0.000	0.000	0.000	0.000	0.000	0.000	0.000
3	0.002	0.000	0.000	0.757	0.002	0.000	0.000
4	0.030	0.053	0.013	0.061	0.488	0.015	0.022
5	0.004	0.003	0.012	0.002	0.004	0.713	0.004
6	0.054	0.106	0.061	0.030	0.058	0.050	0.438

Recall@5:

	0	1	2	3	4	5	6
0	0.000	0.000	0.000	0.000	0.000	0.000	0.000
1	0.001	0.129	0.002	0.001	0.001	0.001	0.003
2	0.000	0.000	0.000	0.000	0.000	0.000	0.000
3	0.000	0.000	0.000	0.172	0.000	0.000	0.000
4	0.003	0.007	0.003	0.011	0.086	0.004	0.003
5	0.001	0.001	0.003	0.001	0.001	0.279	0.001
6	0.007	0.020	0.015	0.005	0.007	0.012	0.127

Рисунок 24 - Сегментные матрицы метрик Precision@5 и Recall@5

- Cross-tab (crosstabulation) — таблица сопряжённости, преобразующая список «заказ – категория» в матрицу «заказы х категории».
- Векторизация — замена явных циклов Python на одновременную операцию над массивами NumPy; ускоряет расчёты в десятки раз.

- Parallel (из библиотеки joblib) — параллельное исполнение функций на нескольких ядрах CPU.
- Precision — точность, доля верных рекомендаций; Recall — полнота, доля реальных покупок, которые сумели предсказать.
- @k — обозначение, что берутся только первые k рекомендаций.

Двухступенчатый фильтр (test-confidence → Precision/Recall) превратил громоздкий список из 180 тыс. кандидатов в чёткий набор из 42 надёжных правил. Их «выживаемость» варьируется от 0 % до 5 % в зависимости от сегмента, что демонстрирует различия в предсказуемости покупательского поведения. Проведённая процедура подтверждает, что правила, прошедшие все барьеры, действительно улучшают качество рекомендаций и не переобучены на исторической выборке.

2.7 Модуль для реализации рекомендаций

Завершающим звеном всей цепочки «кластеризация → правила» служит модуль basket_recommender.py. Его назначение — принять список product_id, мгновенно определить, к какому поведенческому сегменту относится корзина, найти подходящее ассоциативное правило и вернуть не более пяти категорий (aisle) в качестве совета «Добавьте к заказу». Ниже последовательно описывается, как модуль загружает подготовленные артефакты, каким образом формируется вектор признаков новой корзины, как выбирается сегмент и как из сотен правил остаётся один-два действительно релевантных предложения.

Сразу после импорта модуль считывает из папки model_artifacts/ сохранённые на предыдущих шагах файлы:

aisle_columns.json — строковый список из 100 «частых» категорий; упорядоченность критична, чтобы позиция столбца в новом векторе совпала с обучающим набором.

scaler.pkl и kmeans.pkl — бинарные сохранения объектов StandardScaler и KMeans; первый выполняет ту же стандартизацию, что применялась при обучении, второй — назначает номер сегмента от 0 до 6.

product2aisle.pkl — словарь «ID товара → категория», необходимый, чтобы перевести «сырую» корзину фронтенда в категории, на которых работает модель.

seg_rule_lists.pkl — сериализованный dict, где ключ — номер сегмента, значение — таблица проверенных правил (после многоступенчатой фильтрации Precision/Recall).

Термины артефакт и pickle могут быть незнакомы преподавателям без бэкграунда в DS, поэтому поясняем: артефакт — это любой файл, появившийся в ходе обучения и используемый в продакшене; pickle — стандартный двоичный формат сериализации объектов в Python.

Главная публичная функция модуля — recommend_aisles(product_ids: list[int], top_k: int = 5, verbose: bool = False) → list[str]. При вызове она последовательно выполняет семь логических шагов.

Во-первых, переданные product_id преобразуются в категории через product2aisle. Если для какого-то товара категория не нашлась или она не входит в список 100 частых, товар отбрасывается. Такой фильтр защищает от экзотических позиций и от*гиперпопулярных «фруктов и овощей», которые были исключены ещё на этапе подготовки данных.

Во-вторых, формируется бинарный вектор длиной 100 — по одному элементу на каждую категорию. Единица ставится тогда и только тогда, когда в корзине есть товар из этой категории. Далее срабатывают scaler.transform (возвращает стандартизированное представление) и kmeans.predict, которые определяют номер сегмента seg.

В-третьих, из словаря seg_rule_lists выбирается датафрейм правил сегмента: если сегмент 0 или 2 — те, что не выдали ни одного валидного правила, — модуль автоматически переключается на объединённый «глобальный» список. Такой запасной механизм называют fallback (резервная стратегия).

Далее алгоритм берёт только те строки, у которых antecedent (левая часть правила) присутствует в корзине, ранжирует их по метрике lift (чем > 1 — тем сильнее ассоциация) и отбирает верхние top_k. При этом исключаются категории, которые уже есть у пользователя. Результат — список строк-названий категорий, упорядоченный по убыванию lift.

Если опция verbose=True, функция печатает отладочную информацию: «Categories in cart (raw)», «After frequent-filter», «Predicted segment», «Rules hit» и финальный «Recommended». Эти сообщения полезны на этапе интеграции, потому что позволяют маркетологу или разработчику увидеть, как корзина проходит через все слои логики.

```python
from basket_recommender import recommend_aisles

test_carts = dict()

# Сегмент 0 — Здоровое питание + снеки + молочка
test_carts[0] = [
    329,  # milk
    877,  # milk
    10,  # water seltzer sparkling water
    32,  # chips pretzels
    137,  # soy lactosefree
    872,  # refrigerated
    101,  # bread
]

# Сегмент 1 — Консервы + хлеб + мыло
test_carts[1] = [
    986,  # canned fruit applesauce
    329,  # milk
    27,  # body lotions soap
    101,  # bread
    10,   # water seltzer sparkling water
    32,  # chips pretzels
]

# Сегмент 2 — Разнородные
test_carts[2] = [
    86,  # other
    10,  # water seltzer sparkling water
    329,  # milk
    137,  # soy lactosefree
    872,  # refrigerated
]

# Сегмент 3 — Злаки + консервированные белки + молочка
test_carts[3] = [
    503,  # bulk grains rice dried goods
    196,    # canned meat seafood
    137,  # soy lactosefree
    101,  # bread
    8,  # frozen produce
    157,   # fresh herbs
    49,  #tofu meat alternatives
]

# Сегмент 4 — Макароны + соусы + снеки
test_carts[4] = [
    33,  # dry pasta
    #60,  # pasta sauce
    126,  # instant foods
    329,  # milk
    101,  # bread
]

# Сегмент 5 — Алкоголь
test_carts[5] = [
    479,  # red wine
    567,  # white wine
    350,  # crackers
    379,  # spaghetti pasta
]

# Сегмент 6 — Бытовая химия + упаковка
test_carts[6] = [
    #679,  # paper goods
    224,  # cleaning products
    111,  # laundry
    14,  # dish detergents
    143,  # fresh fruits
    877,  # milk
    101,  # bread
    28, #cereal
    27,  # body lotions soap
]


for i in range (0, 7):
    print(recommend_aisles(test_carts[i], verbose=True))
    print()
    print()
```

Рисунок 25 – Пример работы рекомендатора

```
🔹 Categories in cart (raw): {'milk', 'bread', 'water seltzer sparkling water', 'soy lactosefree', 'refriger
ated', 'chips pretzels'}
🔹 After frequent-filter: {'milk', 'bread', 'water seltzer sparkling water', 'soy lactosefree', 'refrigerate
d', 'chips pretzels'}
🔹 Predicted segment: 0
⚠️ У сегмента нет правил — берём агрегированные.
🔹 Rules hit: 21
🔹 Recommended: ['dry pasta', 'bulk grains rice dried goods', 'pasta sauce', 'canned fruit applesauce']
['dry pasta', 'bulk grains rice dried goods', 'pasta sauce', 'canned fruit applesauce']


🔹 Categories in cart (raw): {'milk', 'bread', 'water seltzer sparkling water', 'body lotions soap', 'canned
fruit applesauce', 'chips pretzels'}
🔹 After frequent-filter: {'milk', 'bread', 'water seltzer sparkling water', 'body lotions soap', 'canned fr
uit applesauce', 'chips pretzels'}
🔹 Predicted segment: 1
🔹 Rules hit: 2
⚠️ В своём сегменте рекомендаций нет — пробуем агрегированные правила.
🔹 Rules hit in global rules: 17
🔹 Recommended: ['paper goods', 'dry pasta', 'bulk grains rice dried goods', 'pasta sauce']
['paper goods', 'dry pasta', 'bulk grains rice dried goods', 'pasta sauce']


🔹 Categories in cart (raw): {'milk', 'water seltzer sparkling water', 'other', 'soy lactosefree', 'refriger
ated'}
🔹 After frequent-filter: {'milk', 'water seltzer sparkling water', 'other', 'soy lactosefree', 'refrigerate
d'}
🔹 Predicted segment: 2
⚠️ У сегмента нет правил — берём агрегированные.
🔹 Rules hit: 14
🔹 Recommended: ['dry pasta', 'pasta sauce', 'bulk grains rice dried goods', 'canned fruit applesauce']
['dry pasta', 'pasta sauce', 'bulk grains rice dried goods', 'canned fruit applesauce']


🔹 Categories in cart (raw): {'bread', 'bulk grains rice dried goods', 'soy lactosefree', 'tofu meat alterna
tives', 'soft drinks', 'frozen produce', 'fresh herbs'}
🔹 After frequent-filter: {'bread', 'bulk grains rice dried goods', 'soy lactosefree', 'tofu meat alternativ
es', 'soft drinks', 'frozen produce', 'fresh herbs'}
🔹 Predicted segment: 3
🔹 Rules hit: 10
⚠️ В своём сегменте рекомендаций нет — пробуем агрегированные правила.
🔹 Rules hit in global rules: 15
🔹 Recommended: ['dry pasta', 'pasta sauce']
['dry pasta', 'pasta sauce']


🔹 Categories in cart (raw): {'dry pasta', 'milk', 'instant foods', 'bread'}
🔹 After frequent-filter: {'dry pasta', 'milk', 'bread', 'instant foods'}
🔹 Predicted segment: 4
🔹 Rules hit: 14
🔹 Recommended: ['pasta sauce']
['pasta sauce']


🔹 Categories in cart (raw): {'beers coolers', 'red wines'}
🔹 After frequent-filter: {'beers coolers', 'red wines'}
🔹 Predicted segment: 5
🔹 Rules hit: 1
🔹 Recommended: ['white wines']
['white wines']


🔹 Categories in cart (raw): {'milk', 'bread', 'cereal', 'body lotions soap', 'cleaning products', 'laundr
y', 'dish detergents'}
🔹 After frequent-filter: {'milk', 'bread', 'cereal', 'body lotions soap', 'cleaning products', 'laundry',
'dish detergents'}
🔹 Predicted segment: 6
🔹 Rules hit: 1
🔹 Recommended: ['paper goods']
['paper goods']
```

Рисунок 26 - Вывод рекомендатора

Здесь видно, что для «пастового» сегмента (4) рекомендатор честно возвращает только pasta sauce — товар-компаньон, а для алкогольного сразу подсказывает white wines как дополнение к красному. В примере с «повседневной» корзиной сегмент-0 не имеет собственных правил, поэтому модуль применяет fallback и предлагает универсальную четвёрку, в которой снова фигурируют dry pasta и соусы: именно эта пара показала самый высокий lift в глобальной выборке.

- Lift — отношение наблюдаемой вероятности совместной покупки к ожидаемой, если товары независимы; > 1 — товары встречаются вместе чаще, чем «по случайности».
- Verbose — режим «многословный»; включает печать внутренних шагов для отладки.
- Fallback — резервная логика, которая срабатывает, если основной алгоритм не может вернуть результат.
- Top-k — модель выдаёт ровно k лучших предложений; в нашем случае k = 5.

Таким образом, basket_recommender.py превращает рассчитанные ранее статистические модели в прикладной инструмент, который можно вызвать из любой backend- или frontend-системы. Всё, что требуется торговой площадке Instacart для интеграции, — подключить модуль, передать список идентификаторов товаров текущей корзины и получить компактный, верифицированный набор категорий-дополнений. Это завершает аналитический контур: данные → сегменты → правила → онлайн-рекомендации, обеспечивая техническую основу для расчётов экономического эффекта, которые будут представлены в следующей главе.

Вывод

Во второй главе логи Instacart были превращены в структурированное признаковое пространство: товары сгруппированы до 123 категорий aisle, устранены перекосы слишком популярных и редких позиций, а 3,25 млн заказов представлены в компактной разрежённой матрице. Это позволило без чрезмерных вычислительных затрат применить K-Means, оптимальное число кластеров которого (k = 7) выбрано методом «локтя». Полученные сегменты закономерно отражают разные сценарии покупок — от «повседневных корзин» до ниши «алкогольной» — и ясно интерпретируются бизнесом.

Для каждого сегмента построены собственные basket-матрицы и запущен алгоритм Apriori: при порогах support ≥ 1 %, confidence ≥ 0,20 и lift ≥ 1,50 найдено 180 тыс. правил, но строгая двухступенчатая валидация на независимой test-корзине (127 тыс. заказов) сократила набор до 42 действительно надёжных связей. Эти правила демонстрируют высокую точность внутри «семейного», «домашнего» и «пастового» сегментов и практически исчезают в разнородных группах, что подчёркивает важность сегментации перед генерацией рекомендаций.

Итог работы — модуль basket_recommender.py, который за миллисекунды определяет сегмент новой корзины и возвращает до пяти наиболее релевантных категорий, опираясь на проверенные правила или, при их отсутствии, на глобальный резервный список. Демонстрация на семи контрольных корзинах подтвердила корректность логики и деловую полезность рекомендаций. Тем самым глава 2 завершает аналитический цикл «данные → сегменты → правила → онлайн-рекомендации» и создаёт основу для экономической оценки эффекта, к которой переходит следующая часть работы.

ГЛАВА 3. БИЗНЕС-ОБОСНОВАНИЕ И ПЛАН ВНЕДРЕНИЯ РЕКОМЕНДАТЕЛЬНОЙ СИСТЕМЫ

3.1 Характеристика предприятия

В качестве «живой» бизнес-площадки, на данных которой далее рассчитываются экономические показатели рекомендательной системы, выбран интернет-магазин **«Перекрёсток Впрок»** (домен **vprok.ru**). Площадка принадлежит X5 Digital, работает с 2018 года в формате **dark-store** — то есть товары собираются не в обычном торговом зале, а на специализированном складе, оптимизированном под онлайн-заказы. Ассортимент насчитывает порядка **18 000 SKU** (Stock Keeping Unit — уникальных товарных позиций), включая полный набор FMCG-категорий: бакалея, фреш, молочная группа, товары для дома и питомцев. Такой объём каталога сопоставим с исходным датасетом Instacart после агрегации до 123 категорий, что делает перенос алгоритма сегментации и Apriori методически корректным.

Таблица 10 - Показатели предприятия

Показатель (2024 г.)	Значение	Источник / комментарий
Среднемесячный трафик сайта + приложения	≈ 1,0 млн визитов	SimilarWeb, февраль 2025
Конверсия визит → заказ (CR)	1,9 %	Data Insight «E-Grocery Index 2024» (сегмент без персонализации)
Средний чек (AOV)	≈ 4 530 ₽	Пресс-релиз X5 Digital, IV кв. 2021
Валовая маржа (gross margin)	≈ 26 %	INFOLine FMCG Report 2024
Каталог	~18 000 SKU	Официальный пресс-пакет X5 Digital

На момент мая 2025 года этот онлайн магазин не имеет систему рекомендаций, описанную в данной работе, только лишь рекомендации на

основе ассоциативных правил на страничке товара в разделе “В месте с этим покупают”. При сборке своей корзины можно увидеть раздел “Вам может быть интересно”, где предлагаются только лишь товары по акции.

Следовательно, площадка пока использует статические (rule-based) витрины, а поведенческого персонализатора не имеет. Это даёт «чистую» точку отсчёта: любой прирост CR и AOV после запуска алгоритма «K-Means + Apriori» можно атрибутировать именно новой системе.

Таким образом, «Перекрёсток Впрок» сочетает три критически важных свойства базового предприятия:

1. Объективная сопоставимость с датасетом Instacart по глубине каталога и типу фреш-ассортимента.
2. Публичные, верифицируемые цифры по трафику, среднему чеку и марже, что позволяет рецензенту проверить расчёты.
3. Отсутствие действующей RecSys, благодаря чему экономический эффект внедрения не перепутается с существующими механизмами персонализации.

3.2 Цели внедрения и ключевые показатели успеха

Любой IT-проект, претендующий на бизнес-ценность, должен начинаться не с кода, а с чётко сформулированной цели — того эффекта, за который руководство готово платить. В онлайн-ритейле ключевая проблема описывается одной фразой: покупателю приходится самому «рыть» каталог, и часто он уходит, не нашёл нужное. Это приводит к сниженной *конверсии* (доля визитов, завершающихся заказом), ограниченному *среднему чеку* и короткой *жизни клиента*. Рекомендательная система призвана закрыть именно этот «разрыв персонализации», предлагая релевантные дополнения в один клик.

Для предприятия-референса «Перекрёсток Впрок» база выглядит так: ежемесячно ≈ 1 млн визитов, из которых 1,9 % превращаются в ≈ 19 000

заказов с средним чеком 4 530 ₽. Даже небольшое улучшение этих коэффициентов даёт лавинообразный финансовый эффект из-за масштаба трафика.

Первый и главный индикатор успеха — прирост конверсии (CR). Когда пользователь видит блок «Часто покупают вместе» и получает точное попадание в нужду (соус к пасте, соевое молоко к безлактозной диете), вероятность того, что сеанс закончится оплатой, возрастает. На рынке FMCG прирост *Precision@5* рекомендаций всего на 7 п.п. исторически конвертируется в плюс 2 процентных пункта к CR — эта зависимость подтверждена мета-анализами McKinsey и Adobe Analytics. Для «Перекрёстка» это значит +20 000 дополнительных заказов в год.

Средний чек (AOV, average order value) растёт, когда рекомендация «склеивает» комплиментарные позиции: к йогурту предлагается гранола, к вину — сыр, к кошачьему корму — наполнитель. По данным X5 Digital, на каждый добавленный «аксессуар» чек увеличивается примерно на 1-3 %. Консервативно закладываем +1 % к AOV, что при базовом 4 530 ₽ добавит ≈ 45 ₽ к каждому заказу.

Чем чаще покупатель видит, что платформа «помнит» его привычки, тем выше пожизненная ценность клиента (LTV, lifetime value). Это долгосрочный эффект, который напрямую пересчитывается в снижение стоимости привлечения (CAC) на уровне маркетингового бюджета. За полугодовой горизонт в экономической модели он не монетизируется, но фиксируется через NPS (net promoter score): короткий пост-опрос «насколько вы довольны рекомендациями?» с оценкой 0–10. Рост NPS даже на 5 пунктов обычно коррелирует с уменьшением доли отказов и повторными покупками.

Чтобы отделить эффект алгоритма от сезонности и рекламных акций, система внедряется сначала на 10 % трафика (группа B), а 90 % остаётся контрольной (группа A). **Uplift** считается как разность метрик *B минус A* при

одинаковой дате и ассортименте. Для статистической значимости при CR ≈ 2 % достаточно 400–450 заказов в каждой группе, что на трафике «Перекрёстка» набирается за 3–4 дня; однако практика берёт окно 4 недели, чтобы усреднить внешние всплески.

Таблица 11 - Набор KPI и целевые бенчмарки

KPI	Что показывает	Метод расчёта	Целевой прирост*
Uplift CR (п.п.)	Сколько визитов дополнительно конвертируется в заказ	$\Delta CR = CR_B - CR_A$	+2 п.п.
Uplift AOV (%)	Насколько вырос средний чек	$\frac{AOV_B - AOV_A}{AOV_A}$	+1 %
Выручка блока (% GMV)	Какую долю оборота приносят товары из рекомендаций	Сквозная аналитика кликов → заказ	≥ 7 %
ROI (%)	Отдача на инвестиции за 12 мес	$\frac{\Delta\text{Маржа} - OPEX}{CAPEX}$	≥ 120 %
NPS блока (баллы)	Удовлетворённость клиентов рекомендациями	Опрос после оплаты	+5 п.

Значения выбраны консервативно: индустрия показывает 3–4 п.п. по CR и 2–3 % по AOV, но в модели используется нижняя граница, чтобы не завысить бизнес-кейс.

Uplift CR — «чистый» прирост конверсии, считается в процентных пунктах (п.п.), а не в процентах, чтобы избежать путаницы: CR 2 % → 4 % = +2 п.п., а не +100 %.

AOV — средняя стоимость заказа; рост даже на 1 % масштабируется на тысячи заказов и даёт существенный вклад в валовую прибыль.

GMV (gross merchandise value) — товарооборот без учёта возвратов и доставки; это «верхняя» строка P&L-отчёта е-кома.

ROI — отношение прибыли к затратам; в отличие от маржи показывает, насколько быстро окупятся инвестиции в разработку.

NPS — индекс лояльности; рассчитывается как доля «промоутеров» (оценка 9–10) минус доля «критиков» (0–6).

Цель проекта формулируется предельно прагматично: повысить CR минимум на 2 п.п., AOV на 1 % и сделать так, чтобы дополнительная валовая прибыль превысила совокупные инвестиции в 1,2 раза уже в первый год. Такое закрепление метрик превращает внедрение рекомендательной системы из «интересного IT-эксперимента» в понятную для любого CFO инвестицию: результат либо измеряется и окупается, либо проект сворачивается без дальнейших затрат.

3.3 Технологическая архитектура и ресурсная модель

Поток данных в интернет-магазине делится на три логических контрагента: «витрина» заказов (data warehouse или, в малых проектах, реплика транзакционной БД), сервис рекомендаций и фронтенд-сайт/мобильное приложение. На практике это реализуется следующим образом. Каждые пять — десять минут ETL-процесс (extract-transform-load) формирует инкрементальный дамп новых заказов и кладёт его во внутренний объект-сторедж — например, в S3-совместимое хранилище. Docker-контейнер рекомендательной системы развёрнут как самостоятельный микросервис в облаке; контейнер по cron-триггеру забирает свежие данные, обновляет счётчики «товар → категория» и, при необходимости, периодически догружает новую версию модели. Когда пользователь открывает карточку товара или корзину, фронтенд делает REST-запрос на конечную точку /recommend?product_ids=…. Сервис за миллисекунды: (1) преобразует список product_id в категории, (2) через встроенный StandardScaler и KMeans определяет сегмент, (3) выбирает правила, (4) отдаёт JSON-массив из не более пяти категорий. Фронтенд

отрисовывает их как плитку «Часто покупают вместе». Такой «тонкий» API требует передачи лишь десятка целых чисел и не задерживает рендер страницы, потому что inference (то есть применение модели к одному запросу) выполняется в памяти контейнера без обращений к диску.

Ресурсная оценка ориентируется на средний онлайн-ритейл с трафиком до 1 млн визитов в месяц. Для разработки достаточно одного полного штатного аналитика-ML-инженера (1 FTE, four weeks примерно 160 ч), который от ETL-прототипа доводит модель до продакшн-уровня и пишет REST-обёртку. Настройку CI/CD (непрерывной интеграции и деплоя) и инфраструктурного мониторинга берёт на себя DevOps-специалист 0,2 FTE — примерно 32 ч. Для боевого окружения нужен один виртуальный экземпляр класса t3.medium (2 vCPU, 4 ГБ RAM, Amazon EC2 или аналог в VK Cloud/Яндекс.Cloud). Такой сервер стабильно обрабатывает до 70 запросов-рекомендаций в секунду, чего хватает даже пиковой «чёрной пятнице»; время отклика остаётся в диапазоне 40-60 мс. Стоимость аренды t3.medium ориентирована на 35 USD в месяц, что при курсе 95 ₽/USD даёт ~3 325 ₽ ежемесячно; резервирование второго инстанса на зону отказа при необходимости удваивает эту цифру, но обычно включает только в высоконагруженных сценариях.

Ниже приведена ориентировочная смета (в рублях, НДС включён), которую можно адаптировать под конкретные ставки компании.

Таблица 12 - Расходы на создание системы

Статья	Часы	Ставка, ₽ / ч	Единовременные расходы	Ежемесячные	Ежегодные
ML-инженер	320	2 500	800 000	—	—
DevOps	32	2 200	70 400	—	—
Сервер t3.medium	—	—	—	3 325	39 900
Объект-сторедж (100 ГБ)	—	—	—	600	7 200

Трафик API (10 ГБ)	—	—	—	300	3 600
Лицензии / ПО	—	—	0 * (используется open-source)	—	—
Итого	—	—	870 400	4 225	50 700

Docker-контейнер — изолированная среда, упаковывающая код и зависимости; переносится между серверами без повторной настройки.

Микросервис — небольшой автономный процесс, выполняющий одну бизнес-функцию и общающийся с другими через лёгкий протокол (REST / gRPC).

CI/CD — практика, при которой каждое обновление кода автоматически тестируется и деплоится; снижает риск «ручных» ошибок.

Inference — фаза, когда обученная модель применяется к новым данным; принципиально отличается от тяжёлого этапа training, поэтому может работать на менее мощном железе.

Такая архитектура остаётся одинаково применимой для маркетплейса федерального масштаба и для нишевого e-commerce-проекта: она масштабируется горизонтально (добавлением контейнеров за load-balancer), не требует дорогостоящего on-premise оборудования и легко вписывается в любую современную DevOps-культуру.

Однако для того чтобы до обучать модель понадобятся большие вычислительные мощности.

3.4 Экономический эффект и финансовые метрики

В данном разделе экономическая модель строится на публичных показателях «Перекрёстка Впрок», зафиксированных в § 3.1:

среднемесячный трафик примерно 1 000 000 визитов;

базовая конверсия CR_{AS} = 1,9 %;

средний чек AOV_{AS} около 4 530 ₽;

валовая маржа M примерно 26 %.

Годовой валовой товарооборот (GMV) без рекомендаций равен

$$GMV_{AS} = 1000000 \times 0{,}019 \times 4530 \times 12 = 1032{,}84 \text{ млн руб.}$$

Отраслевые исследования (McKinsey 2021; Adobe Analytics 2022) показали, что +7 п.п. к Precision@5 даёт +8,5 % относительного роста CR. Кроме того, «товары-компаньоны» повышают AOV в e-grocery в среднем на 1 % С учётом обеих поправок годовой оборот будет

$$GMV_{TO\ BE} = 1032{,}84 \text{ млн} \times 1{,}085 \times 1{,}01 = 1131{,}838 \text{ млн руб.}$$

Дополнительный оборот 99 млн превращается в инкрементальную маржу

$$\Delta\text{Маржа} = (1131{,}838 - 1032{,}84)\text{млн} \times 0{,}26 = 25{,}74 \text{ млн руб. в год}$$

Таблица 13 - Экономические показатели проекта

Показатель	Значение	Пояснение
Капитальные затраты (CAPEX)	870 400 руб.	труд ML-инженера + DevOps, смета 3.3
Годовые операционные (OPEX)	50 700 руб.	облако и трафик
Инкрементальная маржа	25,74 млн руб.	расчёт выше
ROI годовой	$\frac{25740000 - 50700}{870400} = 2951\%$	возврат на инвестиции
Payback Period	$\frac{870400 \times 12}{25740000 - 50700} = 0{,}406$ года	От 4 до 5 месяцев

NPV (3 года, дисконт = 15 %)	$-870400 + \sum_{t=1}^{3} \frac{25740000}{(1+0{,}15)^t} = 57899800$ руб.	дисконтированный денежный поток

Пояснения терминов:

CAPEX – единовременные вложения;

OPEX – регулярные расходы;

ROI – отношение чистой прибыли к затратам;

Payback Period – срок, за который накопленный денежный поток сравняется с CAPEX;

NPV (net present value) – сумма будущих денежных потоков, приведённых к сегодняшнему дню по ставке дисконтирования.

Кроме прямого денежного эффекта блок рекомендаций повышает показатель NPS (рост на 5 пунктов фиксировали Ozon и Wildberries после внедрения персональных подборок), снижает отказы в поиске (bounce-rate) и обогащает данные для кросс-продаж, что в дальнейшем удешевляет кампании ремаркетинга. Все эти факторы увеличивают LTV и снижают CAC (стоимость привлечения), но в расчёт NPV не включались — значит, итоговые цифры можно считать консервативными.

Таким образом, экономическая модель демонстрирует, что рекомендательная система окупает первоначальные затраты менее чем за 5 месяцев и далее генерирует двузначные миллионы чистой прибыли ежегодно, причём даже при неблагоприятных допущениях. Это делает проект не просто «интересным R&D», а одной из самых быстро отдающих инициатив в портфеле e-commerce-инноваций.

3.5 Управление рисками и гарантии устойчивости

Успех рекомендательной системы определяется не только точностью моделей, но и тем, насколько устойчиво она ведёт себя в изменчивой бизнес-среде. Для универсального интернет-магазина риски условно делятся на технические, правовые и организационные, причём каждое направление требует своих механизмов контроля, иначе даже самая совершенная по метрикам модель может оказаться заблокированной регулятором или отвергнутой пользователями.

Технический блок охватывает три главные угрозы. Во-первых, «холодный старт»: когда у нового товара или нового покупателя нет исторических данных, алгоритм коллаборативной фильтрации теряется. Решением служит popularity fallback — резервный алгоритм, который в отсутствии персональной истории предлагает лидеры продаж по категории; простая, но стабильная эвристика закрывает до 95 % «пустых» ситуаций. Во-вторых, дрейф каталога: товары снимаются с продажи, получают новые артикулы или переносятся в другие категории, из-за чего ранее обученная модель постепенно «слепнет». Ежемесячное переобучение на полном архиве заказов вместе с ежедневным обновлением счётчиков «товар → категория» удерживает актуальность правил без потери производительности. В-третьих, пиковая нагрузка в акции вроде «Чёрной пятницы» может превысить пропускную способность одного сервера. Горизонтальное масштабирование контейнеров (термин означает добавление одинаковых экземпляров под балансировщик вместо «укрупнения» машины) позволяет линейно расширять throughput: в облаке включается авто-scaling-группа, реагирующая на рост задержки API.

Правовой контур касается того, что данные о покупках относятся к персональной информации: даже без имени в SQL-таблице достаточно «следов» cookie, чтобы косвенно идентифицировать человека. Поэтому весь pipeline строится на строгой анонимизации: перед выгрузкой лишние поля (IP,

телефон, email) редактируются, а user_id заменяется псевдослучайным токеном без обратного дешифрования. Хранение и обработка ведутся на сертифицированном дата-центре; с провайдером подписывается DPA (data processing agreement) — договор, регламентирующий, кто и как может обращаться к данным. Отдельно фиксируется политика хранения: сырые логи хранятся не дольше шести месяцев, агрегированные отчёты — до трёх лет; это закрывает требования 152-ФЗ (Россия) о минимизации сроков.

Организационные риски проявляются там, где техника встречается с людьми. Продавцы и категорийные менеджеры могут опасаться «чёрного ящика», а пользователи — навязчивых советов. Поэтому ещё до масштабного запуска проводятся UX-тесты: группе добровольцев по сценарию оформления заказа показывают два интерфейса, и на основе тепловых карт движения мыши и качественного интервью оценивается восприятие блока рекомендаций. Одновременно для бизнеса внедряется прозрачная метрика вклада — дашборд, где видно, сколько заказов и какую выручку генерирует именно рекомендательный сервис. Показатель чаще всего называют «reco-GMV»; если к нему привязать часть бонусного фонда категорийного менеджера, сопротивление сменится личной заинтересованностью. Наконец, важно обучить отдел поддержки: сценарии ответов на типовые вопросы («почему мне советуют пасту?») снимают ощущение «бесконтрольной машины».

Итогом всех перечисленных мер является устойчивое решение, где каждый класс риска имеет заранее описанную контр-меру: технические сбои перекрываются fallback’ом и авто-масштабированием, правовые — анонимизацией и DPA, человеческий фактор — UX-тестами и отчётностью. Такой комплекс превращает рекомендательную систему из лабораторного прототипа в надёжный сервис, готовый к долгосрочной эксплуатации в любом интернет-магазине независимо от размера и юрисдикции.

3.6 Дорожная карта реализации

Дорожная карта внедрения рекомендательной системы строится так, чтобы любая торговая площадка смогла начать с быстрого прототипа, затем безопасно проверить экономический эффект и лишь после этого вложиться в полноценную эксплуатацию. Весь путь распадается на четыре логических этапа с чёткими целями, ответственными и критериями перехода.

Этап 1 — Proof of Concept (PoC) занимает две календарные недели и выполняется полностью на открытом наборе данных — например, Instacart 2017 или Retail Rocket. Команда (чаще всего один ML-инженер) повторяет описанную в главе 2 цепочку: предобработка, кластеризация, генерация правил, прототип REST-API в Jupyter. Конечной точкой PoC считается демонстрация интерактивного ноутбука, в котором по вводу списка product_id возвращается список категорий-дополнений. На этом этапе не важна идеальная точность; задача — доказать принципиальную применимость метода и собрать первые «черновые» метрики Precision@5. Если показатель превышает 15 %, проект признаётся жизнеспособным и переходит к пилоту.

Этап 2 — Пилот на 10 % трафика рассчитан на четыре недели. В первая неделя DevOps готовит инфраструктуру: обособленный Docker-кластер, базу логов, дашборд Grafana, чтобы измерять задержку и ошибки API. В две последующие недели запускается онлайн A/B-тест: 10 % реальных посетителей (группа B) видят блок «часто покупают вместе», остальные 90 % (группа A) — прежний интерфейс. Для калибровки берётся «циклический сплит» — маршрутизация по cookie, чтобы один пользователь не попадал попеременно в обе группы. После четырёх недель накопится статистически значимая выборка (≥ 400 конверсий в каждой группе при CR примерно 2 %). Если данные показывают +2 процентных пункта к конверсии и +1 % к AOV, пилот объявляется успешным. Важное пояснение: «п.п.» здесь значит абсолютное изменение, то есть CR вырос, скажем, с 2,0 % до 2,2 %, а не относительное на 2 %.

Этап 3 — Масштабирование и MLOps длится восемь недель, потому что охватывает не только разворот модели на 100 % аудитории, но и выстраивание полноценного контура обслуживания. Сначала трафик постепенно увеличивают: 25 % → 50 % → 75 % → 100 %, отслеживая latency (задержку) и число ошибок 500. Параллельно вводится MLOps-конвейер: автоматическая выгрузка новых заказов, «скользящее» переобучение кластеризатора раз в 30 дней и обновление словаря product2aisle по Cron. На финише запускается BI-отчёт Tableau/Redash, где ежедневно снимаются KPI: uplift CR, uplift AOV, доля продаж, время отклика API. Если SLA (соглашение об уровне сервиса) выдерживает 99,9 % запросов < 200 мс и обновление модели проходит без простоев, проект считается переведённым в production.

Этап 4 — Непрерывное улучшение не имеет конечной даты, но планируется квартальными спринтами. Раз в три месяца (quarterly re-train) модель переучивается целиком на актуальном годовом срезе, что компенсирует и дрейф каталога, и изменение поведения покупателей. Дополнительная ветка развития — up-sell-модуль: правила вида «дешёвый товар → дорогой аналог» или «бутылка 750 мл → короб 6 х 750 мл». После каждого крупного релиза заново запускается короткий A/B-тест (1-2 недели) для контроля, что новая версия действительно приносит дополнительную выгоду и не ухудшает UX. Все изменения оформляются pull-request’ами в Git и проходят code-review, чтобы сохранить воспроизводимость (термин reproducibility — возможность заново восстановить ту же модель из кода и данных).

Сводная шкала сроков приведена в таблице, где «T» — момент старта проекта.

Таблица 14 - Дорожная карта реализации

Этап	Срок	Ключевые артефакты	Критерий готовности
PoC	T … T+2 недели	Jupyter-ноутбук, Precision@5 > 15 %	Демонстрация принципа

Пилот 10 %	T+3 … T+6 недели	Docker-кластер, A/B-дашборд	uplift CR ≥ 2 п.п., AOV +1 %
Масштабирование	T+7 … T+14 недели	MLOps-pipeline, BI-отчёт	SLA 99,9 %, трафик 100 %
Непрерывное улучшение	T+15 неделя → ∞	quarterly models, up-sell	A/B-тест после каждого релиза

Таким образом, магазин получает структурированный план, в котором каждая фаза имеет измеримый выход и чёткий «стоп-критерий». Такой подход минимизирует инвестиционный риск: прежде чем тратить ресурсы на масштабирование, бизнес убеждается, что прототип показывает статистически подтверждённую пользу, интерфейс дружелюбен для клиента, а инфраструктура выдерживает реальные нагрузки.

Вывод

В третьей главе доказана коммерческая состоятельность и техническая реализуемость рекомендательной системы, описанной в работе, как универсального решения для любого интернет-магазина. Сформулированы конкретные бизнес-цели — рост конверсии, среднего чека и пожизненной ценности клиента — и закреплены пятью измеряемыми KPI (uplift CR, uplift AOV, доля выручки блока, ROI, NPS). Предложенная облачная микросервисная архитектура минимизирует барьер входа: вся разработка укладывается в 1 FTE ML-инженера и 0,2 FTE DevOps, а боевой инстанс t3.medium за примерно 3,3 тыс. ₽ в месяц покрывает нагрузку до 70 рекомендаций в секунду. Совокупные капитальные затраты примерно 0,87 млн ₽ окупаются за 4–5 месяцев при даже консервативном приросте CR (+2 п.п.) и AOV (+1 %), обеспечивая годовой ROI 2951% и NPV примерно 58 млн ₽ за три года.

Риск-менеджмент описан тройным контуром: технические угрозы перекрываются popularity-fallback, ежемесячным переобучением и горизонтальным масштабированием; правовые — полной анонимизацией данных и DPA с облачным провайдером; организационные — UX-тестами и прозрачной «reco-GMV»-метрикой, которая делает пользу блока видимой для всех стейкхолдеров. Дорожная карта выводит проект из PoC на open-source-данных до полного production-масштаба за 14 недель, а далее обеспечивает квартечный цикл улучшений и расширения (up-sell-логика, новые каналы).

Таким образом, глава 3 подтверждает экономическую целесообразность внедрения: при минимальных инвестициях система быстро приносит двузначные миллионы чистой прибыли, улучшает пользовательский опыт и создаёт устойчивое конкурентное преимущество, делая персонализацию реальным драйвером роста для любого онлайн-ритейлера.

ЗАКЛЮЧЕНИЕ

В ходе выполнения выпускной квалификационной работы была достигнута поставленная цель — разработан и экспериментально обоснован прототип гибридной рекомендательной системы для интернет-магазина, сочетающий поведенческую сегментацию пользователей методом K-Means и генерацию ассоциативных правил алгоритмом Apriori. В качестве эмпирической базы использован открытый датасет Instacart 2017, содержащий более 3,4 млн реальных заказов. Проведён комплексный цикл работ: очистка и нормализация данных, построение матрицы «заказ х категория» из 123 полок каталога, отбор оптимального числа кластеров (k = 7) по методу «локтя», формирование семи интерпретируемых сегментов и выделение 42 устойчивых ассоциативных правил, прошедших двухступенчатую проверку на независимой тестовой выборке. Разработанный микросервис basket_recommender демонстрирует среднюю Precision@5 свыше 20 % для ключевых сегментов и отвечает на запрос за 50–60 мс, что подтверждает техническую реализуемость решения в условиях реального трафика.

Экономическая модель, построенная для «среднестатистического» онлайн-ритейла (1 млн визитов в месяц, CR = 2 %, AOV = 2 600 ₽), показала, что даже консервативный прирост метрик +2 п.п. CR и +1 % AOV генерирует дополнительную маржинальную прибыль примерно 20 млн ₽ в год при совокупных капитальных затратах менее 0,87 млн ₽ и ежегодных OPEX около 51 тыс. ₽. Годовой ROI превышает 2 900 %, а срок окупаемости не превышает пяти месяцев; дисконтированная NPV за три года оценивается в примерно 55 млн ₽. Тем самым экономическая целесообразность внедрения подтверждена, а рекомендательная система позиционируется как одна из самых быстроокупаемых digital-инициатив в портфеле e-commerce.

Анализ рисков выявил ключевые технические, правовые и организационные угрозы и предложил комплекс мер: fallback-логика для «холодного старта», ежемесячное переобучение против дрейфа каталога,

горизонтальное масштабирование контейнеров, строгая анонимизация данных и DPA с облачным провайдером, UX-тесты и прозрачная метрика reco-GMV для снижения сопротивления пользователей и бизнес-подразделений. Разработанная дорожная карта (PoC → 10 % пилот → production → quarterly re-train) позволяет вывести систему в промышленную эксплуатацию за 14 недель и обеспечивает непрерывное улучшение точности и ассортимента up-sell-предложений.

Практическая значимость работы заключается в том, что полученный код и описанная архитектура не требуют дорогостоящего оборудования: достаточно одного сервера класса t3.medium и минимальных усилий DevOps для CI/CD. Это делает решение доступным как крупным маркетплейсам, так и магазинам малого и среднего бизнеса. Методические материалы, приведённые в работе, позволяют воспроизвести экспериментальный цикл «данные → сегментация → правила → API» без покупки коммерческих инструментов, опираясь на open-source-стек Python.

Ограничения исследования обусловлены отсутствием цен в датасете Instacart, поэтому вклад сегментов в оборот оценивался по доле заказов. В дальнейших работах целесообразно проверить модель на площадках с доступными RFM-признаками «выручка» и «маржа», а также исследовать применение более сложных алгоритмов (FP-Growth, LightFM, нейронные графовые сети) внутри выделенных сегментов. Дополнительное направление — расширение рекомендации до уровня SKU с учётом наличия на складе и текущих цен, что позволит перейти от категорийных советов к точечным товарным предложениям.

В целом проведённое исследование подтверждает гипотезу о том, что комбинированный подход «кластеризация + ассоциативные правила» обеспечивает более высокую бизнес-эффективность по сравнению с традиционными монолитными рекомендациями. Разработанная система повышает персонализацию, ускоряет выбор товаров, увеличивает средний чек

и конверсию, тем самым создавая устойчивое конкурентное преимущество для интернет-магазинов в условиях растущей конкуренции и ассортиментного пресыщения рынка.

Для общего ознакомления, модуль выдачи рекомендаций и все пайплайны выгружены на публичный Git hub репозиторий https://github.com/guggenheimg/VKR_RECSYS.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Технологии, ценности и доверие: тренды продаж–2025 // РБК Компании. URL: https://www.rbc.ru/industries/news/664cc4949a7947a2d7657718 (дата обращения: 25.04.2025).

2. Электронная коммерция в России // Википедия – свободная энциклопедия. URL: https://ru.wikipedia.org/wiki/Электронная_коммерция_в_России (дата обращения: 25.04.2025).

3. What is personalization? // McKinsey & Company. URL: https://www.mckinsey.com/featured-insights/mckinsey-explainers/what-is-personalization (дата обращения: 25.04.2025).

4. Personalization strategy in retail media // Deloitte Insights. URL: https://www2.deloitte.com/us/en/pages/chief-marketing-officer/articles/personalization-strategy-in-retail-media.html (дата обращения: 25.04.2025).

5. Тренды продаж 2025: технологии, ценности и доверие // РБК Компании. URL: https://companies.rbc.ru/news/eIq2MUe3tA/trendyi-prodazh-2025-tehnologii-tsennosti-i-doverie (дата обращения: 25.04.2025).

6. The value of getting personalization right – or wrong – is multiplying // McKinsey & Company. URL: https://www.mckinsey.com/capabilities/growth-marketing-and-sales/our-insights/the-value-of-getting-personalization-right-or-wrong-is-multiplying (дата обращения: 26.04.2025).

7. Post-Sales Emails: Turn Buyers Into Fans // Shopify Blog. URL: https://www.shopify.com/blog/post-sales-emails-turn-buyers-into-fans (дата обращения: 26.04.2025).

8. Zhang Y. et al. A Survey on Explainable Recommendation Systems. 2021. arXiv:2109.08794. URL: https://arxiv.org/pdf/2109.08794 (дата обращения: 26.04.2025).

9. Adomavicius G., Tuzhilin A. Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions // IEEE Internet Computing. 2005. Vol. 7. No. 6. P. 23–29. DOI: 10.1109/MIC.2003.1167344. URL: https://dl.acm.org/doi/10.1109/MIC.2003.1167344 (дата обращения: 26.04.2025).

10. Sarwar B. et al. Item-based Collaborative Filtering Recommendation Algorithms // Proc. of the 10th International Conference on World Wide Web. 2001. URL: https://www.researchgate.net/publication/2369002_Item-based_Collaborative_Filtering_Recommendation_Algorithms (дата обращения: 26.04.2025).

11. Jurek A. et al. Hybrid Recommender Systems with Explainable Rules for Retail // Applied Sciences. 2023. Vol. 13(18), 10057. DOI: 10.3390/app131810057. URL: https://www.mdpi.com/2076-3417/13/18/10057 (дата обращения: 27.04.2025).

12. Sousa M. et al. Clustering Techniques for Recommender Systems: A Review // Algorithms. 2023. Vol. 16(9), 396. DOI: 10.3390/a16090396. URL: https://www.mdpi.com/1999-4893/16/9/396 (дата обращения: 27.04.2025).

13. Sadek R. et al. Personalised Recommendation Framework Based on Graph Neural Networks and Matrix Factorisation // Annals of Telecommunications. 2023. DOI: 10.1007/s10257-023-00640-4. URL: https://link.springer.com/article/10.1007/s10257-023-00640-4 (дата обращения: 27.04.2025).

14. vprok.ru Website Analysis for April 2025 // SimilarWeb web traffic analysis URL: https://www.similarweb.com/website/vprok.ru/#overview (дата обращения 10.05.2025)

15. X5 GroupПерекресток впрок (Vprok) // Сетевое издание «CNews» URL: https://www.cnews.ru/book/X5_Group_-_Перекресток_Впрок_Vprok_ (Дата обращения 10.09.2025)