

ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ
ФГАОУ ВО НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

Факультет компьютерных наук

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

Исследовательский проект на тему:

Сжатие словарей для нейросетевого анализа исходных кодов программ

Выполнил студент:

группы #БПМИ171, 4 курса Гусев Андрей Алексеевич

Принял руководитель ВКР:

Чиркова Надежда Александровна
Научный сотрудник
Факультет компьютерных наук НИУ ВШЭ

Москва 2024

Содержание

Аннотация	3
1 Данные	7
1.1 Этап 1	7
1.2 Этап 2	9
2 Декомпозиция задачи	16
2.1 Задача Image Captioning	16
2.2 Обзор классификаторов изображения	18
2.2.1 ResNet	19
2.2.2 Inception	19
2.2.3 MobileNet	19
2.2.4 MobileNet	21
2.3 Используемые метрики в задаче классификации	21
2.4 Проектирование архитектуры программного продукта (в процессе)	22
А Приложение 1	24

Аннотация

Ваша аннотация на русском языке.

Ключевые слова

Глубинное обучение, разреживание моделей, рекуррентные нейронные сети

Введение

В современном мире маркетплейсы стали неотъемлемой частью электронной коммерции, предоставляя платформы для продажи товаров и услуг различным производителям и ритейлерам. В 2023 году маркетплейсы продолжили быть главной движущей силой российской онлайн-торговли. Рост объема трат на маркетплейсах в 1,5 раза по сравнению с предыдущим годом свидетельствует о том, что интерес потребителей к онлайн-покупкам только укрепляется. На это влияют общерыночные факторы: продолжают развиваться альтернативные каналы поставок продукции ушедших брендов, улучшаются условия доставки, повышается удобство пользования платформами, расширяется сеть пунктов выдачи.

Согласно исследованиям Tinkoff Ecommerce, количество транзакций на маркетплейсах за год выросло на 63% (см. рисунок 0.1). Лидерами по росту количества покупок стали Мегамаркет (число транзакций выросло в 4,3 раза), Wildberries (в 2 раза) и Ozon (в 1,6 раза).

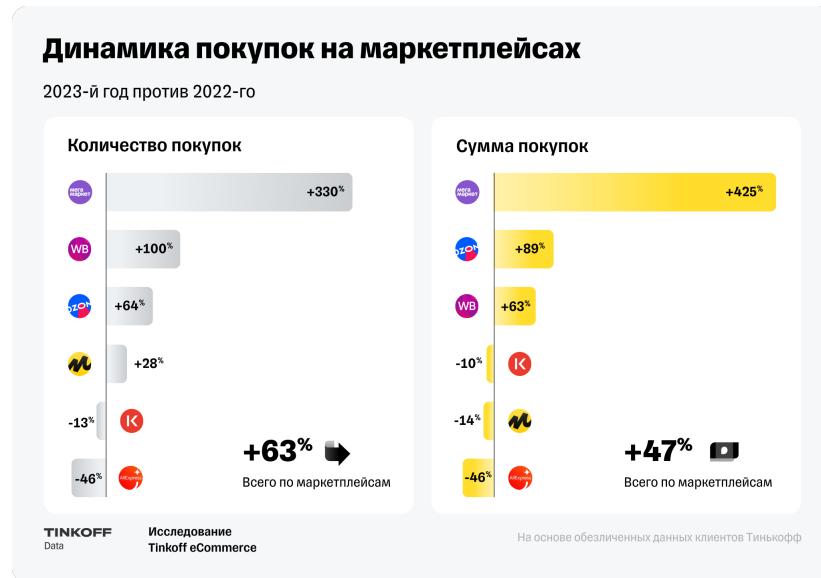


Рис. 0.1: Динамика покупок на маркетплейсах в регионах.

Появился тренд на рост популярности маркетплейсов в российских регионах. В 2023 году жители российских городов стали значительно активнее совершать покупки на маркетплейсах: выросло как количество транзакций на онлайн-площадках, так и их сумма (см. рисунок 0.2). По количеству совершенных транзакций особенно заметен рост в таких городах, как Омск (+91%), Красноярск (+88%), Новосибирск (+79%), Челябинск (+79%) и Волгоград (+75%). В Москве зафиксирован наименьший прирост числа транзакций (+41%). Увеличение интереса жителей регионов к маркетплейсам объясняется рядом причин: расширением географии присутствия площадок, развитием сетей ПВЗ и логистических сервисов,

улучшением условий доставки.



Рис. 0.2: Динамика покупок на маркетплейсах в регионах.

Вместе с тем выросло количество селлеров на 8%. Рынок становится более зрелым: место неопытных продавцов занимают более профессиональные. Они ведут бизнес более уверенно, укрепляют свои позиции на площадках и торгуют на нескольких платформах одновременно. Количество селлеров, ведущих торговлю на двух и более маркетплейсах, за год увеличилось на 17%. Самой привлекательной платформой для старта бизнеса является Wildberries: 63% продавцов в конце 2023 года выбирали ее в качестве первой площадки (см. рисунок 0.3).

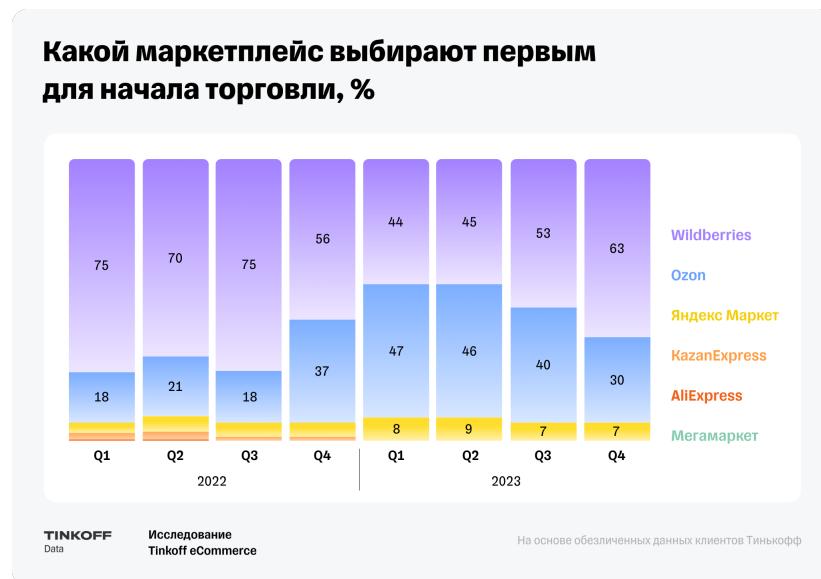


Рис. 0.3: Популярность маркетплейсов за 2023 год.

С увеличением конкуренции продавцы все чаще обращаются к системам, которые могут помочь им в продажах, а также автоматизировать процесс работы. Одним из ключевых

факторов успешной продажи становится эффективная SEO-оптимизация карточек товаров. Подбор наиболее подходящей категории и создание продаваемого описания, содержащего ключевые слова, позволяет улучшить видимость товаров в результатах поиска как на самом маркетплейсе, так и в поисковых системах, что напрямую влияет на увеличение продаж.

На данный момент существуют несколько сервисов (например, TurboTextPro, CopyMonkey, Gerwin), позволяющих сгенерировать описание товаров по характеристикам, ключевым словам или фото. Однако, качество сгенерированных описаний не всегда позволяют использовать их в системах автономного управления. В настоящее время российский рынок не предлагает специальных технологий и решений для подбора наиболее подходящей категории товара на маркетплейсе. Таким образом, задача создания качественного инструмента для эффективной SEO-оптимизации карточек товаров является актуальной.

Цель данной работы — разработать сервис, в основе которого будет реализован функционал, способный классифицировать товары на маркетплейсе на основе их фотографий и генерировать соответствующие к ним описания. Сервис реализуется для маркетплейса Wildberries, так как он представляют наибольшую популярность у продавцов и покупателей. В качестве дальнейшего развития проекта можно рассмотреть другие площадки.

Для достижения поставленной цели необходимо решить следующие задачи:

- подготовить набор данных, содержащий изображения товаров и соответствующие им категории и текстовые описания;
- провести исследование текущих архитектур нейронных сетей, используемых для классификации изображений и генерации текста, и на основе этого исследования выбрать наиболее подходящую архитектуру или их комбинацию для решения поставленной задачи;
- обучить выбранную нейронную сеть на подготовленных данных, оптимизировать и настроить параметры модели для повышения её производительности и качества результатов, а затем оценить эффективность реализованной архитектуры нейронной сети;
- интегрировать модель в программное обеспечение.

1 Данные

Данные для поставленной задачи собирались самостоятельно, поскольку в открытых источниках не имелось удовлетворяющего всем требованиям датасета. ([тут можно описать, какие варианты есть](#)). Как было сказано ранее, выбор маркетплейса пал на «Wildberries». Соответственно данные собирались согласно особенностям структуры данного маркетплейса. Для удобства введем некоторые термины, которыми будем оперировать далее:

- **Карточка товара** – это страница продукта на маркетплейсе, где размещена информация о товаре, фотографии, описание цены и кнопка «Купить»
- **Конечная категория** – это категория, на которых располагаются карточки товаров
- **Материнская категория** – это категория, которая содержит конечные и материнские категории и на которой не располагаются карточки товаров

Каталог «Wildberries» разделен на категории, в которых размещены карточки товаров одного типа. Категории выстроены по принципу дерева. Есть основные «широкие» категории, такие как «Женщинам», «Дом», «Продукты», которые объединяют внутри себя более мелкие подкатегории. Например, в разделе «Дом» имеются подкатегории «Ванная», «Кухня», «Спальня» и тд, которые в свою очередь могут подразделяться на еще более маленькие подкатегории.

Требуемые данные располагались на товарных карточках, в которые можно попасть только зная конечную категорию товара. Поэтому было принято решение разделить сбор данных на 2 этапа. На первом этапе был произведен сбор всех имеющихся на «Wildberries» конечных категорий. На втором – сбор необходимой информации с карточек товаров.

1.1 Этап 1

«Wildberries» предлагает 22 основные категории (см. рисунок 1.1), из которых одна является конечной категорией. Данные категории в дальнейшем будем называть категориями первой вложенности. Их подкатегории, соответственно, будут называться категориями второй вложенности. И так далее, спускаясь все ниже по дереву категорий. Экспериментальным путем была выявлена максимальная глубина вложенности – 5.

Для правильного формирования таргета для классификации при сохранении ссылки на конечную категорию нужно было учитывать весь путь по дереву категорий, начиная с первой вложенности. Решением данной задачи стало создание таблицы, где отражалось какие

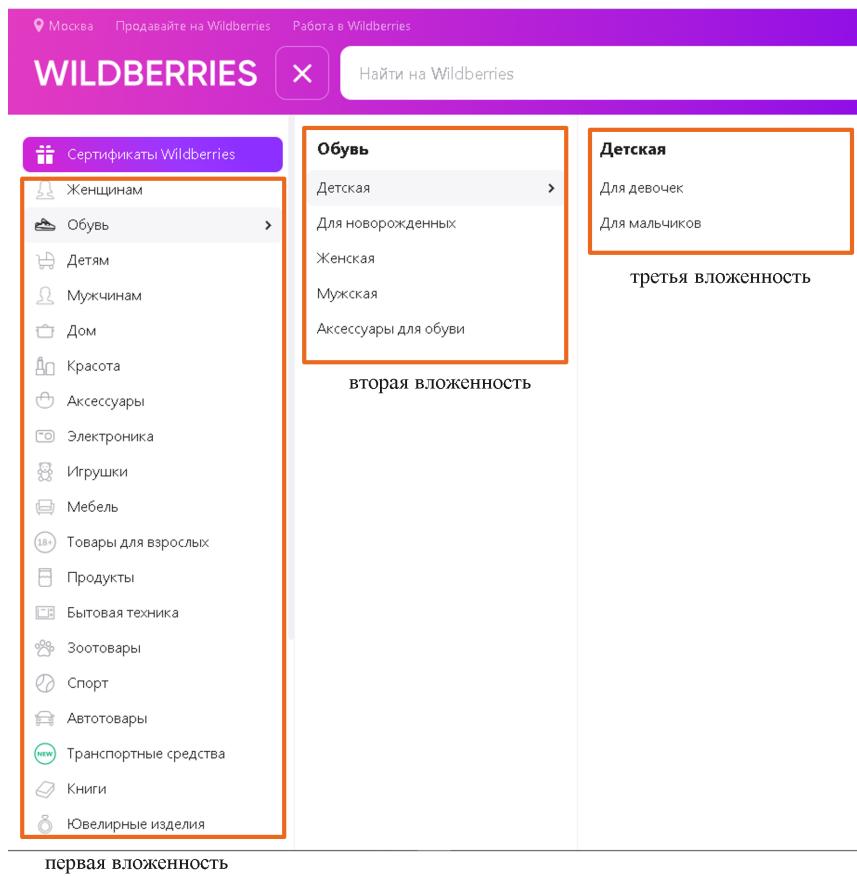


Рис. 1.1: Структура каталога «Wildberries» на примере категории «Обувь».

категории было предшествовавшими конкретной конечной категории (см. таблицу 1.1). При отсутствии более глубокой вложенности на месте данных категорий ставились «NaN». Таким образом, было собрано 1668 конечных категорий.

Таблица 1.1: Фрагмент таблицы, полученной после первого этапа сбора данных.

	category1	category2	category3	category4	category5
437	Дом	Предметы интерьера	Фоторамки и фотоальбомы	Фотоальбомы	NaN
438	Дом	Предметы интерьера	Картинки и постеры	Рамы для постеров	NaN
439	Дом	Предметы интерьера	Картинки и постеры	Постеры	Детская тематика
440	Дом	Предметы интерьера	Картинки и постеры	Картинки	Арт и абстракция
441	Дом	Предметы интерьера	Картинки и постеры	Постеры	Фэнтези

Составление данной таблицы производилось посредством парсинга данных с сайта «Wildberries» через Python с использованием библиотек selenium и BeautifulSoup. Блокировок со стороны маркетплейса замечено не было. Особенность и неудобством парсинга была динамическая подгрузка страниц, которая вынуждала выдерживать паузы в несколько секунд для удовлетворяющей прогрузке страницы. Данное обстоятельство привело с значительному увеличению времени парсинга данных.

При анализе собранной таблицы были выявлены некоторые особенности категори-

альной политики «Wildberries». Во-первых, категории у данного маркетплейса не фиксированы. Например, было отмечено, что часть категорий активно перемещается из раздела в раздел, какие-то категории могут пропадать, также могут появляться новые категории. Данные, собранные в текущем датасете, актуальны на конец января 2024 года. Однако для поддержания списка категорий в актуальном состоянии необходимы механизмы регулярного обновления данных. Во-вторых, на маркетплейсе имеются конечные категории, ссылающиеся на одни и те же url страницы. Подобные категории будут называться дублирующими. Подобные дубли могли иметь разное происхождение: особенности маркетинга и неудачное время парсинга, выпавшее на перемещение категорий. С точки зрения маркетинга подобные дублирования оправданы, поскольку потенциальный покупатели могут по своим соображениям относить одни и те же товары к разным категориям. Для примера, категория «Коврики» находилась в разделе «Автотовары_Коврики» и «Электроника_Автоэлектроника&и&навигация_Коврики». Для корректной работы модели была написана отдельная процедура удаления подобных дублирующих категорий. Выбор, какой из дубликатов оставлять, производился вручную. Всего было найдено 69 дублирующих ссылок, которые могли встречаться 2 и более раза. Таким образом, после удаления в таблице осталось 1580 категорий.

Далее можно было переходить ко 2му этапу.

1.2 Этап 2

Второй этап сбора данных заключался в прохождении по собранному ранее списку конечных категорий и сбора из каждой из них информации с карточек товаров. Было принято решение брать по 20 товаров из каждой конечной категории. Из каждой карточки товара сохранялось первое фотография от продавца, первая фотография из отзыва и описание товара (см. рисунок 1.2). Первая фотография от продавца бралась по причине ее обязательного присутствия в карточке товара, а также гарантированного качественного изображения товара на ней. Однако поскольку разрабатываемый сервис рассчитан на работу в большинстве случаев с фотографиями от пользователей, все дефекты, присущие любительским фотографиям могут иметь место быть. Поэтому для стабильности предсказаний классификационной модели, было решено подавать в нее также фотографии из отзывов, которые максимально близко будут похожи на фотографии, с которыми будет работать в дальнейшем сервис. Описания товара были нужны для задачи генерации текстовых описаний к изображениям. В итоге при полном наборе для каждой конечной категории имелось 40 фотографий (20 фотографий от

продавцов и 20 фотографий из отзывов) и 20 описаний.

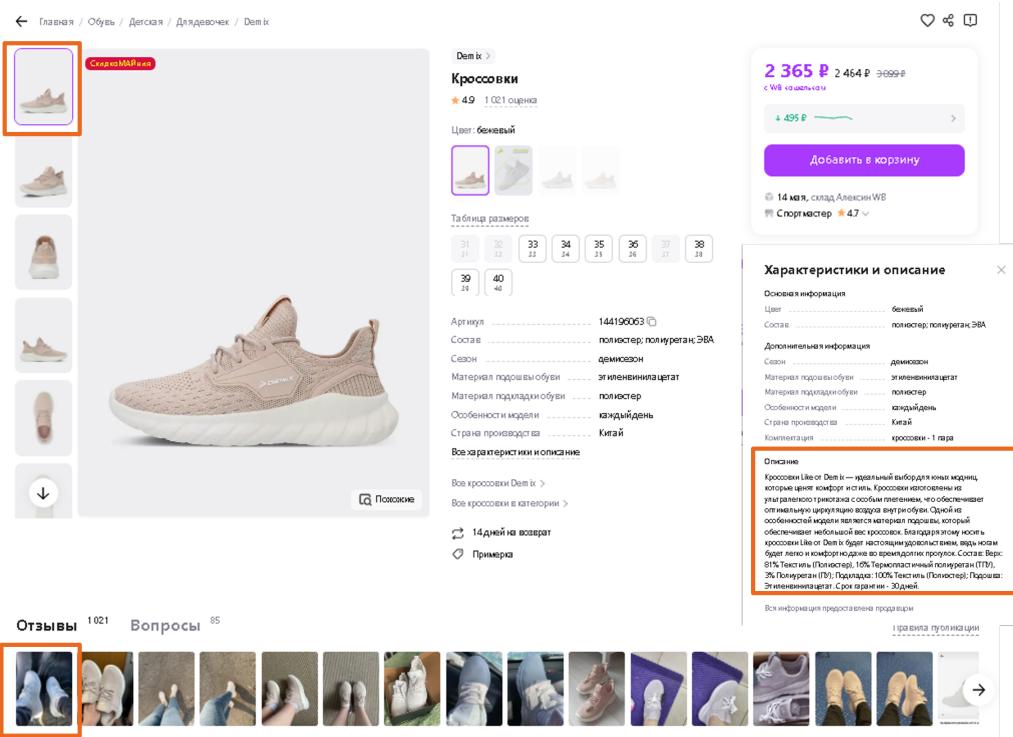


Рис. 1.2: Пример карточки товара на «Wildberries». Верхний левый прямоугольник – первая фотография от продавца. Нижний левый прямоугольник – первая фотография из отзыва. Правый прямоугольник – текстовое описание товара.

При выполнении данного этапа было несколько трудностей. Во-первых, стандартная на «Wildberries» динамическая загрузка страниц, увеличивающая время парсинга более чем в 3 раза. Приходилось делать паузы при открытии страницы с карточкой товара, при открытии описания, лежащее в отдельной вкладке, и пролистывание страницы вниз для подгрузки информации об отзывах. Примерное время парсинга данных, затраченное на второй этап, равнялось 2м неделям. Во-вторых, имели товары с меткой «18+», для которых требовалось дополнительное нажатие кнопки, подтверждающее достижение указанного возраста. В-третьих, некоторые поля в карточке товара заключали в себе картинки, которые вынуждали в определенных случаях дополнительно пролистывать страницу вниз. В-четвертых, для нажатия кнопки с целью получения описание к товару, выдвигалось требование расположение кнопки в зоне видимости экрана. Это приводило к еще более тонкой настройке пролистывания страницы, подобранный под конкретный размер экрана компьютера.

Для сохранения данных из карточек товара была придумана специальная структура с целью дальнейшего удобства использования в задаче классификации и генерации текста. Все товары, собранные из одной конечной категории, сохранялись в отдельную папку, содержащую следующие элементы:

- папку «card», куда складывались фотографии от продавцов
- папку «feedbacks», куда складывались фотографии из отзывов
- файл «descriptions.csv», где сохранялись описания к товарам

Название данной папки определялось посредство таблицы 1 и складывалось из всех материнских категорий, участвовавших в пути к конечной категории. Например, для конечной категории «Фотоальбомы» (см. таблицу 1.1) название папки было следующее: «Дом_Предметы а для категории «Фэнтези» - «Дом_Предметы&интерьера_Картины&и&постеры_Постеры_Фэнтези». Более подробно об использовании подобной структуры ранения данных будет описание в главе ____ в разделе ____.

Первичный анализ собранных данных выявил, что не у всех товаров имелись отзывы с фотографиями и описания. Описания имелись в 99.8% проценте случаев. В таблице 1.2 приведены некоторые статистические данные о собранных фотографиях от продавца и из отзыва. Можно заметить, что некоторые конечные категории были полностью без фотографий в отзывах. Однако, опираясь на перцентили, можно сделать вывод, что таких категорий было довольно мало. Касательно фотографий от продавцов можно сделать 2 вывода. Во-первых, есть категории, представленные менее чем 20ю товарами. Во-вторых, есть как минимум одна категория, в которой имеется только 1 товар. Подобные категории нас не устраивают, потому что далее будет производиться деление каждой категории на 2 части, и категории с одним товаром невозможно будет разделить.

Таблица 1.2: Описательная статистика по фотографиям от продавца (столбец «card») и фотографиям из отзыва (столбец «feedbacks»).

	card	feedbacks
count	1580	1580
mean	19.98	17.72
std	0.63	4.23
min	1	0
25%	20	18
50%	20	19
75%	20	20
max	20	20

Всего категорий, представленных менее 20 товарами, было выявлено 5 штук (см. таблицу 1.3). Из них представляли наибольший интерес __ и __, из-за чрезесчур малого количества товаров. Категорию с одним товаром было решено удалить. Таким образом, осталось 1579 конечных категорий, с которыми шла вся дальнейшая работа.

Таблица 1.3: Таблица с категориями, имеющими менее 20 товаров.

кол-во товаров	категория
18	Дом_Кухня_Кухонный&текстиль_Чехлы&для&ручек&холодильников
4	Дом_Освещение_Лифты&для&люстр
19	Мебель_Гардеробная&мебель_Ящики
1	Мебель_Офисная&мебель_Перегородки&офисные
19	Мебель_Офисная&мебель_Шкафы

При более детально рассмотрении собранных данных было замечено, что фотографии из отзывов довольно шумные (см. рисунок 1.3). Очень много одинаковых фотографий, фотографий, где не очень понятно, что изображено. Поэтому для дальнейшей работы использовались только фотографии товара от продавца.



Рис. 1.3: Фотографии из отзывов в категории «Игрушки_Антистресс».

Далее интересно было рассмотреть количество собранных данных в разрезе категорий первой вложенности (см. рисунок 1.4). Из круговой диаграммы можно заметить, что категории довольно несбалансированный. Например, категория «Дом» вмещает в себя порядка 6000 пример. В то время как в категории «Ювелирные&изделия» только 320 примеров.

Подобную картину можно наблюдать в категориях всех вложений (см. приложение А).

После появление базового понимания данных надо было приступить к их детальному изучению. Как правило, парсинг большого количества данных, тем более с постоянно меняющихся маркетплейсов, не проходит идеально. В сохранных данных могут быть оплошности, которые помешают грамотно решать задачу классификации и генерации текста. Приведем некоторые примеры, выявленных особенностей, требуемых принятие решений с нашей сто-

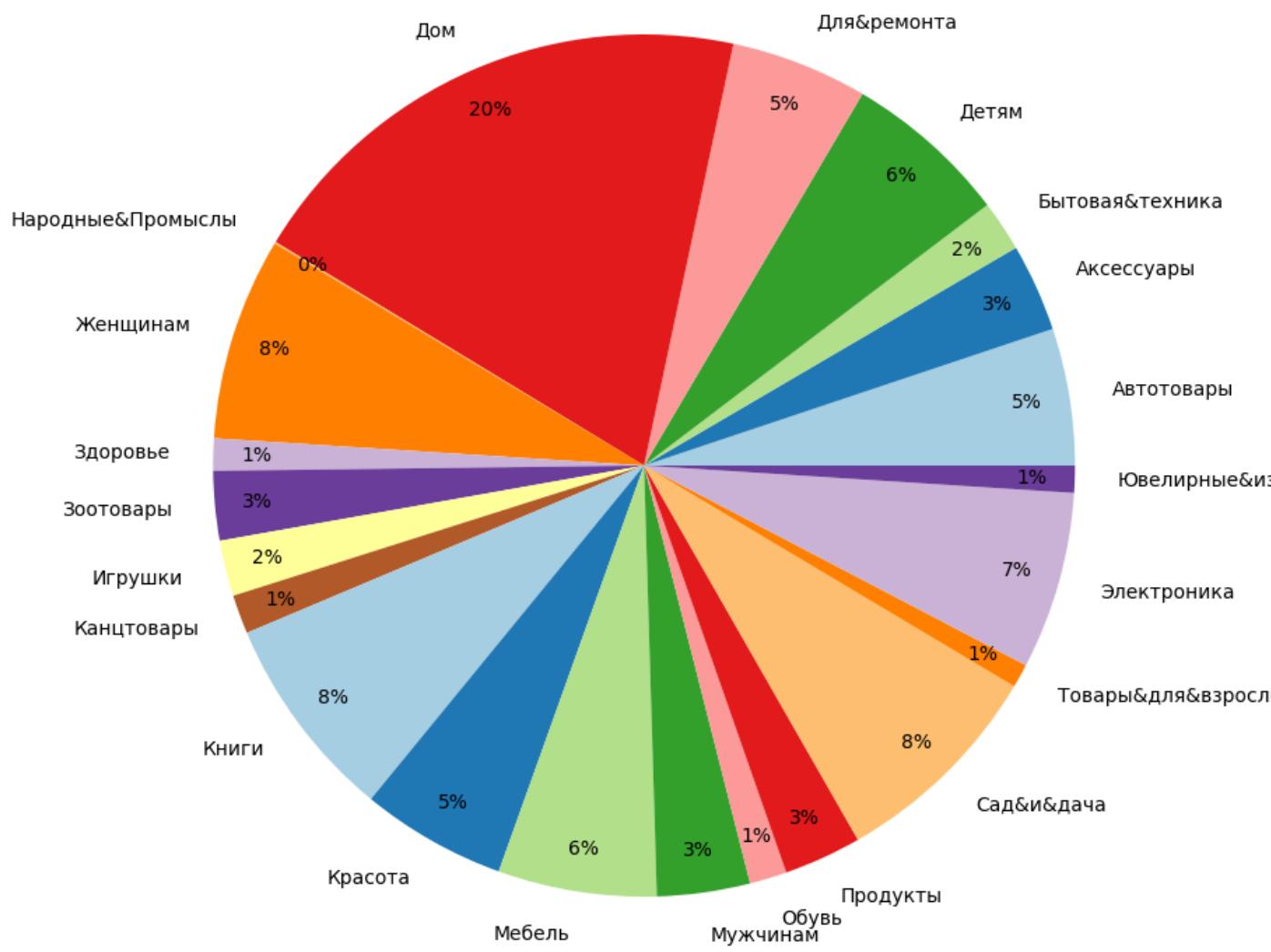


Рис. 1.4: Распределение данных по категориям первой вложенности.

роны:

- Встречаются категории, которые с точки зрения категорийного менеджмента, имеют место быть. Одна для классификационной модели машинного обучения такие категории будут не осиливаемыми. Например, в разделе «Женщинам» есть подкатегории «Женщинам_Белье» и «Женщинам_Большие&размеры_Белье». Если детально изучить фотографии, которые в этих категориях присутствуют, можно сделать вывод, что особых различий между ними нет. Единственное, что было подмечено, что на очень немногих фотографиях стоит надпись «4XL» или что-то подобное, указывающее, что у данного товара имеются большие размеры. Более того, в обеих категориях присутствуют одинаковые товары.

- В данных попадались «мусорные» категории. Предположительно, подобное возникало из-за изменения url ссылок на категории со стороны «Wildberries». В подобных категориях находились товары, собранные случайным образом из всевозможных категорий с маркетплейса.
- Распределение товаров по категориям не очень четкая задача. В связи с этим встречались одинаковые товары, находящиеся в разных категориях. Например, одна и та же продукт мог находиться в категориях «Зоотовары_Груминг&и&уход», «Зоотовары_Для&кошек_Груминг&и&уход» и «Зоотовары_Для&собак_Груминг&и&уход».
- В части материнских категорий встречались разделы «Подарки» (например, материнские категории «Мужчинам_Подарки&мужчинам» и «Женщинам_Подарки&женщинам»), куда были собраны товары из совершенно разных категорий, таких как «Аксессуары», «Дом», «Продукты» и тд.
- Поскольку при парсинге из каждой категории брались первые 20 товаров, появляется неконтролируемый фактор того, какие товары стоят вначале. Как правило, пользователи смотрят только на первые товары в выдаче. Поэтому на маркетплейсах существуют множество механизмов и правил отбора товаров, которые будут показаны пользователю в начале. В нашем случае было замечено, что некоторые категории стали более шумными из-за сезонных товаров. Например, в категории «Зоотовары_Фермерство» были найдены пасхальные яйца.
- Бывали категории, которые по смыслу имели место быть как отдельные категории, однако в них были собраны не совсем подходящие товары. Например, в категории «Мужчинам_Религиозная_Православие» находились обычные рубашки и штаны, часть из которых присутствовала также в категории «Мужчинам_Рубашки» и «Мужчинам_Брюки».

Приведенные особенности сохраненных данных требовали ручной очистки датасета. Необходимо было применять следующие действия: полное удаление категории, удаление конкретной фотографии из отзыва, удаление товара полностью (фотографию от продавца, из отзыва и описание к нему) и произведение полного переноса товара (фотографию от продавца, из отзыва и описание к нему) из одной категории в другую. Для удобства и быстроты данной процедуры были написаны функции, позволяющие механизмами Python вносить изменения в собранные данные.

По окончании данной процедуры было подмечено, что категории требовали очистки в разной степени. Какие-то категории, как «Зоотовары», требовали практически полного

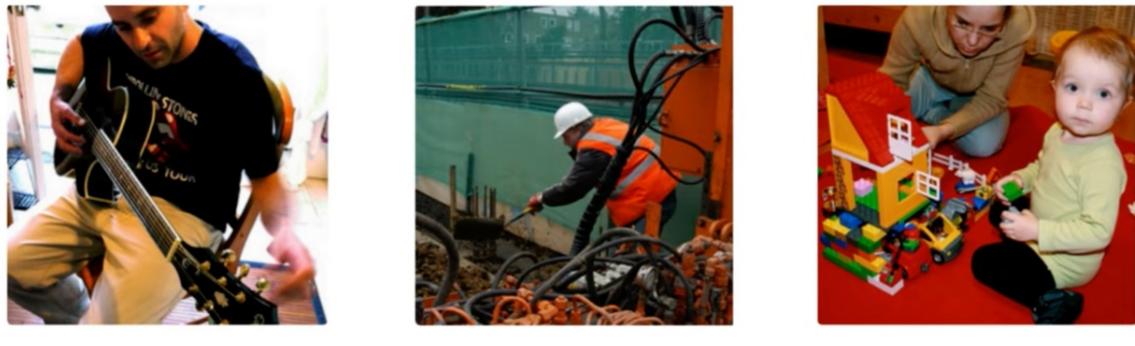
переформирования. Другие категории обходились легкой очисткой, например «Продукты». Некоторые категории совсем не требовала вмешательства. Таким образом, финальный датасет стал состоять из 1459 конечных категорий (см. таблицу 1.3).

Таблица 1.4: Описательная статистика по фотографиям от продавца после очистки собранных данных. Примечание: статистика по фотографиям из отзывов не приведена, поскольку, как упоминалось ранее, решено было в дальнейшем работать только с фотографиями от продавца.

	card
count	1459
mean	20.09
std	2.32
min	4
25%	20
50%	20
75%	20
max	54

2 Декомпозиция задачи

Для достижения целей и задач проекта требуется решить задачу создания подписей к изображениям (англ. image captioning, см. примеры на рисунке 2.1). Кроме того, необходимо решить задачу классификации для подбора наиболее подходящей категории по фотографии товара. Для успешной реализации проекта необходимо выбрать подходящие модели и определить метрики, которые будут использоваться для оценки их эффективности.



"Man in black shirt is playing guitar."

"Construction worker in orange safety vest is working on road."

"Two young girls are playing with lego toy."

Рис. 2.1: Примеры задачи Image Captioning.

2.1 Задача Image Captioning

Image Captioning — задача создания текстового описания входного изображения, основанная на комбинации сверточных нейронных сетей для обработки изображения и рекуррентных нейронных сетей для генерации подписи, то есть реализующая архитектуру кодировщик-декодировщик (англ. encoder-decoder, см. рисунок 2.2).

Кодировщик отвечает за преобразование изображения в компактное представление (вектор признаков), которое сохраняет важную информацию об изображении. Обычно в качестве энкодера используется CNN, такие как ResNet, Inception, MobileNet или EfficientNet. Эти модели предварительно обучены на больших наборах данных, таких как ImageNet, для извлечения визуальных признаков. В качестве вектора признаков изображения может использоваться выходной слой CNN, но чаще всего это слой перед последним полносвязанным. Также в качестве Image Encoder для извлечения признаков изображения могут быть использованы Vision Transformers (ViT) [vit]. Трансформеры представляют собой архитектуру, основанную на механизме внимания, которая была успешно применена к различным



Рис. 2.2: Архитектура задачи Image Captioning.

задачам компьютерного зрения, включая генерацию подписей к изображениям. ViT может достичь лучших результатов, чем большинство современных сетей CNN. Однако была разработана модель EfficientNet V2, которая быстрее обучается и обладает меньшей потребностью в ресурсах по сравнению с ViT, но при этом лучше работает со многими задачами компьютерного зрения [[efficientnetv2](#)]. Несмотря на высокую производительность, EfficientNet V2 не использует механизм внимания, как ViT, что может быть недостатком в некоторых случаях. Поэтому выбор кодировщика зависит от конкретных требований проекта и условий, в которых будет использоваться модель. В некоторых случаях может быть полезным попробовать обе архитектуры и выбрать ту, которая лучше соответствует задачам и ресурсам.

Декодировщик принимает вектор признаков изображения и генерирует последовательность слов, которая описывает изображение. Декодеры, как правило, основаны на рекуррентных нейронных сетях (RNN), однако современные подходы могут использовать трансформеры. В ранних моделях чаще всего используется LSTM (Long Short-Term Memory) или GRU (Gated Recurrent Unit) для генерации последовательности слов. Одна из основных проблем такого подхода - это то, что RNN имеет фиксированную длину контекста и обрабатывает информацию последовательно. Это означает, что RNN видит только небольшой контекст, поэтому при генерации слов часто забывает старые результаты своей генерации. Так же такой подход не позволяет полностью учесть контекст изображения на каждом этапе генерации. В данном случае выделенные признаки изображения с помощью CNN лишь единожды подаются на вход LSTM блока, так что со временем генерация полностью теряет память о исходном изображении и начинает "додумывать самостоятельно". Для решения этих проблем приме-

няется механизм внимания. В случае image captioning, механизм внимания позволяет сети "обращаться" к различным частям изображения на каждом шаге генерации текста. Таким образом контекст самого изображения не теряется со временем генераций, и механизм внимания позволяет сети сфокусироваться на разных частях изображения с разной степенью "важности" на каждом шаге генерации, что значительно увеличивает качество финального описания. На рисунке 2.3 представлена архитектура декодировщика.



Рис. 2.3: Архитектура декодировщика в задаче Image Captioning.

Более современные подходы используют трансформеры, такие как модель Transformer от Vaswani et al.[**transformers**]. Трансформеры лучше справляются с захватом долгосрочных зависимостей в последовательности слов.

При обучении модели обычно используется кросс-энтропийная функция потерь для сравнения сгенерированных описаний с эталонными в обучающем наборе данных. Для оценки качества генерации используются метрики, такие как BLEU, METEOR и CIDEr.

2.2 Обзор классификаторов изображения

В этом разделе будет дан обзор существующих классификаторов изображений, а также будет выбрана архитектура модели, которая будет использоваться в проекте, в том числе для использования ее в качестве кодировщика в задаче Image Captioning.

2.2.1 ResNet

2.2.2 Inception

2.2.3 MobileNet

MobileNet — это семейство моделей нейронных сетей, специально разработанных для работы на мобильных и встраиваемых устройствах [mobilenet]. Они оптимизированы для эффективности и производительности, обеспечивая высокую точность при низких вычислительных затратах. MobileNet широко используется в задачах компьютерного зрения, таких как классификация изображений, детекция объектов и сегментация.

Основной инновацией MobileNet является использование глубоких разделяемых сверточных слоев вместо стандартных сверточных слоев. Глубокие разделяемые сверточные слои состоят из двух отдельных операций: глубинного свертывания (англ., depthwise convolution) и точечного свертывания (англ., pointwise convolution). Глубинное свертывание применяется отдельно к каждому каналу входного изображения, а точечное свертывание используется для объединения выходов глубинного свертывания. MobileNet использует параметры разложения для управления шириной сети (англ., width multiplier) и разрешением входного изображения (англ., resolution multiplier). Width multiplier уменьшает количество каналов в каждой сверточной операции, что снижает количество вычислений и параметров, а resolution multiplier уменьшает размер входного изображения, что дополнительно снижает вычислительные затраты. Благодаря глубинным разделяемым сверточным слоям и параметрам разложения, MobileNet достигает хорошего баланса между производительностью и вычислительными затратами. На рисунке 2.4 представлена архитектура MobileNet слоя.

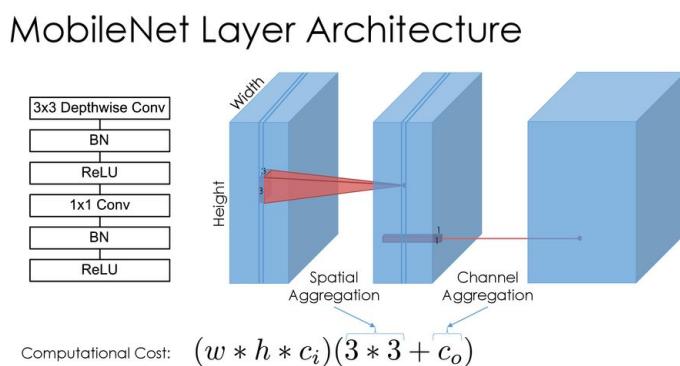


Рис. 2.4: Архитектура MobileNet слоя.

Существует несколько версий MobileNet, каждая из которых приносит свои улучшения и оптимизации.

MobileNetV2 — включает улучшения, такие как блочные инвертированные резидуальные соединения (англ., inverted residuals) и линейные узкие слои (англ., linear bottlenecks) [mobilenetv2], см. рисунок 2.5. Эти улучшения помогают сохранять более полезные характеристики признаков и обеспечивают лучшую производительность при тех же вычислительных затратах.

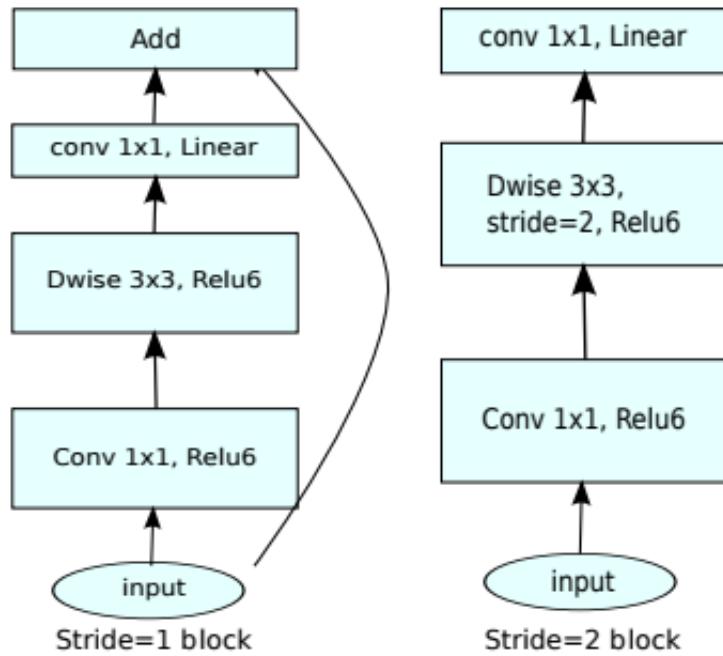


Рис. 2.5: Inverted residuals и Linear bottlenecks.

mobilenet-v3-block

MobileNetV3 — включает дополнительные улучшения, такие как блоки сжатия и возбуждения (англ., squeeze-and-excitation, SE) и расширенные функции активации [mobilenetv3], см. рисунок 2.6. Блоки SE позволяют сети лучше фиксировать канальные зависимости в данных, помогает повысить способность модели извлекать значимые функции из входных данных, что приводит к повышению производительности задач распознавания изображений. MobileNetV3 использует расширенные функции активации, такие как h-swish и h-sigmoid. Эти функции обеспечивают более плавные градиенты во время обучения, что может привести к более быстрой сходимости и повышению общей производительности. MobileNetV3 предоставляет две версии: MobileNetV3-Large и MobileNetV3-Small, предназначенные для различных требований производительности и эффективности.

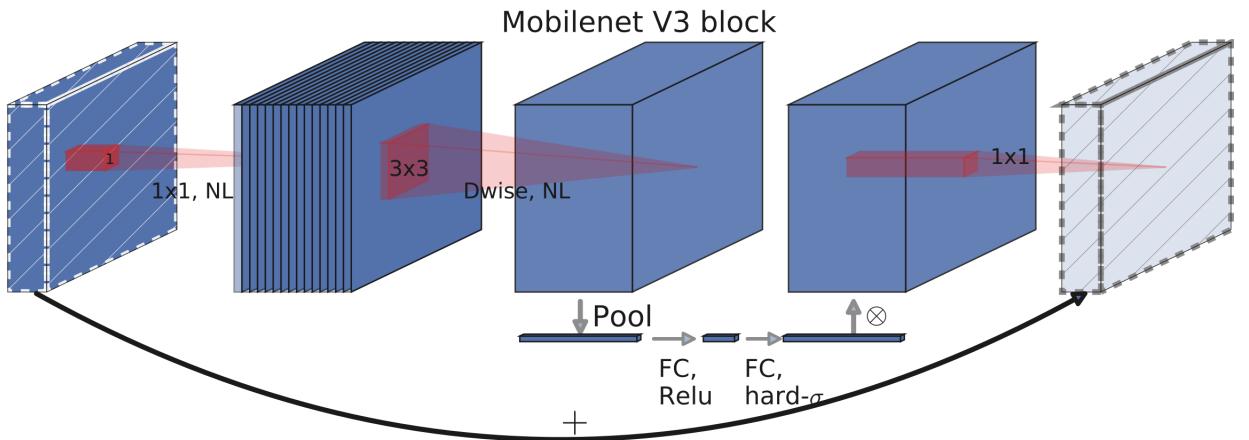


Рис. 2.6: MobileNet V3 блок.

2.2.4 MobileNet

2.3 Используемые метрики в задаче классификации

В задаче классификации для оценки качества моделей машинного обучения используются специальные показатели — метрики.

Для описания метрик используется матрица ошибок классификаций (англ. confusion matrix). Пусть есть два класса и метод, предсказывающий принадлежность каждого объекта одному из классов. В таблице 2.1 представлена матрица ошибок классификации.

	$y = 1$	$y = 0$
$\hat{y} = 1$	True Positive (TP)	False Positive (FP)
$\hat{y} = 0$	True Negative (TN)	False Negative (FN)

Таблица 2.1: Матрица ошибок классификации. y — истинный класс, \hat{y} — результат модели.

Простейшей метрикой является accuracy — доля правильных ответов:

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

Стоит отметить, что эта метрика может быть применима только на сбалансированном датасете.

Для оценки качества работы алгоритма на каждом из классов по отдельности вводятся метрики точность (англ. precision) и полнота (англ. recall):

$$precision = \frac{TP}{TP + FP} \quad (2)$$

$$recall = \frac{TP}{TP + FN} \quad (3)$$

Precision — доля объектов, которые классификатор определил как положительные и при этом они действительно являются положительными, а recall — метрика, которая показывает, какую долю положительных объектов из всех объектов положительного класса обнаружила модель. Введение precision не позволяет определять все объекты в один класс, так как в этом случае получается рост FP. Recall показывает способность метода вообще обнаруживать данный класс, а precision — способность отличать этот класс от других. В отличие от accuracy, recall и precision не зависят, от соотношения классов и поэтому могут быть применимы на несбалансированном датасете.

Также существует несколько способов объединить recall и precision в объединенный критерий оценки. Например, F-мера (в общем случае F_β) — среднее гармоническое recall и precision:

$$F_\beta = (1 + \beta^2) \cdot \frac{precision \cdot recall}{(\beta^2 \cdot precision) + recall} \quad (4)$$

β в данном случае определяет вес точности в метрике.

F-мера достигает максимума при точности и полноте, равными 1, и близка к 0, если один из аргументов близок к 0.

2.4 Проектирование архитектуры программного продукта (в процессе)

В 2022 году граждане Российской Федерации столкнулись с трудностями работы с магазинами приложений для смарт-устройств. Так, например, с 24 февраля из российского раздела App Store были удалены почти 7000 мобильных приложений [[russia_app](#)]. Поэтому при построении архитектуры приложения был выбран подход к созданию клиент-серверного приложения, которое будет работать практически на любом устройстве (стационарный компьютер, ноутбук, планшет или смартфон). Одним из способов реализации такого подхода является написание telegram бота. В настоящее время это одно из трендовых направлений в ИТ сфере. В 2024 году Telegram посещают 900 миллионов человек в месяц. По количеству аудитории он входит в пятерку самых популярных мессенджеров в мире. Выбор в пользу написания телеграм-бота вместо веб-приложения может быть обусловлен рядом факторов, включая удобство использования, скорость разработки, доступность, безопасность, а также

гибкость и масштабированность для обработки большого количества пользователей.

За логику, работоспособность и правильное функционирование бота отвечает серверная часть (англ. backend), которая скрыта от пользователя. Серверная часть включает в себя:

- модуль для обработки запросов от пользователя;
- модуль для взаимодействия с обученными моделями классификации для определения категории товара и генерации его описания.

В контексте работы с моделями машинного обучения и их применения в системах автономного управления, вынос предсказаний моделей в очередь сообщений может быть полезно для оптимизации работы и повышения устойчивости системы.

Для проектирования, разработки и обучения нейронных сетей использовался следующий стек технологий:

- Язык программирования Python — язык общего назначения, с помощью которого можно решать сложные задачи машинного обучения и быстро создавать прототипы для последующей их отладки. Язык гибкий и мультиплатформенный, имеет обширный набор библиотек для искусственного интеллекта. Python удобно использовать для обработки и подготовки обучающих данных.
- Pytorch — библиотека глубокого обучения с открытым исходным кодом, написанная на языке Python и созданная на базе Torch. Используется для решения различных задач машинного обучения: компьютерное зрение, NLP, создания и обучения нейронных сетей [[pytorch](#)].

Вычисления происходили с использованием аппаратных ускорений на GPU Nvidia Tesla V100 и A100.

Для реализации серверной части был выбран Flask — это небольшой и легкий веб-фреймворк, написанный на языке Python, предлагающий полезные инструменты и функции для облегчения процесса создания веб-приложений с использованием Python [[flask](#)].

Docker — инструмент для создания и запуска контейнеров.

Kubernetes нужен, когда у вас много контейнеров и узлов, которыми нужно управлять. Также Kubernetes подходит, если вам нужна распределенная отказоустойчивая система.

На рисунке ?? представлена структура приложения.

A Приложение 1

Распределение данных по категориям второй вложенности.

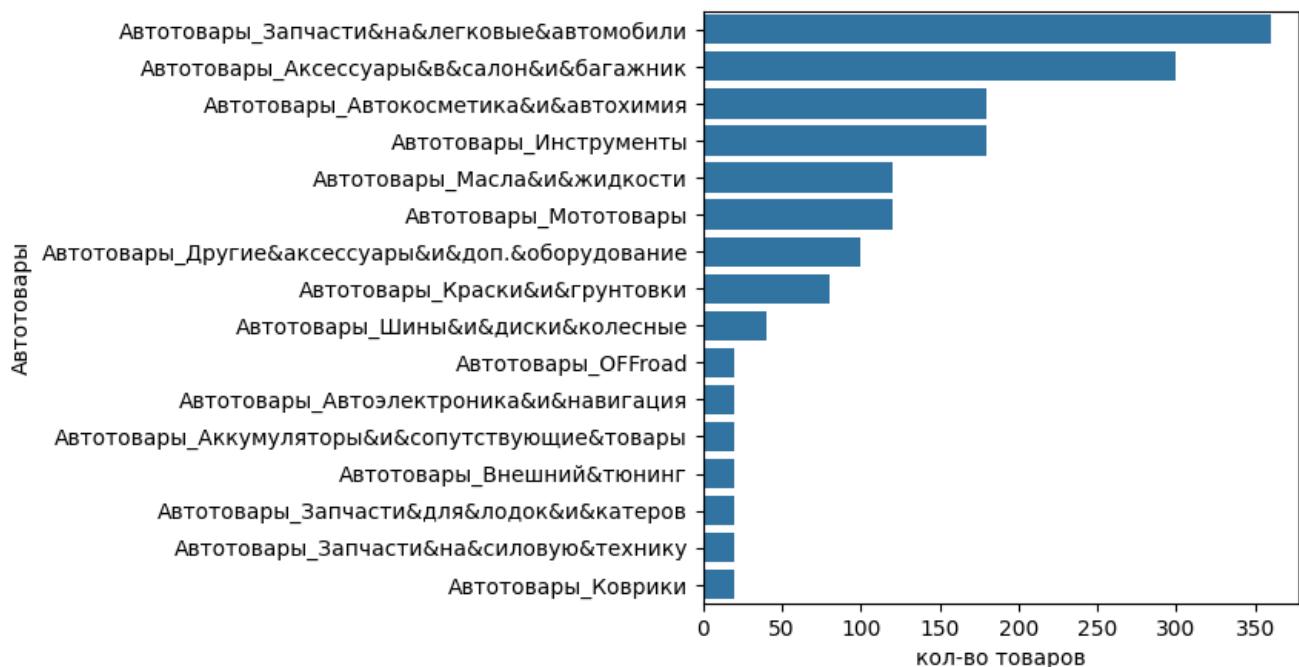


Рис. А.1: Распределение данных в категории «Автотовары».

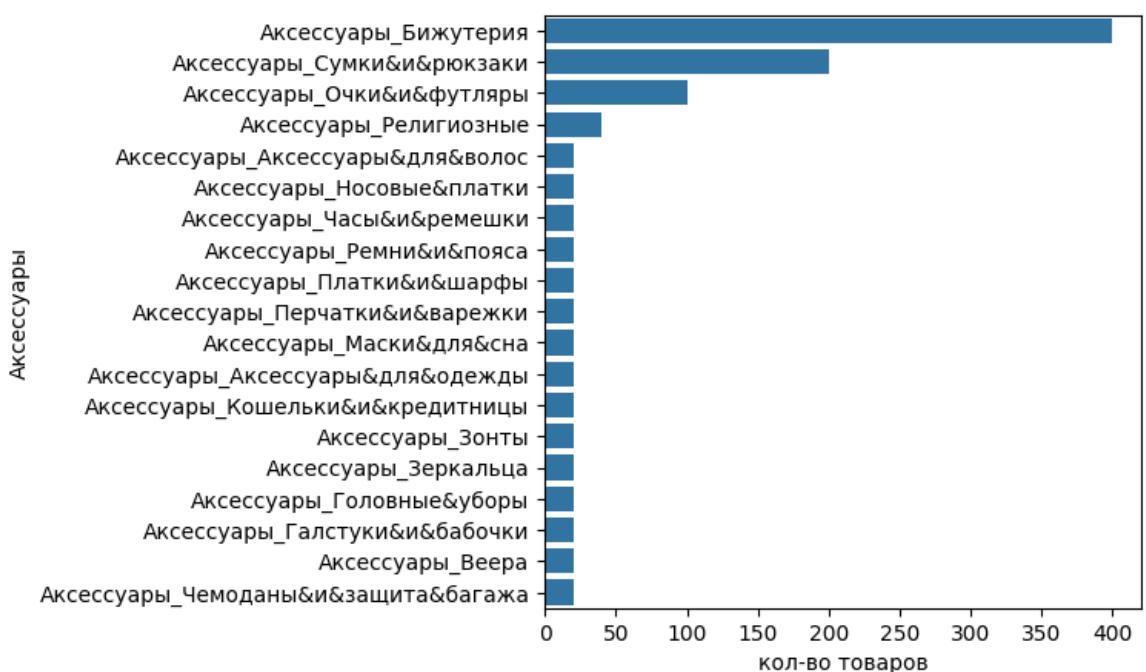


Рис. А.2: Распределение данных в категории «Аксессуары».

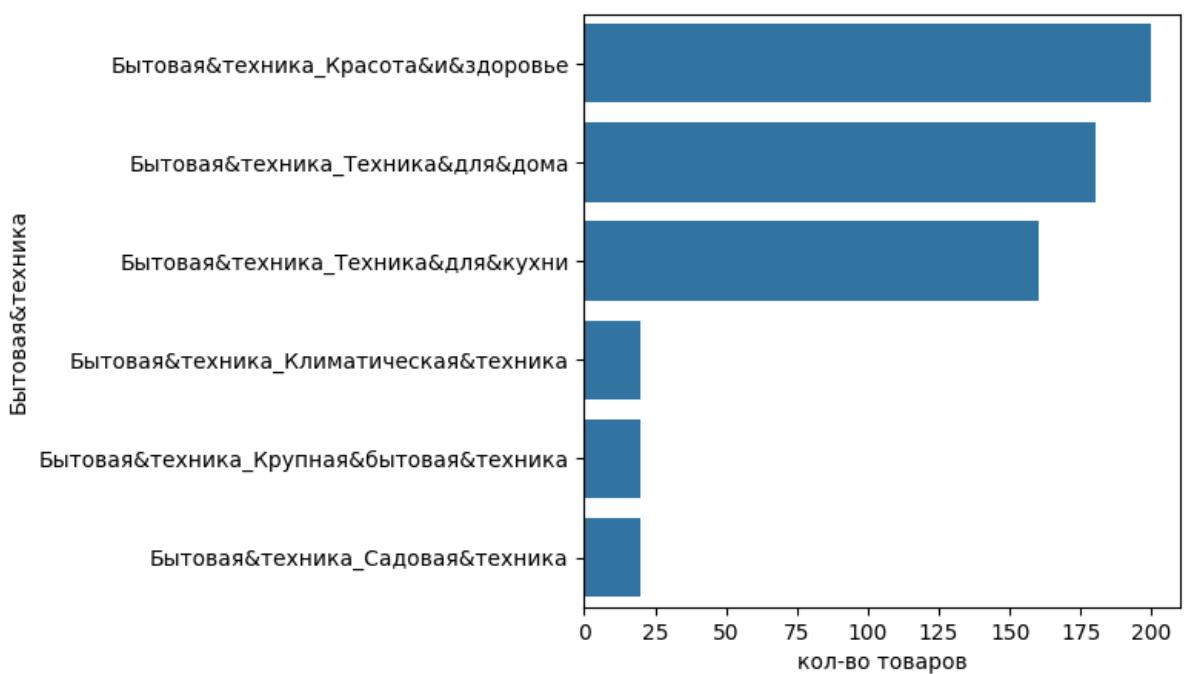


Рис. А.3: Распределение данных в категории «Бытовая&техника».

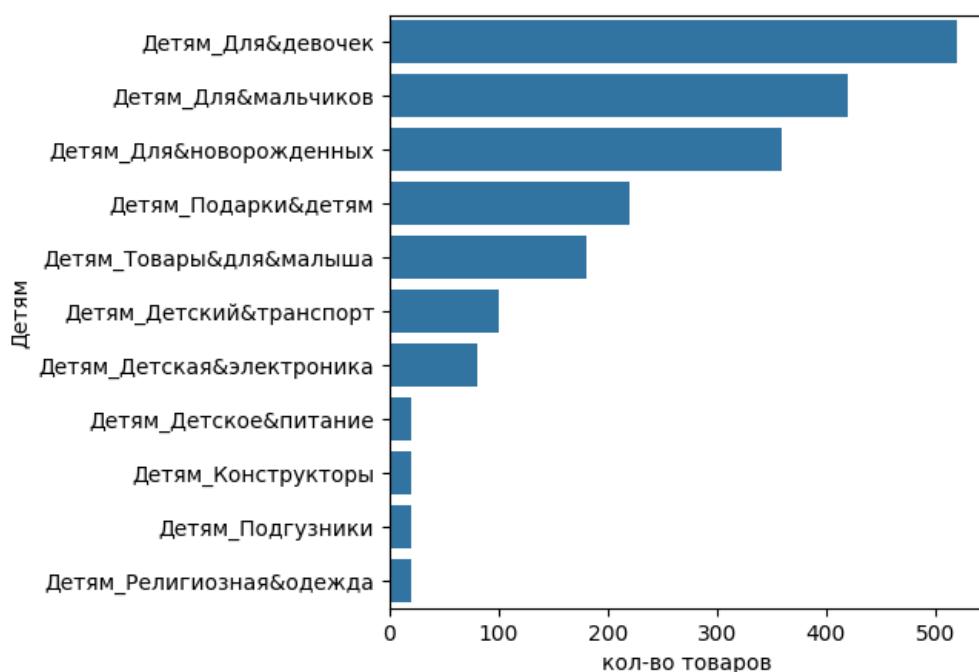


Рис. А.4: Распределение данных в категории «Детям».

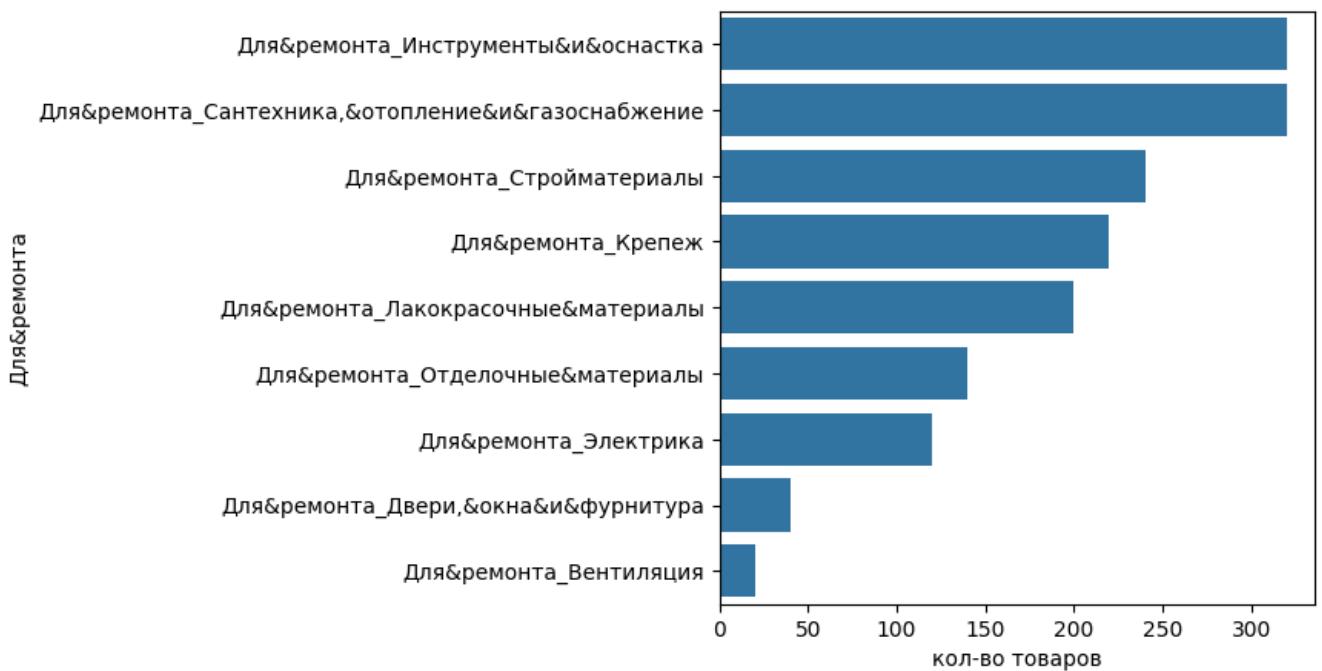


Рис. А.5: Распределение данных в категории «Для&ремонта».

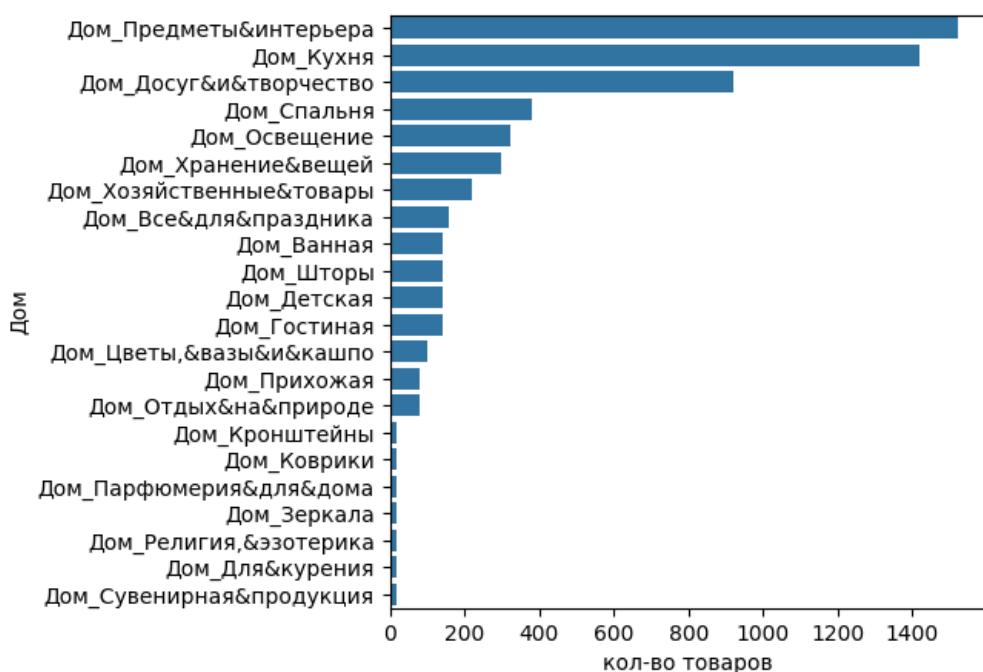


Рис. А.6: Распределение данных в категории «Дом».

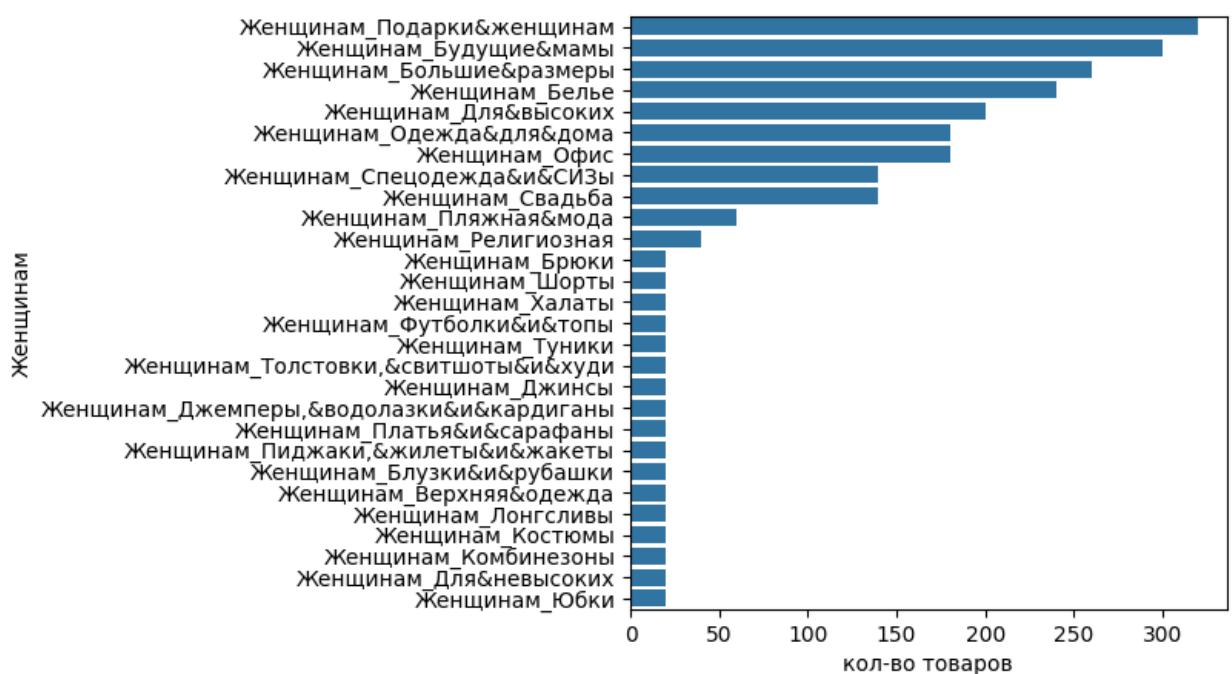


Рис. А.7: Распределение данных в категории «Женщинам».

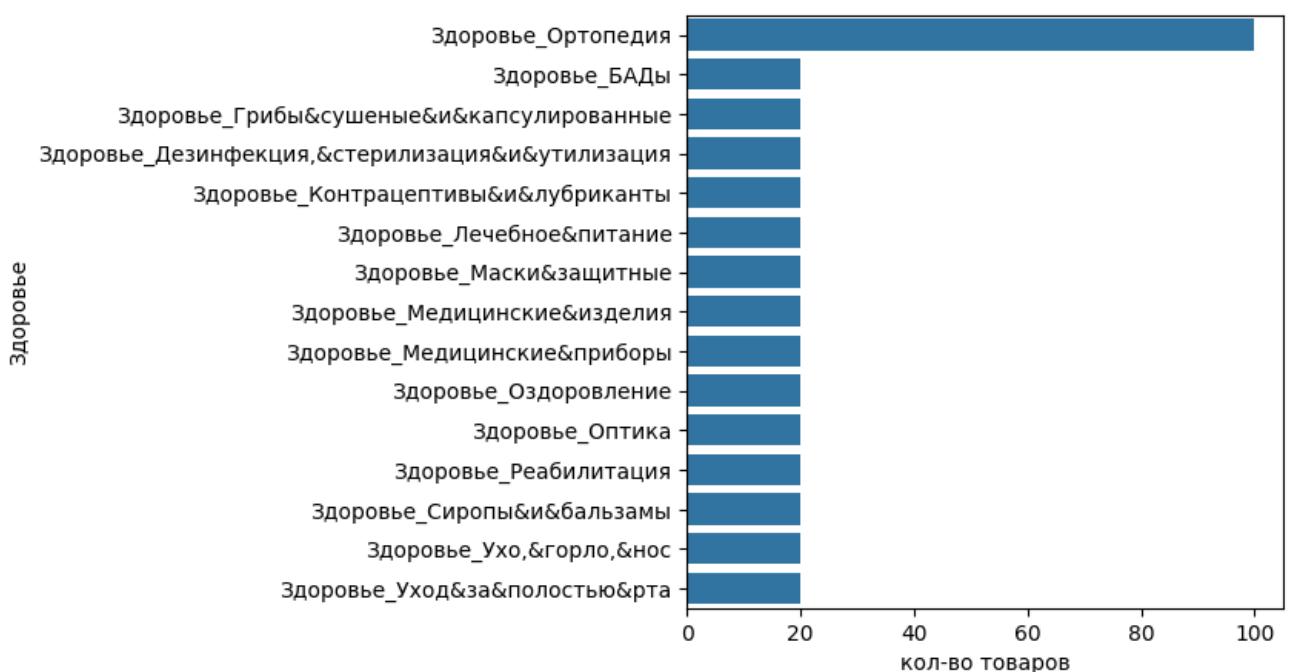


Рис. А.8: Распределение данных в категории «Здоровье».

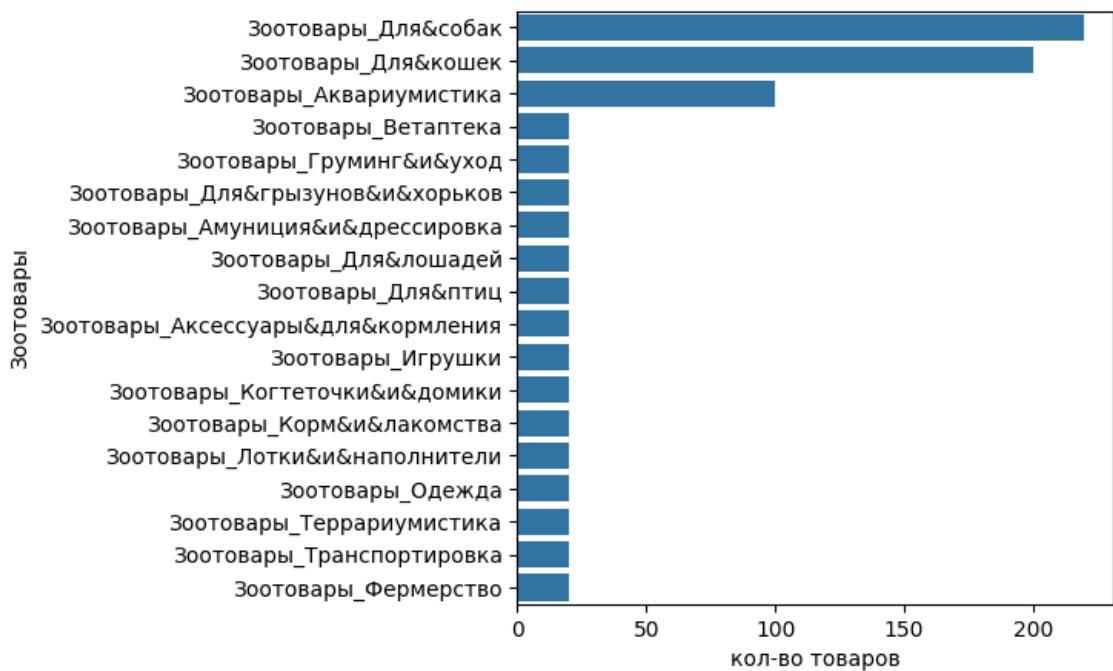


Рис. А.9: Распределение данных в категории «Зоотовары».

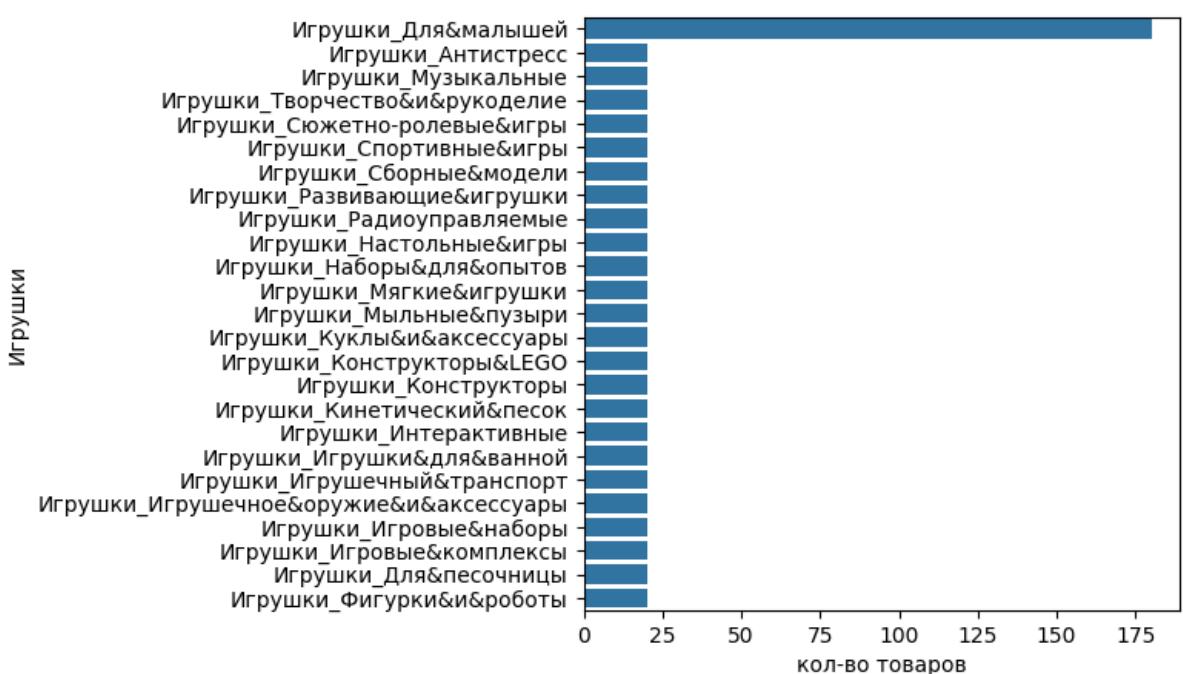


Рис. А.10: Распределение данных в категории «Игрушки».

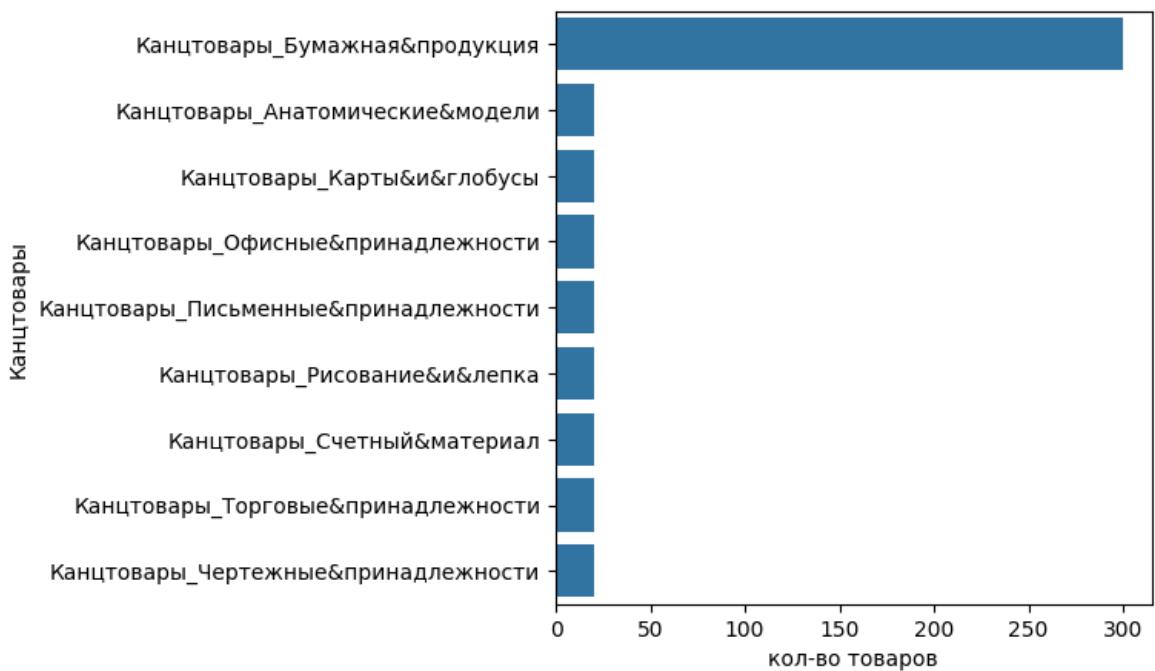


Рис. А.11: Распределение данных в категории «Канцтовары».

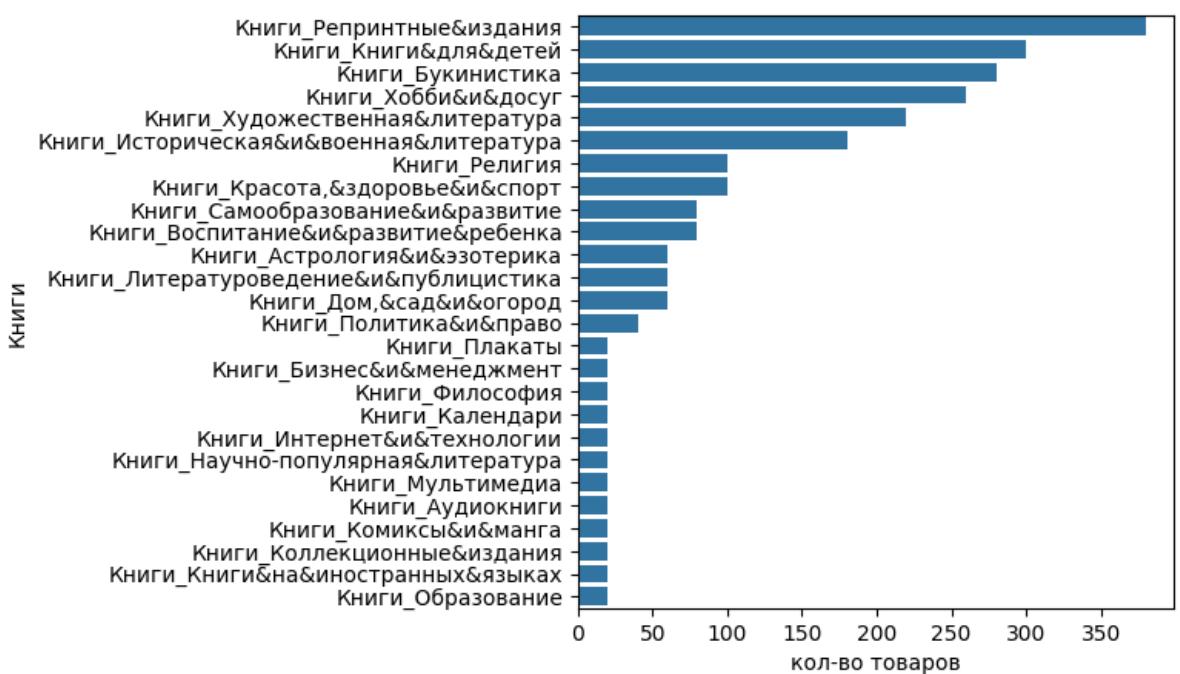


Рис. А.12: Распределение данных в категории «Книги».

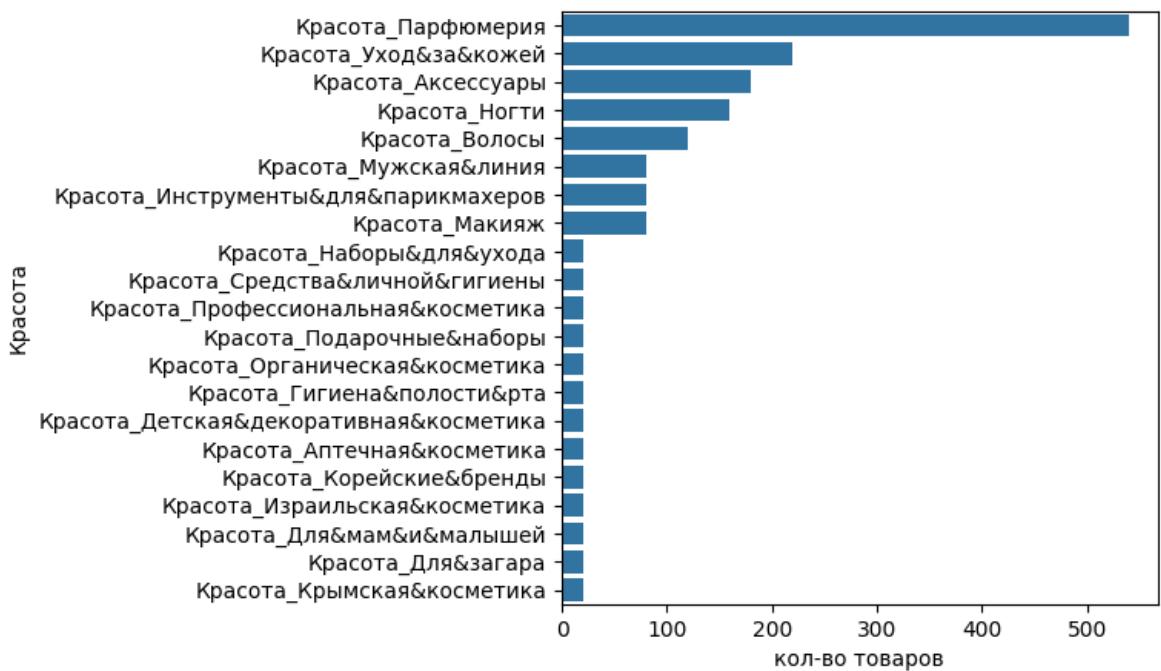


Рис. А.13: Распределение данных в категории «Красота».

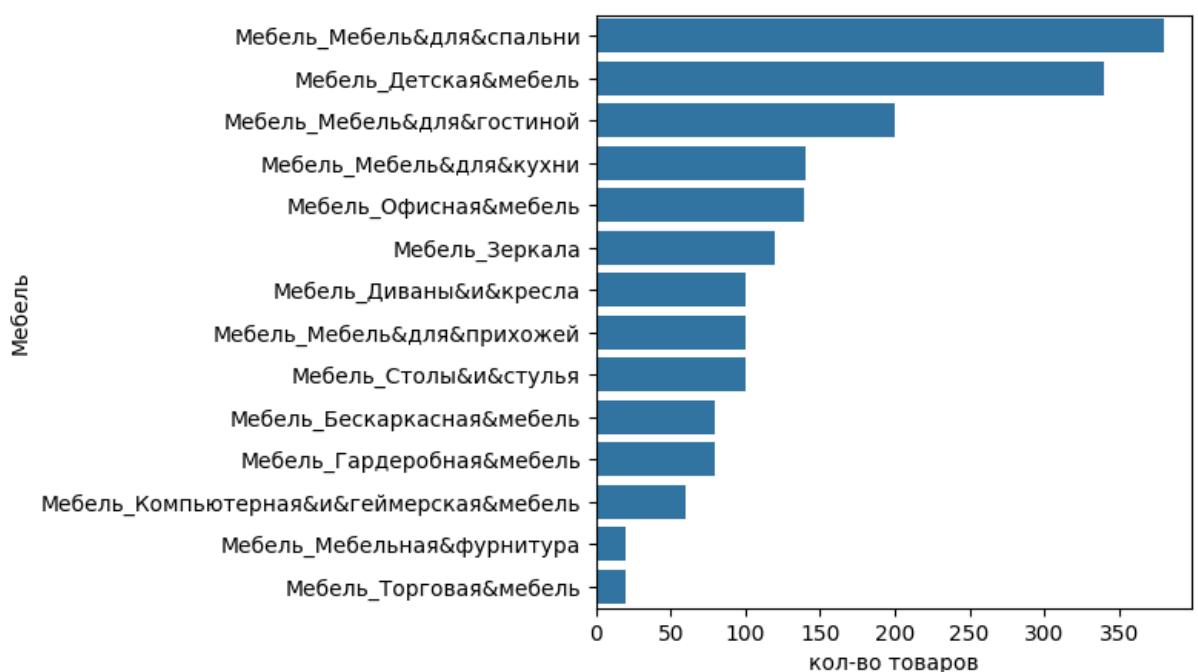


Рис. А.14: Распределение данных в категории «Мебель».

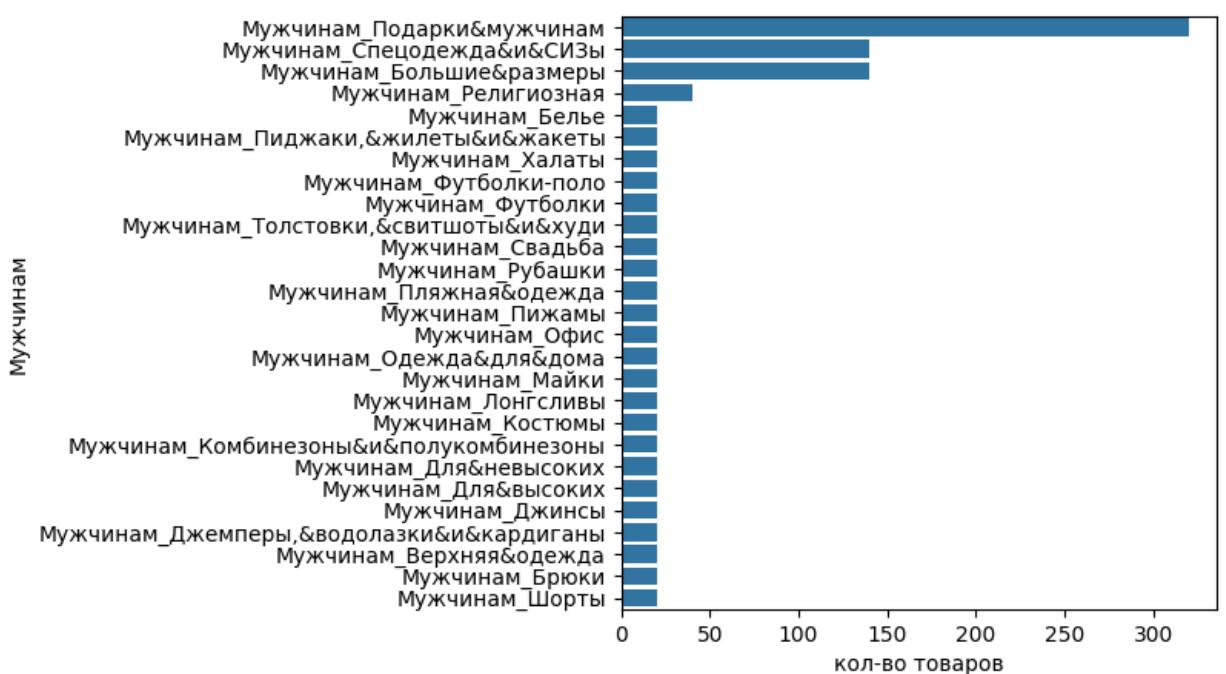


Рис. А.15: Распределение данных в категории «Мужчинам».

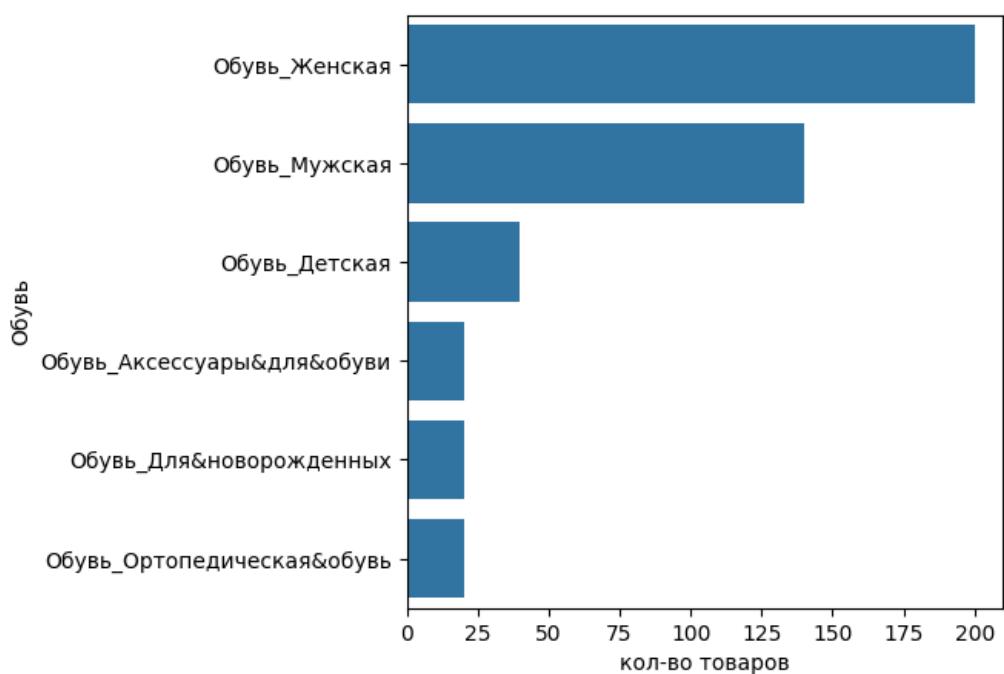


Рис. А.16: Распределение данных в категории «Обувь».

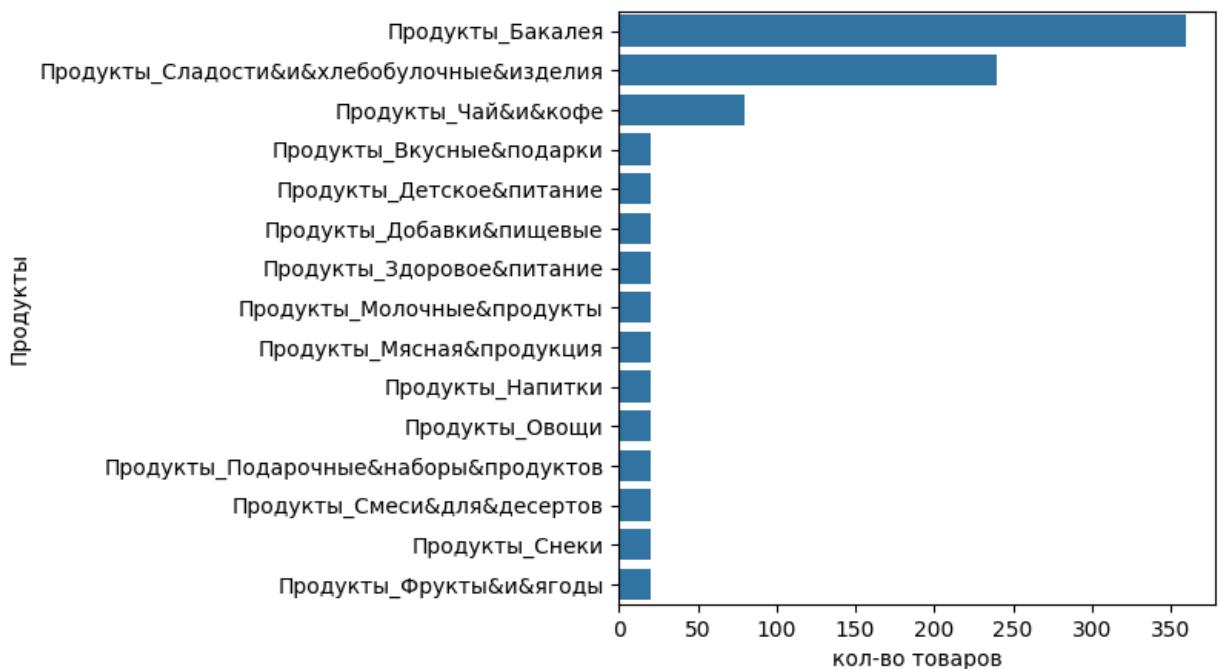


Рис. А.17: Распределение данных в категории «Продукты».

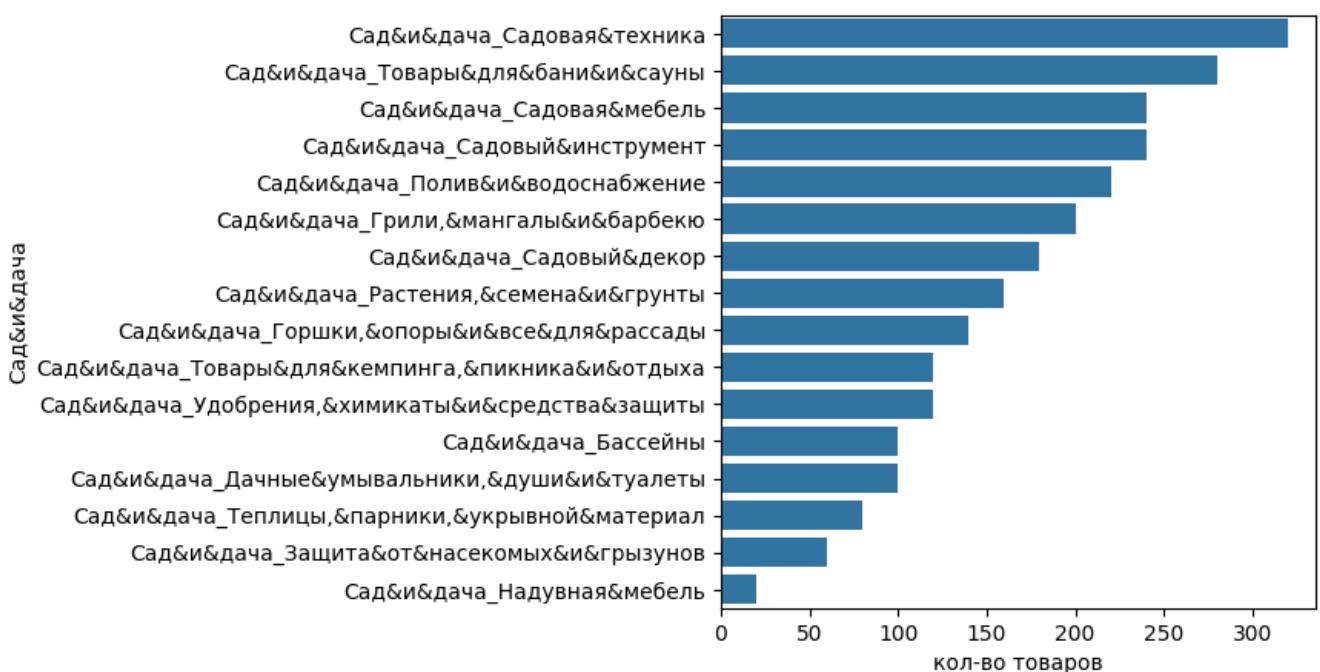


Рис. А.18: Распределение данных в категории «Сад&и&дача».

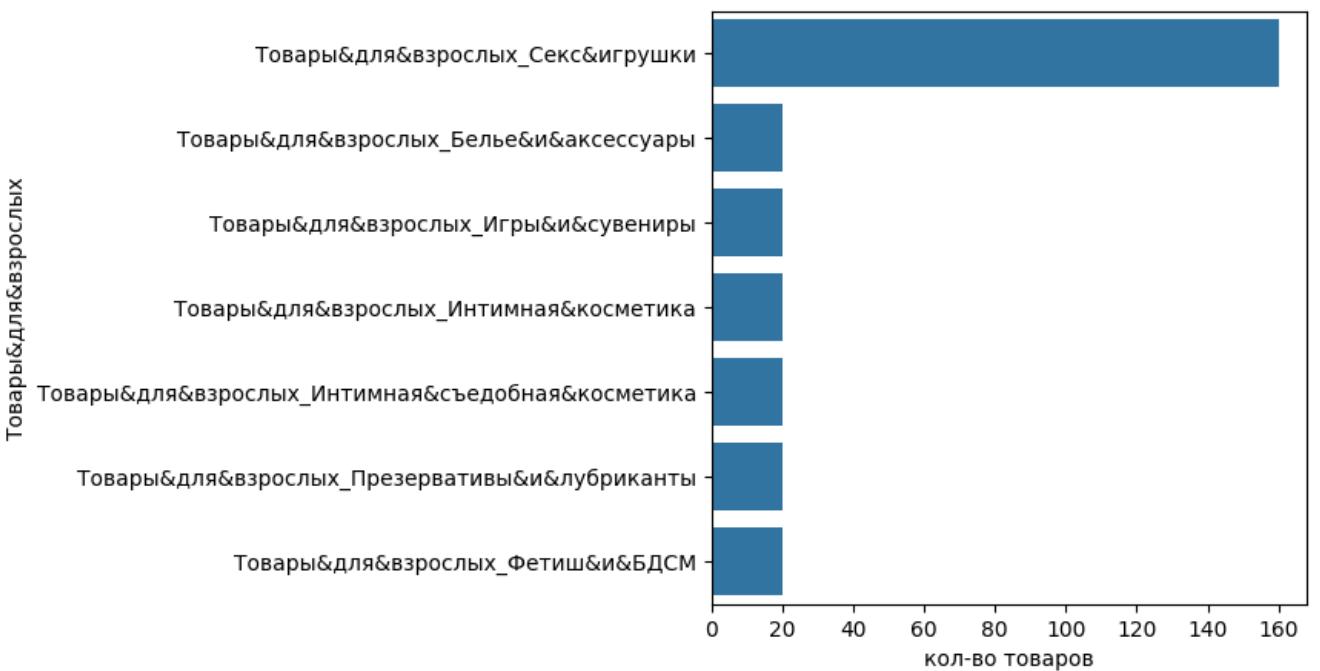


Рис. А.19: Распределение данных в категории «Товары&для&взрослых».

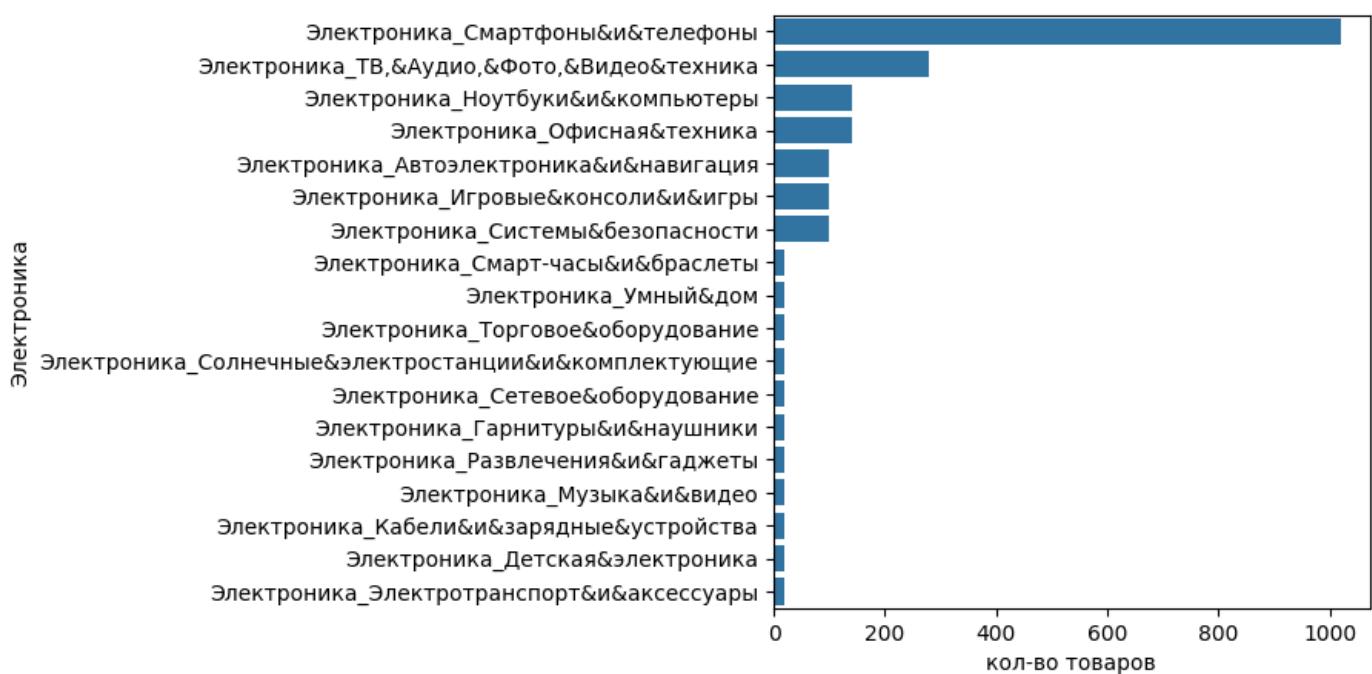


Рис. A.20: Распределение данных в категории «Электроника».

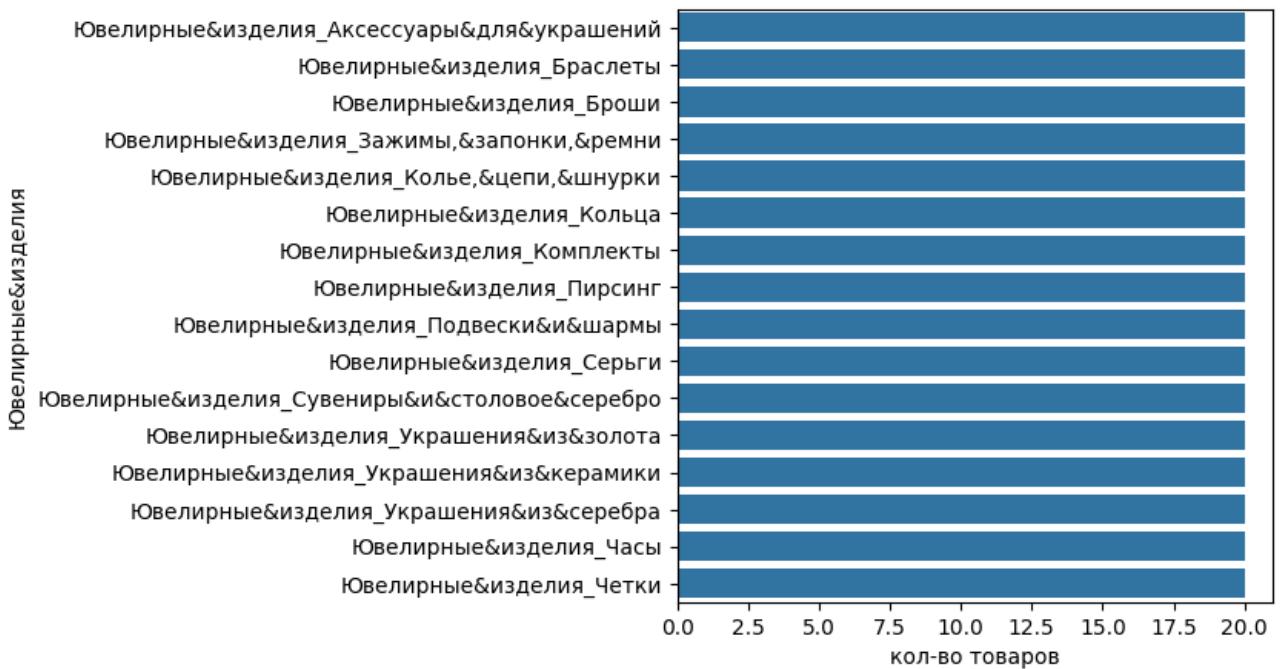


Рис. A.21: Распределение данных в категории «Ювелирные&изделия».