Федеральное государственное образовательное бюджетное учреждение высшего образования

«ФИНАНСОВЫЙ УНИВЕРСИТЕТ ПРИ ПРАВИТЕЛЬСТВЕ РОССИЙСКОЙ ФЕДЕРАЦИИ»

Факультет информационных технологий и анализа больших данных
Департамент анализа данных и машинного обучения

Выпускная квалификационная работа

на тему: «Создание метрики полезности продавцов и исследование ее для работы с сегментами»

Направление подготовки: 01.03.02 Прикладная математика и информатика

Профиль: Анализ данных и принятие решений в экономике и финансах

Выполнил студент учебной группы
ПМ19-1
Сухань Мария Александровна

Научный руководитель работы
доцент Департамента анализа данных и машинного обучения, к.э.н., доцент
Гринева Наталья Владимировна

ВКР соответствует предъявляемым требованиям:
Руководитель Департамента анализа данных и машинного обучения, к.т.н., доцент
_____Д.А. Петросов
«___» _____2023 г

Москва – 2023

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Сегодня маркетплейсы стремительно развиваются и становятся востребованной площадкой для продавцов и покупателей. В текущей ситуации значительная часть бюджетов e-commerce распределяется на решения и поддержку сервисов, удовлетворяющих покупателей. Однако не менее важно грамотно работать с продавцами, потому что успех, прибыль и репутация площадки зависит от качества товаров и услуг, предоставляемых ими.

Актуальность темы исследования

Согласно исследованию рынка маркетплейсов 2022 года [40], а также работе Е.Б. Габаловой [12], сегмент e-commerce растет стремительно, при этом продавцы не всегда понимают, какие факторы влияют на ликвидность. Разработка рекомендательной системы и комплекса мероприятий на основе оценки полезности продавца потенциально позволит улучшить качество сервиса, позитивно повлиять на опыт потребителей, помочь площадке усилить поддержку селлеров и взаимодействие с ними, а также повысить конкурентоспособность и репутацию. Таким образом, предложенное и проанализированное в работе решение становится актуальным и востребованным на рынке, поэтому российские компании начинают активно развивать это направление. Например, компания Avito формирует команду Selling Coach, а компания Ozon разрабатывает собственную рекомендательную систему для продавцов.

Цель и задачи работы

Целью работы является создание метрики полезности продавца для дальнейшего построения рекомендательной системы и разработки комплекса мер по работе с сегментами продавцов.

Задачи работы:

1) Изучить особенности и характеристики маркетплейсов, теоретические основы продуктовых метрик и алгоритм сегментирования рынка;

2) Проанализировать выбранные показатели и провести разведочный анализ данных для определения пороговых значений;

3) Определить функцию полезности продавца как совокупность метрик, отражающих не только прибыль, но и качество услуг;

4) Разработать модель рекомендаций для повышения качества предлагаемых продуктов и услуг;

5) Провести анализ дальнейшего применения полученных результатов и оценить возможный эффект от внедрения выбранных мер;

Объект и предмет исследования

Объектом исследования являются основные показатели продавцов в компаниях-маркетплейсах. Предметом исследования являются методы машинного обучения, разведочный анализ данных и алгоритмы создания рекомендательных систем.

Методологическая база

Методологическая база включает в себя анализ работ в области создания рекомендательных систем, исследования рынка российского e-commerce и машинного обучения: И.Д. Елина, Е.Б. Габалова, Максимовой Н.Б., В.Ф. Володько, Я.Ю. Старовойтова, С. Дибба, Л.Симкина, Васильевой А.С. и Латыповой А.Э. Кроме теоретического обзора литературы для достижения цели выпускной квалификационной работы использовались следующие методы: разведочный анализ данных, предобработка датасетов и оценка качества предиктивных моделей.

Научная новизна

Несмотря на возрастающую актуальность и интерес к теме крупных российских компаний, решение еще не реализовано на российском рынке e-commerce. Проанализированная в работе методика позволяет оценить эффективность работы продавцов, учитывая не только объем продаж, но и другие факторы, такие как доля вовремя доставленных постингов, количество товаров с лучшим индексом цен, рейтинг и так далее. Таким образом, работа является оригинальной и имеет научную новизну, так как представляет новый

подход к оценке полезности продавцов для разных сегментов и к созданию рекомендательной системы для продавцов.

Апробация результатов исследования:

Результаты исследования были представлены на 14-ой Международной научно-практической конференции студентов и аспирантов (НИУ ВШЭ 18 мая 2023, г. Москва). Тема доклада: «Создание метрики полезности продавцов и исследование ее для работы с сегментами».

Практическая значимость

Практическая значимость работы заключается в возможности применения предложенного алгоритма оценки полезности продавца для создания рекомендательной системы и разработки комплекса мер, повышающих качество услуг и потенциальную прибыль площадки.

Структура работы

Выпускная квалификационная работа состоит из введения, трех глав, заключения. В первой главе представлен анализ особенностей рынка маркетплейсов, исследование основных этапов жизненного цикла продавца и обзор подходов к сегментации пользователей. Вторая глава включает в себя практическую реализацию создания метрики полезности продавцов через тестирование различных подходов к формированию условий и определению весов показателей. Завершается исследование анализом дальнейшего применения полученных результатов и оценкой точности ML модели для создания рекомендательной системы. Работа состоит из 80 страниц, включая приложение, 3 таблиц, 21 рисунка и приложения. Список использованных источников включает 40 наименований.

Глава 1. Теоретические основы продуктовых метрик и сегментация рынка

1.1 Принцип устройства маркетплейса и жизненный цикл продавца на площадке

Сегодня Интернет является неотъемлемой частью жизни. По данным компании «We Are Social» [39] почти два миллиарда человек в мире совершают покупки через Интернет на постоянной основе, что делает его важным пространством для продвижения товаров, услуг и увеличения продаж. Несмотря на то, что интернет-магазины являются популярной формой коммерции, с увеличением их количества покупатели становятся все более требовательными к выбору площадки. Потребители хотят быстро, выгодно и удобно приобретать различные товары в одном месте за пару кликов, но даже самые большие торговые центры в мире не могут предложить такой ассортимент товаров, как, например, маркетплейс. В связи с этим, опираясь на исследования Е.Б. Габаловой [12], можно утверждать, что маркетплейсы являются популярной формой Интернет-торговли из-за наличия большого количества преимуществ как для продавца, так и для покупателей и глобального бренда в целом.

Согласно исследованию С.П. Гурской [14] маркетплейсы растут почти в двадцать раз быстрее, чем интернет-магазины. Маркетплейсы предоставляют возможность просмотреть различные товары одной категории в одном месте для сравнения поставщиков, качества, условий доставки и цены. К тому же лояльность клиентов к такой площадке выше, чем у по отдельности у маленьких магазинов с небольшой базой клиентов. Это также косвенно подтверждает тренд ухода от брендов, так как потребители от модели поиска конкретных марок переходят к модели выбора товара по совокупности характеристик и наличию честных положительных отзывов в глобальной сети. Таким образом, тенденции современного мира приводят к росту и расширению маркетплейсов, поэтому неудивительно, что, например, компания OZON выбрала стратегией вложение бюджета в стремительное

расширение, наращивание точек и увеличение числа пунктов выдачи. Согласно статье Максимовой Н.Б. [24], посвященной исследованию рынка маркетплейсов в 2022 году, компания Ozon заняла шестое место среди крупнейших онлайн маркетплейсов в Европе.

Разберемся с понятием подробнее. Маркетплейс – это торговая площадка, на которой совершаются покупки и продажи товаров и услуг. Основными задачами площадки являются: увеличение продаж, запуск бизнесов, упрощение логистики, расширение географии, продвижение товаров и услуг, привлечение трафика. В основе работы лежит одна из трех основных моделей бизнеса: C2C (customer-to-customer), B2B (business-to-customer) или B2B (business-to-business).

Идея маркетплейса получила новый виток развития в двухтысячные годы после выхода книги Криса Андерсона «Длинный хвост. Новая модель ведения бизнеса» [3]. В книге подтверждалось, что залогом увеличения продаж, в первую очередь, является широкий ассортимент. То есть, чем больше товаров различных категорий будет предложено магазином, тем выше шанс, что покупатель что-то купит. Это подтверждает и тот факт, что маркетплейсы сейчас отвечают почти за половину всего ритейла в мире. [40]

В подобной модели бизнеса три заинтересованных участника: покупатели, продавцы и компания-создатель маркетплейса. Для покупателя выгода состоит в том, что он может быстро получить конкретное предложение, сравнив стоимость, качество и условия доставки аналогичных товаров. Выгода для продавцов заключается в выходе на широкую аудиторию и возможности продавать товар без затрат на содержание магазинов, большого штата сотрудников и оплаты дорогой аренды. Для компании-создателя маркетплейса – это возможность зарабатывать с комиссий продавцов, а также за счет внешней, внутренней рекламы, тарифов за продвижение товаров и продаж собственной продукции.

Обычно алгоритм работы маркетплейса выстроен по следующей модели:

1) Продавец привозит товар на склад маркетплейса, фулфилмент или собственный склад (зависит от схемы продаж, выбранной продавцом)

2) Продавец через личный кабинет публикует фото и описание товаров на сайте маркетплейса

3) Используя различные инструменты, глобальная площадка продает и доставляет товары клиенту

4) По определенной договором периодичности партнер перечисляет продавцу деньги

Стоит также отметить, что маркетплейсы предлагают различные форматы сотрудничества продавцам и имеют особенности организации логистики в зависимости от выбранных схем, провайдеров, регионов, условий и договоренностей. То есть по сути площадка является связующим звеном между клиентом (покупателем) и исполнителем (продавцом).

Для наглядности, на основе изученных исследований, статей и электронных ресурсов, изобразим принцип работы маркетплейса схематически (рис.1):

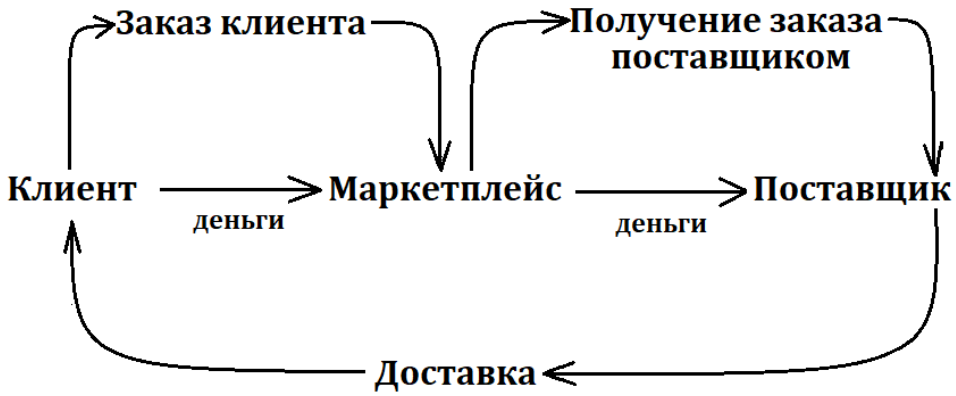

Рис.1 Схема работы маркетплейса

Источник: составлено автором

Продавец (поставщик) является ключевым звеном в схеме работы маркетплейса, поэтому важно понимать жизненный цикл продавца на площадке. Согласно книге Д. Б. Берга Е. А., Ульяновой, П. В. Добряка [4], модель полного жизненного цикла отдельного объекта представляет собой

описание последовательности всех фаз и этапов его существования от замысла и появления («рождения») до исчезновения («отмирания»).

Существуют разные модели жизненного цикла. Рассмотрим одну из самых популярных. Согласно внутренним исследованиям OZON и принятой в компании модели жизненный цикл продавца на площадке состоит из 4 этапов:

1) Лидогенерация – то есть процесс привлечения новых продавцов при помощи маркетинговых инструментов (например, лендинг Ozon.Starat или бустинг новичков). Это важный начальный этап для наращивания базы продавцов. Согласно книге К. Андреевой «Лидогенерация. Маркетинг, который продает» [1], лидогенерация позволяет понимать, например, какая оптимальная «стоимость» одного потенциального клиента (в данном случае продавца), сколько времени занимает его поиск, какая часть дойдет до продаж. Поэтому лидогенерация — это важный начальный этап для наращивания базы продавцов. Здесь важно понимать, какие каналы привлечения наиболее эффективные и откуда приходят продавцы: другие схемы, маркетинговые акции, продукты для продавцов и так далее. На этом этапе либо продавец сам приходит на площадку, либо площадка через определенные инструменты привлекает продавца. Основные цели этапа: привлечь большое количество качественных продавцов и не потерять продавцов.

2) Онбординг - совершение первого целевого действия на площадке (регистрация, загрузка документов, выбор схемы сотрудничества). Онбординг определяет, дойдет ли селлер до продажи или нет, поэтому этот этап является очень важным в жизненном цикле продавца. Процесс прохождения онбординга также помогает выявить слабые места в продукте, например, в рамках работы аналитиком в компании OZON, мной была построена воронка прохождения онбординга продавца от этапа создания склада в настройку метода на склад, что позволило выявить неконверсионные и сложные для селлера этапы, то есть понять, какие новые продуктовые изменения следует вводить и на что обращать внимание бизнесу. Маркетплейсу выгодно, чтобы как можно больше продавцов прошли онбординг, так как если продавец

сможет разобраться с тем, как завести склад, пролить стоки и так далее, тем более вероятно он дойдет до своей первой продажи, то есть принесет прибыль маркетплейсу. Помимо этого, в процессе онбординга компания может рекламировать платные услуги, например, рекламу и продвижение карточек.

Соответственно, на этом этапе продавец регистрируется и настраивает свой личный кабинет. Основными целями этого этапа для площадки являются получение максимальной конвертации из регистрации в получение денег и минимизирование времени от регистрации до первого получения денег.

3) Работа продавца на площадке, то есть реализация заказов, использование фичей, продвижение карточек и другое. Здесь площадка пытается упростить этапы работы для продавцов и «выкатить» как можно больше продуктов, упрощающих работу продавцов. На этом этапе продавец осуществляет продажи и поставки товаров через площадку. Целями этого этапа для площадки являются максимизация валовой стоимости товаров продавца и минимизация процента оттока продавцов.

4) «Смерть» или отток продавца с площадки. Это завершающий этап жизненного цикла продавца, поэтому площадке важно понимать, из-за каких причин и с какой частотой продавцы переходят в разряд «мертвых», так как большие потери продающих продавцов могут сильно сказаться на обороте и чистой прибыли. Цель для маркетплейса на этом этапе – вернуть неактивных продавцов.

В целом для всех этапов основной задачей маркетплейса является удержание как можно большего числа продающих продавцов, при этом для площадки важно не терять общее качество товаров и сервиса, для чего нужно отслеживать и оценивать полезность каждого селлера. Понимание коэффициента полезности продавца помогает разделять продавцов на различные сегменты и использовать нужные инструменты для продвижения и работы с каждой группой.

1.2. Ключевые метрики работы продавцов и их влияние на показатели маркетплейса в целом

Продавцы на площадке предлагают продукты и услуги, которые ищут покупатели, соответственно, чем больше активных поставщиков на площадке, тем больше разнообразие товаров и услуг на платформе. Кроме того, от продавцов напрямую зависит качество товаров и услуг. Хорошие поставщики стремятся предложить лучшую продукцию и качественный сервис за более низкие цены, что не только улучшает опыт покупателей, но и делает маркетплейс более конкурентоспособным на рынке.

Продавцы также могут влиять на репутацию маркетплейса. Если качество продукции не соответствует описанию, клиенты будут недовольны и потеряют доверие к маркетплейсу в целом. Поэтому площадка должна следить за поведением продавцов, чтобы поддерживать высокий уровень качества продукции и доверия к платформе. В этом помогает анализ основных метрик продавцов и их ключевых конверсий, который также позволяет понимать динамику прогресса маркетплейса и скорость роста площадки.

В качестве основных метрик продавцов обычно рассматривают: ARPU, AOV, AIV, OPC, AOQ, GMV и так далее. Однако подробнее в исследовании стоит остановиться на тех показателях, которые наиболее полно определяют успешность продавца, а значит, потенциально могут быть использованы в формуле расчета коэффициента полезности. Кроме того, эти метрики являются целевыми для любого маркетплейса, так как площадке важно следить за двумя ключевыми показателями: выручкой и качеством сервиса. Рассмотрим научные работы и обзоры, достоинства, недостатки и проблемы метрик, выявленные в ходе аналитических и научных исследований.

1) Отзывы покупателей и оценка товаров — это одна из наиболее информативных метрик для определения полезности продавца. Положительные отзывы свидетельствуют о хорошем обслуживании и высоком качестве товара, а отрицательные - о проблемах, с которыми встречаются покупатели. Однако, если в оценке полезности продавца

ориентироваться только на этот показатель и не рассматривать другие метрики, то легко допустить неверные суждения по следующим причинам:

- Нечестные оценки. Продавец может самостоятельно создавать себе положительные отзывы с разных аккаунтов или удалять отрицательные отзывы при наличии такого функционала;
- Различное количество продаж. Рейтинг продавца определяется как среднее среди всех оценок покупателей, при этом количество оценок у крупных и мелких продавцов будет различным. То есть, если у крупного продавца будет один негативный отзыв из пары сотен, то рейтинг никогда не достигнет оценки в 5.0, при этом для мелкого продавца с одной оценкой в 5.0, средний рейтинг будет определяться как 5.0;
- Ограниченные варианты оценки. Некоторые платформы предоставляют только несколько вариантов для оценки продавца, например, положительный, нейтральный или отрицательный отзыв. Это может ограничивать возможности клиента в оценке качества услуг продавца;

2) Количество продаж и валовая стоимость товара (Gross merchandise value, GMV) является важной метрикой в оценке полезности по следующим причинам:

- покупатели склонны выбирать продавцов с большим количеством продаж и узнаваемым брендом;
- крупные продавцы считаются более полезными для площадки в целом, так как увеличение объема продаж означает увеличение дохода маркетплейса. Таким образом, Wildberries давно придерживается стратегии улучшения функционала и поддержки именно крупных продавцов. Ozon, согласно внутренним исследованиям компании, тоже начинает активно выходить на эту стратегию;
- высокие показатели продаж могут означать, что продавец эффективно общается с клиентами, использует маркетинговые инструменты, участвует в акциях, продвигает продукты, предоставляют полезные и

востребованные товары для покупателей, а значит, может потенциально обеспечить рост компании.

Однако несмотря на все достоинства этого показателя, его может быть недостаточно для точной и «честной» оценки полезности продавца по следующим причинам:

- GMV не учитывает удовлетворенность покупателей, качество сервиса и товаров продавца, а также уровень удержания клиентов;
- GMV не отражает количество продаж: сумма продаж у высоко востребованного продавца с дешевыми товарами может быть равна одной продаже крупного продавца с дорогим товаром;

Таким образом количество продаж является существенным показателем в оценке полезности продавца. Зачастую GMV даже может выступать в качестве самостоятельной метрики полезности, однако общая и «справедливая» оценка полезности продавца должна основываться на более широком наборе метрик, включая уровень обслуживания клиентов, качество работы, скорость доставки, процент возвратов, наличие знаний о продукте и так далее.

3) Возврат товаров - метрика, которая свидетельствует о качестве товара и качестве обслуживания. Этот показатель играет важную роль в оценке полезности продавца по следующим причинам:

- количество возвратов может быть косвенным показателем качества товаров и правильной обработки заказов продавцом;
- высокое количество возвратов и отмен заказов может повлиять на доверие покупателей к маркетплейсу и к снижению индекса NPS, что в свою очередь может привести к уменьшению показателей конверсии и посещаемости сайта в целом. В условиях рынка это может сместить выбор пользователей в сторону конкурентов;
- в некоторых схемах доставки стоимость возвратов (обратный поток) оплачивает площадка, а значит, большой процент возвратов отрицательно сказывается на чистой прибыли маркетплейса;

Таким образом, уровень возвратов может использоваться в качестве метрики полезности продавца, так как он может отражать не только качество товаров, но и качество обслуживания, понимание потребностей клиентов и точность описания товаров. При этом важно помнить, что возвраты могут быть вызваны различными причинами, включая неправильный размер, неудобный фасон, неожиданные расходы, повреждения в процессе доставки и так далее.

4) Процент заказов, выполненных вовремя, процент просрочек и отмен. Это важные селлерские показатели, так как просрочки и отмены напрямую влияют на качество пользовательского опыта, а значит, площадке в глобальном смысле важно следить за репутацией и качеством сервиса. Безусловно, отмена заказа покупателем может происходить по разным причинам, но слишком высокий процент отмен может свидетельствовать о том, что продавец неясно описывает товар, предоставляет сервис низкого качества или не соблюдает сроки доставки. Это может снижать доверие покупателей к продавцу и уменьшать вероятность повторных покупок. К тому же в ходе исследования процента отмен могут быть сделаны важные продуктовые выводы, например, при большом проценте отмен можно обратить внимание на карточку товара и найти причину, по которой покупатели совершают отмены или проанализировать скорость и сроки доставки.

5) Индекс цен является важной метрикой не только для продавца, но и для площадки в целом по ряду причин. Во-первых, если аналогичный товар продавец выставляет на разных площадках по разной цене, это может свидетельствовать о плохих продуктовых решениях и низком качестве селлерского сервиса: возможно, продавцы выставляют товар дороже на определенной площадке из-за высоких тарифов и комиссий, что в дальнейшем может стать причиной высокого оттока продавцов. Во-вторых, индекс цен во-многом является ключевым фактором при принятии решения о покупке: если пользователь будет видеть два аналогичных товара на различных

маркетплейсах по разным ценам, вероятнее всего, выбор будет сделан в пользу площадки с более низкой ценой. Следовательно, индекс цен является важной метрикой в оценке полезности продавца для площадки.

Подводя итог, можно утверждать, что показатели продавцов напрямую влияют на показатели маркетплейса в целом. Если продавцы на площадке успешны, а продукты и услуги, которые они предлагают, соответствуют ожиданиям покупателей, то пользователи будут оставлять больше положительных отзывов и оценок, что приведет к улучшению рейтинга маркетплейса. Это повысит уровень доверия клиентов к площадке, что, в свою очередь, приведет к привлечению большего количества покупателей, увеличению объемов продаж и выручки.

С другой стороны, если продавцы маркетплейса предлагают продукты и услуги низкого качества, то это может привести к ухудшению репутации площадки и потере как существующих клиентов, так и потенциальных покупателей. В результате, меньше человек будут посещать сайт площадки, что в свою очередь приведет к снижению объемов продаж и выручки маркетплейса.

Таким образом, показатели продавцов имеют прямое влияние на успех площадки в целом и являются ключевым фактором ее роста и развития. Значит, маркетплейс должен устанавливать высокие требования для продавцов, аккуратно отслеживать метрику их полезности (которая включает как показатели качества, так и показатели продаж) и обеспечивать поддержку, систему рекомендаций и обратную связь, чтобы помочь продавцам улучшить продукты, качество сервиса и услуги.

1.3. Основные принципы и критерии сегментирования продавцов

Сегментирование – это разделение пользователей на группы по совокупности определенных признаков.

В процессе исследования сразу возникает вопрос: зачем бизнесу нужна сегментация? Проведённые анализы онлайн-университета InSales доказывают, что чем больше конкурентов у компании (в частности, у маркетплейса или у интернет-магазина), тем более детально следует сегментировать аудиторию, включая как покупателей, так и продавцов, потому что узкие потребительские группы помогают создать и настроить наиболее эффективную и полезную для бизнеса персонализированную воронку.

Опираясь на статью В.Ф. Володько «Сегментирование и позиционирование на рынке» [11], можно выделить следующие преимущества сегментирования на рынке:

1) Более эффективное использование ресурсов и улучшение целевых действий воронки. Это значит, что компании могут более результативно использовать свои ресурсы, так как они могут сосредоточиться на наиболее выгодных сегментах рынка и сократить затраты на маркетинг для менее выгодных сегментов. В разрезе продавцов это позволяет понять, какие показатели стоит улучшать и какие стратегии выбрать для продвижения и улучшения работы каждой группы, чтобы повысить конверсию в первую продажу и увеличить объём выручки в целом;

2) Лучшая конкурентоспособность. Это значит, что компании могут создавать продукты и услуги, соответствующие потребностям конкретных сегментов, то есть улучшать опыт пользователей. А чем более удобные для пользователя услуги предлагает площадка, тем выше вероятность выбора ее покупателем по сравнению с конкурентами. Кроме того, работа с каждым сегментом и понимание конкретных особенностей каждой группы продавцов позволяет предприятиям адаптироваться к быстро меняющимся требованиям рынка, постоянно совершенствоваться и быть успешными в долгосрочной перспективе.

3) Увеличение лояльности потребителей продукта (в частности продавцов), удержание базы и повышение репутации. Благодаря разбиению и целенаправленному подходу к каждой группе, многие продавцы могут эффективнее реализовывать свои товары и услуги. К тому же продавцы будут лояльными к площадке, а значит, будут распространять про нее положительные отзывы, что будет позитивно сказываться на репутации площадки.

4) Увеличение прибыли: правильное разбиение пользователей на группы позволяет выбрать наиболее эффективную для конкретных условий стратегию, что ведет к росту продаж и увеличению GMV. Кроме того, понимание проблем и особенностей продавцов в каждом сегменте может способствовать выявлению потребностей в определенных категориях товаров и услуг. Эту информацию можно использовать, чтобы расширить ассортимент, улучшить качество сервиса, привлечь новых клиентов и заработать дополнительный доход на рекомендациях для продавцов и продвижениях их карточек товаров.

Однако стоит отметить, что Филип Котлер [6] в своей работе писал, что единого метода сегментирования не существует. Несмотря на то, что данный тезис был выдвинут более двадцати лет назад, универсальный метод все еще не был разработан. По мнению А.П. Карасева [18], в отечественной литературе в недостаточной степени рассмотрены проблемы понятия сегментирования, поскольку зачастую наблюдаются расхождения в терминологиях и описании технологий и этапов процесса сегментирования.

Однако, опираясь на работы Д.Г. Свечкаревой [31], Кирилловой Л.К.[19], Ж.Р. Габбасова [13], можно составить следующий алгоритм сегментации:

1) Постановка цели. Для максимально эффективной сегментации необходимо первым шагом определить конкретные цели в зависимости от текущей ситуации и задач компании, то есть ответить на вопрос «Зачем проводить сегментацию?». Например, для улучшения показателей качества

или для увеличения продаж. Понимание целей и стратегии компании помогает так же правильно определить веса каждой метрики в формуле и обозначить приоритетные показатели;

2) Выбор критериев. Необходимо правильно подобрать метрики для определения коэффициента полезности, чтобы определить значимые для бизнеса критерии. Это является основой для расчета, оценки и анализа коэффициента полезности продавца. Если метрики выбраны неправильно, это может привести к ошибкам при оценке производительности или неправильному распределению бонусов. Грамотно подобранная комбинация метрик обеспечивает более точную картину производительности продавца, которая отражает как его качественную, так и количественную работу и позволяет лучше оценить вклад каждого продавца в компанию и ее достижения. Она также дает возможность компаниям сравнивать полезность разных продавцов и использовать эти данные для улучшения своих бизнес-стратегий;

3) Определение метода сегментации. Существуют разные подходы к разделению пользователей на группы, которые зависят от фокуса бизнеса и от того, какие группы продавцов привлекательны для маркетплейса. Правильное определение метода сегментации позволяет более точно выделить целевую аудиторию и улучшить стратегию, а также помогает узнать, какие конкретные факторы влияют на поведение и потребности продавцов в каждой группе. Определив логику подсчета коэффициента полезности продавца, можно разбивать покупателей на группы, например, по различным численным промежуткам;

4) Описание каждого сегмента. Необходимо понимать критерии групп для выбора правильной стратегии для каждого сегмента. Этот этап помогает определить целевую аудиторию бизнеса и разработать маркетинговую стратегию, которая будет нацелена на конкретные нужды продавцов. В итоге, описание каждого сегмента рынка помогает бизнесу повысить свою эффективность и конкурентоспособность. Кроме того, важно

определить набор признаков, которые наиболее полно и однозначно будут отражать показатели группы;

5) Подготовка рекомендаций. Исходя из выделенных проблем и потребностей аудитории можно сформулировать комплекс мероприятий для достижения поставленных целей. Например, для продавцов с высоким коэффициентом полезности, можно усовершенствовать модель взаимодействия для быстрых коммуникаций и подготовить выгодные оптовые тарифы; продавцам со средним рейтингом предлагать промо, акции и продвижение карточек товаров, а для продавцов с низким коэффициентом полезности ввести систему регулирований и штрафов. Кроме того, на основании показателей успешных продавцов можно создать рекомендательную систему с комплексом мероприятий по улучшению работы с карточками или с показателями качества;

Таким образом, как было правильно отмечено в статье Н.Б. Изаковой [17], «важность процесса сегментирования в управлении деятельностью компании справедливо признается многими зарубежными и российскими учеными». То есть можно сделать вывод о значимости разделения пользователей на целевые группы, так как сегментация продавцов является одним из ключевых факторов успеха маркетплейсов, позволяющих улучшить пользовательский опыт, повысить качество продавцов, улучшить эффективность маркетинга и увеличить выручку. Для каждой группы клиентов создаются индивидуальные продукты, рекламные кампании и условия сервиса, что позволяет более эффективно работать с продавцами, а следовательно, и с клиентами.

Выводы к главе 1:

1) Основными задачами маркетплейса являются: увеличение продаж, упрощение логистики, расширение географии, продвижение товаров и услуг, привлечение трафика;

2) Жизненный цикл продавца состоит из четырех этапов: лидогенерации, онбординга, реализации заказов, оттока с площадки. Сильнее всего стоит отслеживать этап работы продавца на площадке, так как на этом этапе формируются ключевые показатели маркетплейса;

3) Показатели продавцов имеют прямое влияние на успех площадки в целом и являются ключевым фактором ее роста и развития;

4) Для формирования честной и справедливой оценки полезности продавца следует использовать следующие метрики: GMV, доля заказов, доля возвратов, индекс цен, доля вовремя доставленных постингов;

5) Для грамотного сегментирования стоит определить цель, выбрать информативные критерии, определить правило разделения пользователей на группы, описать каждый сегмент и подготовить рекомендации;

Глава 2. Анализ показателей и проведение exploratory data analysis для работы с данными

2.1 Построение распределения метрик и проведение «разведочного анализа» данных каждого показателя

Разведочный анализ данных (Exploratory data analysis или EDA), согласно книге «Аналитическая культура» [2] — это определяющий этап проведения любого аналитического исследования. EDA позволяет предобработать данные: очистить их от аномальных выбросов и создать информативный набор для дальнейшего построения гипотез. Целью такого анализа являются обнаружение проблемных данных и отклонений; выявление основных структур, определяющих вид данных; выбор наиболее значимых переменных; построение допущений для статистических выводов; проверка гипотез; исследование причин и следствий наблюдаемых явлений и разработка начальных моделей. Все это позволяет провести грамотную предобработку данных, минимизировать дальнейшие ошибки и исключить неверные расчеты и результаты. Чтобы проверить влияние каждого фактора на общую формулу и оценить распределения, проведем EDA исследованных в первой главе переменных.

Для проведения разведочного анализа данных был выбран датасет с основными показателями продавцов маркетплейса «Ozon» (над реальными данными были проведены операции масштабирования с целью сохранения политики конфиденциальности данных). В исходном наборе данных пять основных характеристик продавцов, влияющих на показатели полезности и 137086 строки. Так как каждая строка – это уникальный идентификатор, то в выборке исследованы 137086 продавца.

Для проведения анализа в датасете используются следующие столбцы:

1) CompanyID – номер компании, то есть уникальный числовой идентификатор в системе маркетплейса (для сохранения конфиденциальности данных идентификатор заменен на порядковый номер);

2) avg_GMV – средний GMV, взвешенный на количество рабочих дней на маркетплейсе за последний месяц. Средний показатель репрезентативнее по сравнению с абсолютным, так как он уравнивает метрики старых продавцов, проработавших полный месяц и новых продавцов, зарегистрировавшихся в течение последнего месяца. Кроме того, средние показатели более информативны в оценке полезности по сравнению с метрикой темпа прироста, так как на процент прироста может влиять большой набор факторов, таких как: сезонность, пандемии, покупательская способность, общая динамика рынка, крупные акции и так далее.

3) n_unique – количество уникальных товаров продавца с лучшим прайс индексом (PI) за последний месяц. Прайс-индекс – это показатель того, насколько цена конкретного товара отличается от средней цены на аналогичные товары на рынке. Товары с наименьшим прайс-индексом по сравнению с другими площадками считаются более привлекательными для покупателей, так как у таких товаров представлено лучшее соотношение цены и качества.

4) returns – процент возвратов за последний месяц (измеряется в шкале от 0 до 100). Рассчитывается как отношение количества товаров, возвращенных по вине продавца, ко всем купленным товарам.

5) rating – средний рейтинг (средняя оценка товаров) продавца за последний месяц. Рассчитывается как отношение суммы баллов отзывов на всех товарах к общему количеству отзывов. Измеряется в шкале от 0 до 5. В случае, если рейтинг равен нулю, отзывов на товаре не было из-за непопулярности товара или короткого срока нахождения продавца на площадке.

6) ontime – процент вовремя доставленных постингов за последний месяц (измеряется в шкале от 0 до 100). Рассчитывается как отношение количества вовремя доставленных постингов ко всем постингам продавца.

Сформулируем основную задачу построения модели полезности продавца: «Какие признаки с какими трешхолдами стоит взять, чтобы

определить продавца как полезного для площадки?». Трешхолд (threshold) — это пороговое значение, которое используется для принятия решений в различных областях, например, в машинном обучении, продажах, аналитике и так далее.

Для начала необходимо провести препроцессинг данных для очистки от дубликатов, выбросов и понимания распределений входных данных.

Реализуем следующие шаги:

1) Очистим исходный датасет от дубликатов с помощью функции drop_duplicates().

2) Удалим неинформативный столбец с номерами компаний с помощью метода drop(), чтобы не искажать корреляционные матрицы и методы определения весов коэффициентов. Кроме того, номер компании не связан с ее показателями, а значит, никак не влияет на исследуемые признаки.

3) Проанализируем сводную статистику (рис.2) с помощью функции describe()

	avg_GMV	n_unique	returns	rating	ontime
count	137086.000	137086.000	137086.000	137086.000	137086.000
mean	22626.647	206.536	3.415	4.263	80.328
std	110051.128	2913.412	9.697	0.701	28.768
min	0.000	0.000	0.000	0.000	0.000
25%	689.646	0.000	0.000	3.401	79.115
50%	2060.645	4.000	0.000	4.680	90.909
75%	11014.130	29.000	2.857	4.880	100.000
max	9707408.556	411993.000	100.000	5.000	100.000

Рис.2 Сводная статистика датасета

Источник: составлено автором

По сводной статистике можно предположить, что распределения всех показателей не являются нормальными, так как во всех метриках среднее значительно отличается от 50-ого перцентиля (медианы) и интервалы между квартилями значительно различаются. То есть, 25-ый и 75-ый перцентили несимметричны относительно 50-ого перцентиля. При этом в последствии для

построения предсказаний с целью повышения точности можно попробовать нормализировать данные, так как показатели имеют разные шкалы измерения: рейтинг определяется по пятибалльной шкале, GMV выражено в десятках тысяч, а онтайм и возвраты имеют процентные значения от нуля до ста. Также замечаем, что количество вхождений в каждой величине одинаково и равно количеству компаний (137086). Это означает, что в датасете нет строк с пустыми значениями; от которых его необходимо очистить.

5) Перед анализом выбросов необходимо проверить корреляцию признаков (рис.3), чтобы исключить величины, сильно зависящие друг от друга.

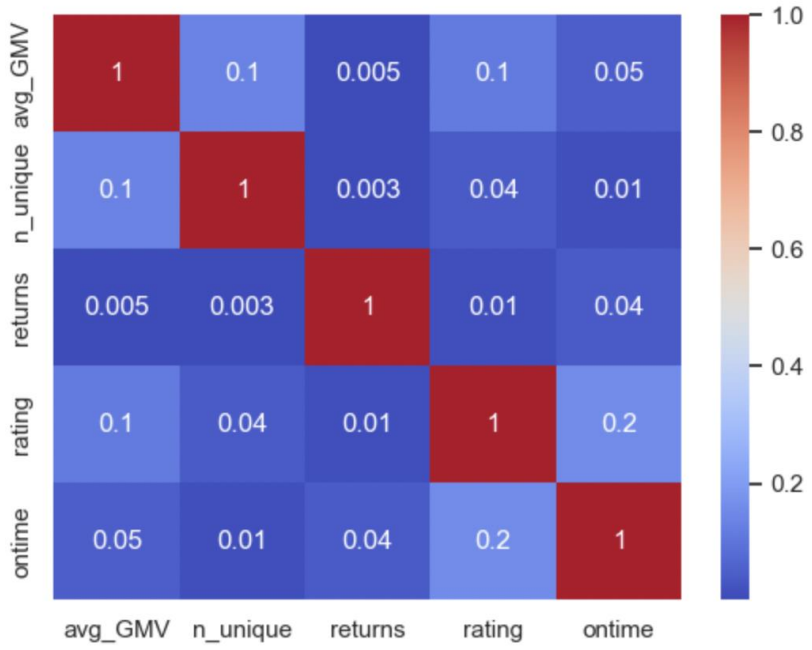

Рис.3 Матрица корреляций

Источник: составлено автором

Матрица корреляций имеет низкие значения. Это значит, что переменные $X_1, X_2 \ldots . X_n$ не оказывают сильного влияния друг на друга, то есть отсутствует мультиколлинеарность, что свидетельствует о правильно подобранных признаках, так как наличие мультиколлинеарности может привести к некорректным и непредсказуемым результатам при анализе данных и построении моделей. В частности, мультиколлинеарность может привести к проблемам неустойчивости оценок коэффициентов регрессии,

снижению точности прогнозирования и ухудшению интерпретации результатов.

Для того, чтобы построить модель классификации продавцов по признаку полезности для площадки, необходимо аналитически определить критерии метрик, по которым мы можно считать продавца полезным или неполезным.

Возьмем за таргет модели бинарное разделение продавцов на эталонных и не эталонных. Эталонными будем считать тех продавцов, которые соответствуют набору критериев полезного продавца — это лидеры по числу GMV, проценту онтайм, рейтингу, наличию товаров с лучшим PI и низкой долей возвратов. Если продавец соответствует этим критериям, мы присваиваем ему значение таргета True, если нет – False. Бинарное распределение удобно для использования, так как такой подход позволяет использовать логистическую регрессию для оценки влияния различных факторов на полезность продавцов.

Стоит отметить, что решение о полезности продавца масштабируемо, выгодно для бизнеса и бесплатно для продавцов. Оценка полезности помогает определить, какие продавцы являются прибыльнее и какие продукты более востребованы. Это может помочь маркетплейсу принимать эффективные решения о том, каких селлеров следует привлекать и какие продукты следует продвигать. Кроме того, эта метрика может помочь площадке определить, какие продавцы нуждаются в дополнительной поддержке и обучении для улучшения свои результатов и увеличения продаж. В целом, оценка полезности продавцов является важным инструментом для оптимизации работы маркетплейса и увеличения его прибыльности.

Главный из проанализированных в первой главе признаков – это GMV, который является одним из самых важных показателей полезности и успешности продавца на площадке. Проведем исследование этой метрики, для чего сначала построим график зависимости числа продавцов от порога GMV (рис.4).

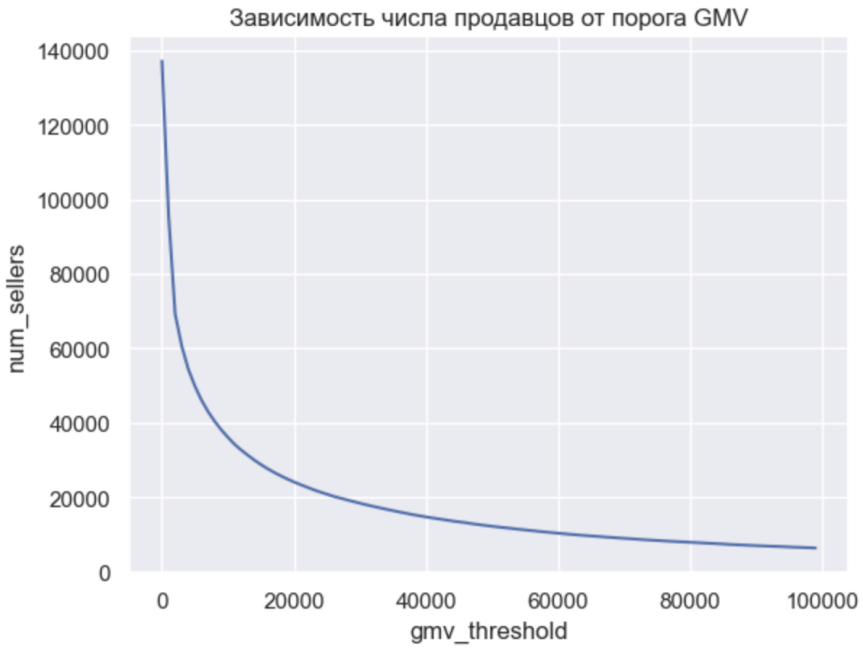

Рис.4 График зависимости числа продавцов от порога GMV

Источник: составлено автором

График представляет собой часть гиперболы. Как видно по графику, большая часть компаний имеет средний GMV до 20000 рублей. Кроме того, после достижения этого порога число компаний начинает резко снижаться. Попробуем определить границу точнее, для чего построим распределение числа компаний, попадающих в значения перцентилей (рис. 5)

	quantile	GMV_quantile	num_companies_higher	percent_companies_higher
0	0.0	0.00	137086.0	100.00
1	0.1	163.16	123380.0	90.00
2	0.2	460.47	109671.0	80.00
3	0.3	999.29	95960.0	70.00
4	0.4	1566.00	86948.0	63.43
5	0.5	2060.65	68543.0	50.00
6	0.6	3923.98	54834.0	40.00
7	0.7	7726.22	41126.0	30.00
8	0.8	16328.77	27418.0	20.00
9	0.9	44058.67	13709.0	10.00

Рис.5 Таблица распределения количества компаний от значений перцентилей

Источник: составлено автором

Таблица имеет поля: quantile – перцентиль; GMV_quantile – значение GMV, соответствующее данному перцентилю; num_companies_higher – количество компаний, лежащих выше значения перцентиля; percent_companies_higher – процент компаний, лежащих выше значения перцентиля. Исходя из полученных значений можно сделать вывод, что до 80-ого перцентиля значения GMV возрастают постепенно, после чего наблюдается резкий рост. Аналогичная ситуация происходит у метрики количества компаний: выше 70-ого перцентиля лежит почти в два раза больше компаний, чем выше 80-ого перцентиля, то есть можно предположить, что на 20% компаний, лежащих выше 80-ого перцентиля, приходится большая часть общего GMV.

Попробуем проставить успешность продавцов по исследуемой метрике с помощью ABC-анализа. ABC-анализ продавцов — это метод классификации продавцов на основе объема продаж. Он основан на принципе Парето, который утверждает, что 20% усилий приносят 80% результатов. Этот метод позволяет выделить наиболее важных продавцов и сконцентрировать на них внимание и ресурсы, чтобы максимизировать выручку и прибыль. Группа A составляет 20% продавцов, которые приносят 80% выручки. Согласно статье Васильевой А.С. и Латыповой А.Э. [10], в которой исследуются преимущества и недостатки ABC-анализа, данный метод является одним из самых практичных благодаря его простоте и универсальности использования, прозрачности этапов анализа и возможностям оптимизации ресурсов.

После выделения продавцов группы A («полезные продавцы»), посчитаем следующие показатели:

- Сумма GMV «полезных» продавцов = 2 773 276 898,595 рублей
- Сумма GMV «бесполезных» продавцов = 328 519 616,131 рублей
- Процент GMV «полезных» продавцов от общего GMV = 89,41%
- Процент «полезных» продавцов от общего числа продавцов = 20,0006%

Таким образом, 20% продавцов приносят почти 90% GMV площадки, соответственно эту категорию можно считать полезными для маркетплейса. В

заключении можно сделать вывод, что справедливым трешхолдом для GMV является 80-ый перцентиль (категория А продавцов по ABC анализу).

Проведем исследование признака ontime, для чего сначала построим график зависимости числа продавцов от порога ontime (рис.6)

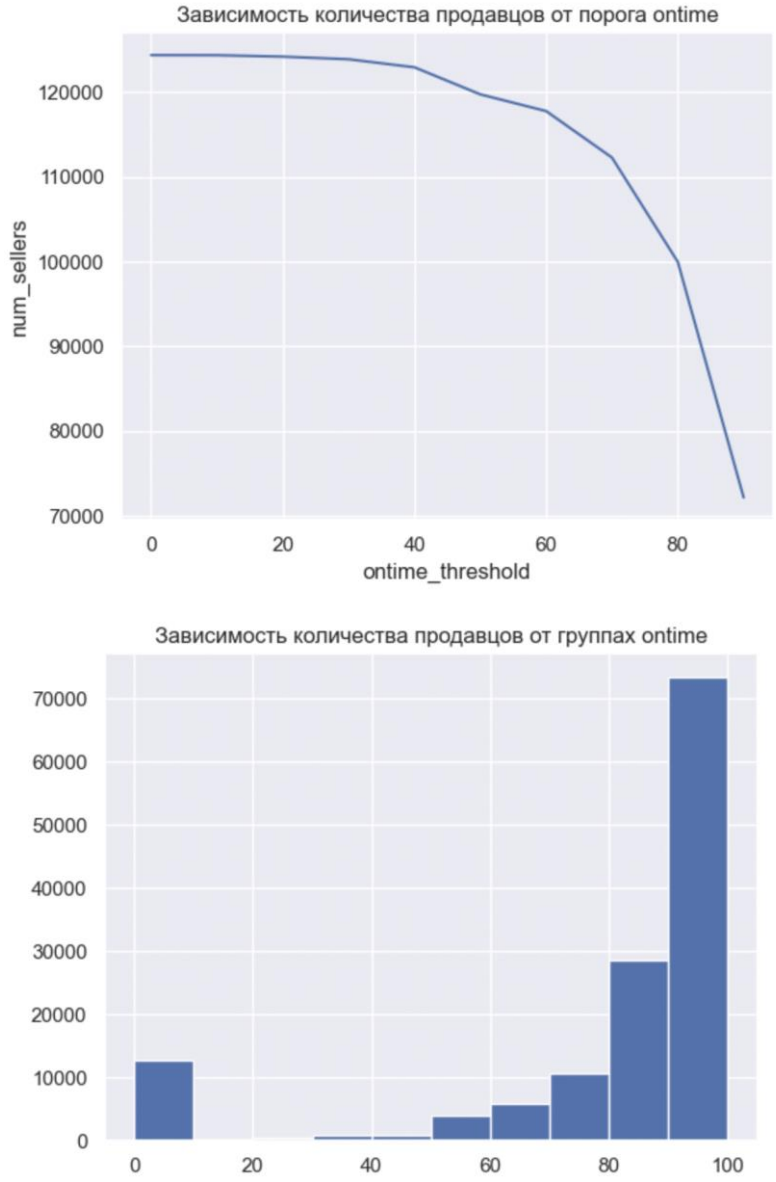

Рис.6 Графики зависимости числа продавцов от порога ontime

Источник: составлено автором

По графикам видно, что чем выше порог ontime, тем меньше компаний попадают в неравенство, при этом большая часть компаний имеют высокие показатели вовремя доставленных постингов (попадают в промежуток от 80% до 100%). Это свидетельствует о хорошем качестве сервиса в целом. Чтобы однозначно определиться со значением трешхолда для «полезных» продавцов,

проанализируем детальнее значения распределения процента компаний в зависимости от значений ontime (таб.1):

Таблица 1

Распределение процента компаний в зависимости от значений ontime

Отрезок ontime (отрезки вида (a;b])	Процент компаний, попадающих в промежуток
от 0% до 10%	0.01%
от 10% до 20%	0.12%
от 20% до 30%	0.22%
от 30% до 40%	0.69%
от 40% до 50%	2.33%
от 50% до 60%	1.44%
от 60% до 70%	4.0%
от 70% до 80%	8.97%
от 80% до 90%	20.27%
от 90% до 100%	52.66%

Источник: составлено автором

По значениям таблицы можно сделать вывод, что 72,93% компаний имеет хорошие показатели процента доставок, выполненных вовремя. При этом стоит отметить, что официальные трешхолды блокировок по пособию для продавцов Ozon[1] допускают до 20% просрочек. Кроме того, данная метрика не требует очень строгих ограничений, так как увеличение порогового значения до 90% может привести к «потере» продавцов с высокими показателями продаж. Таким образом, оценка трешхолда для «полезного» продавца в 80% является достаточно справедливой и показательной оценкой. Значит, эталонными значениями можно считать все значения признака ontime, строго превышающие 80%.

[1] https://seller-edu.ozon.ru/ - база знаний маркетплейса — как продавать на Ozon

Следующим шагом проведем исследование показателей рейтинга продавцов, для чего построим графики зависимости числа продавцов от значения рейтинга (рис. 7)

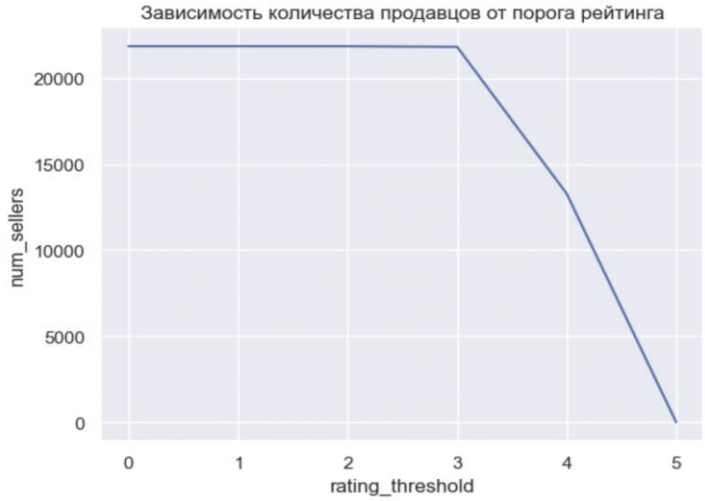

Рис.7 График зависимости числа продавцов от значения рейтинга

Источник: составлено автором

По графику можно наблюдать несколько «переломных» значений, выше которых число компаний резко снижается: это значения 3 и 4. При этом из сводной статистики следует, что среднее значение рейтинга оценивается в 4,26 баллов. Кроме того, 61,26% компаний имеют рейтинг 4 балла и выше. С учетом того, что рейтинг не является одним определяющих факторов полезности, завышение этого показателя может привести к ухудшению оценок. Однако, с другой стороны, низкие пороговые значения могут снизить качество определения полезности продавца. Таким образом, справедливым трешхолдом в оценке рейтинга продавца является трешхолд в 4 балла.

Перейдем к исследованию следующей метрики и проанализируем показатели доли возвратов. Для этого построим графики зависимости числа продавцов от значения доли возвратов (рис. 8)

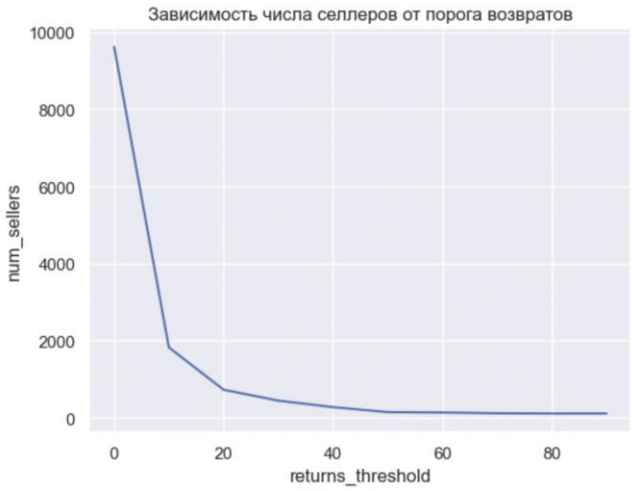

Рис.8 График зависимости числа продавцов от значения доли возвратов

Источник: составлено автором

Из графика следует, что с увеличением значения доли возвратов количество компаний, попадающих в ограничения, уменьшается. То есть можно сделать вывод о низком проценте возвратов на площадке и достаточно высоком качестве услуг продавцов. Чтобы однозначно определить численное пороговое значение метрики, проанализируем распределения (таб. 2):

Таблица 2.

Распределение процента компаний в зависимости от значений доли возвратов

Отрезок доли возвратов (отрезки вида (a;b])	Процент компаний,попадающих в промежуток
от 0% до 10%	91.5%
от 10% до 20%	5.05%
от 20% до 30%	1.41%
от 30% до 40%	0.83%
от 40% до 50%	0.56%
от 50% до 60%	0.05%
от 60% до 70%	0.09%
от 70% до 80%	0.03%
от 80% до 90%	0.01%
от 90% до 100%	0.49%

Источник: составлено автором

Исходя из полученных значений можно сделать вывод, что значительная часть компаний имеют долю возвратов меньше 20%. Так как этот процент по сравнению с пороговым значением в 10%, будет отсекать меньшее количество продавцов с высокими показателям GMV, оценка порога в 20% достаточно справедлива. Кроме того, исходя из текущего фокуса маркетплейсов в пользу значений выручки, показатели качества будут иметь в формуле меньший вес по сравнению с GMV. То есть можно сделать вывод, что для площадки в целом допустимо значение выбранного процента в качестве трешхолда доли возвратов.

Проведем исследование заключительной метрики полезности продавца – количество товаров с лучшим прайс индексом. Для этого построим график зависимости числа продавцов от значения количества товаров с лучшим прайс индексом (рис 9)

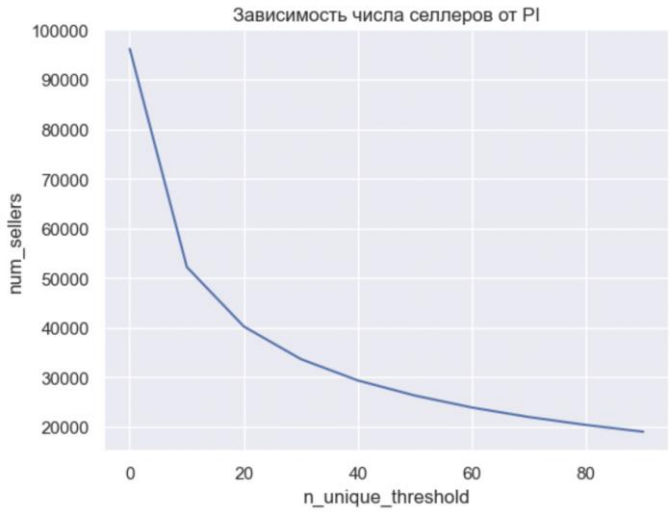

Рис.9 График зависимости числа продавцов от значения количества товаров с лучшим прайс индексом

Источник: составлено автором

По графику можно сделать вывод, что значение числа компаний с увеличением количества товаров с лучшим PI «плавно» уменьшается. При этом из сводной статистики следует, что медиана метрики составляет 4 товара. Так как не все продавцы имеют товары с лучшим PI, оценка данного показателя не должна иметь строгие ограничения. То есть этот признак должен добавлять баллы тем продавцам, которые имеют целевые товары, но при этом

не должен «наказывать» продавцов, не имеющих их в наличии. То есть можно сделать вывод, что справедливым трешхолдом для определения полезности продавца можно считать значение, выше которого лежит половина компаний, то есть ограничение наличия четырех и более товаров с лучшим PI. При этом при построении модели данное ограничение не должно быть строгим для продавца, а должно лишь добавлять баллы к оценке.

Таким образом, можно сделать вывод, что в бинарном распределении значения таргета «полезным» продавцом для площадки будет считаться продавец с наилучшими показателями GMV (выше 80-ого перцентиля), наибольшим процентом вовремя доставленных постингов (строго большим 80%), наилучшим рейтингом (от 4-ёх баллов), наименьшей долей возвратов (20% и ниже) и продавец с высоким количеством товаров с низкими показателями прайс индекса (4 товара и больше).

2.2 Построение функции совокупной полезности продавца как сочетания составляющих метрик

Для создания целевой функции полезности продавца необходимо определить веса входящий в функцию метрик. Прежде всего значения весов должны отражать текущие цели бизнеса. Согласно исследованию рынка маркетплейсов в 2022 году [39] в связи с быстрыми темпами роста сегмента электронной коммерции в первую очередь целью маркетплейсов в России является увеличение капитала, так как рост прибыли позволяет маркетплейсам инвестировать в развитие своих сервисов и технологий, что может привести к улучшению опыта покупателей и продавцов на платформе. Кроме того, увеличение прибыли позволяет маркетплейсам привлекать больше продавцов на платформу, расширять ассортимент товаров и инвестировать в маркетинг и рекламу, что в свою очередь может привести к увеличению узнаваемости бренда. Таким образом, для маркетплейсов в России капитал — это важный фактор улучшения сервиса, привлечения продавцов и клиентов на платформу, увеличения объема продаж и доли рынка. То есть в формуле полезности признак GMV должен иметь больший вес по сравнению с другими показателями.

После GMV приоритет важности должны получить метрики качества: доля вовремя доставленных постингов и доля возвратов, так как это помогает улучшать удовлетворенность покупателей, усиливать преимущества перед конкурентами и увеличивать лояльность клиентов. Покупатели и продавцы могут быть больше склонны использовать маркетплейс, если они довольны качеством предоставляемых услуг, что в последствии может привести к повышению дохода маркетплейса и увеличению его доли на рынке.

В последнюю очередь приоритет должен отдаваться метрикам успешности привлечения клиентов и удержания покупателей: рейтингу, количеству товаров с лучшим PI и доле возвратов, так как они помогают экономить ресурсы и снижать издержки и могут служить индикатором надежности продавца.

Для оценки биномиальной метрики и создания целевой переменной удобно использовать логистическую регрессию. Логистическая регрессия — это метод статистического моделирования, который позволяет предсказывать вероятность бинарного исхода на основе набора факторов или предикторов.

Также этот метод может быть использован для определения наиболее эффективных показателей, которые влияют на оценку полезности продавца. Стоит отметить, что использование логистической регрессии распространено в схожих расчетах, например, в скоринге для определения кредитоспособности заемщиков. В связи с этим, несмотря на ее происхождение из статистики, логистическую регрессию и можно часто встретить в наборе алгоритмов data mining, применяемых в аналогичных расчетах.

Для построения регрессии создадим в датафрейме дополнительный столбец usefulness, имеющий бинарные значения 0 и 1, где 1 обозначает признак «полезного» продавца. Этот столбец будет являться таргетом модели, то есть значением Y. Для тестирования работы модели регрессии заполним значение таргета с использованием следующих неравенств: (рис.10)

```
table['usefulness'] = 0

table.loc[(
          (table['avg_GMV'] >= table['avg_GMV'].quantile(0.8))
          &(table['returns']<=20)
          &(table['rating']>=4)
          &(table['ontime']>80))
          |(
           (table['avg_GMV'] > 0)
           &(table['n_unique']>=4)
           &(table['returns']<=20)
           &(table['ontime']>80)), 'usefulness'] = 1
table
```

Рис.10 Совокупность ограничений для тестирования регрессии

Источник: составлено автором

Выбор неравенств обусловлен следующей логикой: значение товаров с лучшим PI не является обязательным ограничением полезности, это значение только присваивает дополнительные баллы. Следовательно, его нельзя

использовать со всей совокупностью обязательных условий, а значит, допустимо дополнительно присвоить значение полезности 1 тем продавцам, у которых хорошие показатели качества и есть большое количество товаров с лучшим PI, при условии, что GMV может находиться ниже значения трешхолда (однако при этом оно должно быть строго больше нуля, что свидетельствует о наличии продаж в течение последнего месяца).

Дополнительно к массиву X применим метод StandardScaler(), чтобы привести значения к одной шкале измерения и повысить точность выходного результата. Затем разделим данные на тестовые и тренировочные в соотношении 70:30 и объявим модель LogisticRegression(), для которой дополнительно присвоим class_weight=«balanced» из-за неравномерности данных. С использованием возможностей библиотеки GridSearchCV из пакета sklearn.model_selection найдем лучший solver = «newton-cg». Эту модель обучим с помощью методов fit и predict и построим ROC-кривую (рис.11)

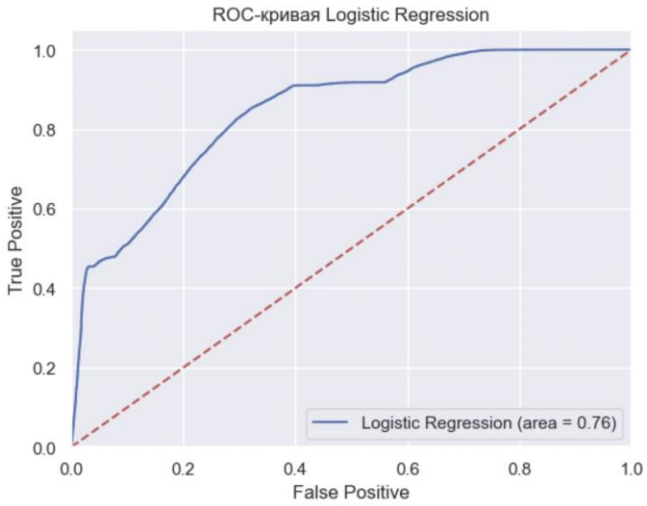

Рис.11 ROC-кривая логистической регрессии

Источник: составлено автором

ROC-кривая используется в логистической регрессии для оценки качества классификатора, который предсказывает вероятность отнесения объекта к определенному классу. ROC-кривая показывает зависимость между долей верно классифицированных объектов и долей ложноположительных результатов при изменении порога классификации, то есть использование ROC-кривая в логистической регрессии позволяет оценить качество

классификатора, выбрать оптимальный порог классификации и настроить модель для достижения наилучшей производительности. Как видно по результату построения ROC-кривой, площадь под ней достаточно большая, а значит, модель имеет высокую точность, равную 0.76, то есть логистическая регрессия может использоваться для дальнейшего исследования. Однако, так как нас больше интересует не классификация объектов на основе вероятности отнесения их к определенному классу, а оценка вероятности бинарного события, попробуем использовать probit -модели (пробит-модель) и logit - модели (логит-модели).

Logit-модель и probit-модель имеют схожие математические основы и обе могут использоваться для оценки вероятности бинарных исходов. Однако, probit-модель используется, если данные имеют нормальное распределение или если предполагается, что они имеют нормальное распределение. В таких случаях probit-модель может дать более точные оценки вероятностей исходов.

Построим распределения трех случайных метрик (рис. 12)

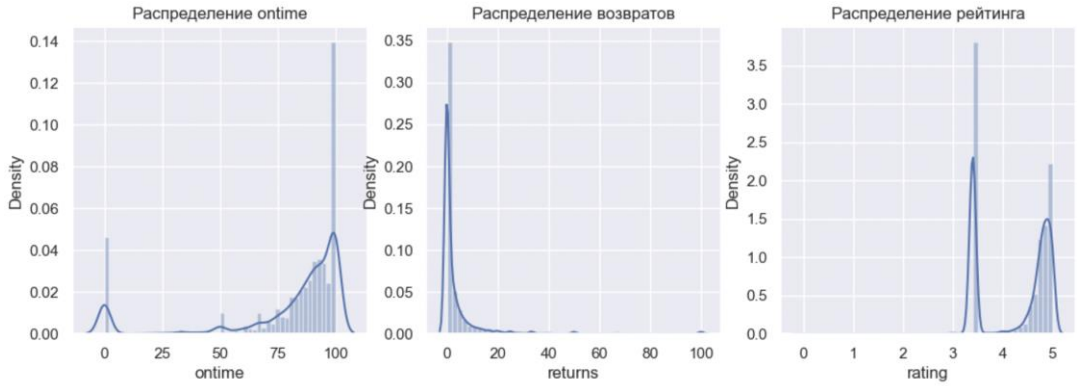

Рис.12 Графики распределений метрик ontime, доли возвратов и значений рейтинга

Источник: составлено автором

По графикам можно сделать вывод, что ontime, доля возвратов и значения рейтинга не имеют нормально распределения, громе того исходя из исследования значений переменных в пункте «1.2 Построение распределения метрик и проведение «разведочного анализа» данных каждого показателя» можно сделать вывод, что значения GMV и количества уникальных товаров не будут распределены нормально, а будут смещены в левую сторону. То есть

в связи с тем, что данные в датасете не распределены нормально logit-модель более точна и удобна для использования по сравнению с probit-моделью.

Так как для полезных продавцов с большим количеством товаром с лучшим прайс индексом трудно аналитически однозначно подобрать ограничение значения GMV, протестируем различные модели и сравним их метрики точности.

Для первой модели попробуем ограничить значение GMV 25-ым и 50-ым перцентилями, полученными из сводной статистики (рис. 4), и с помощью метода mutual_info_classif оценим вклад параметров в каждую модель (рис.13)

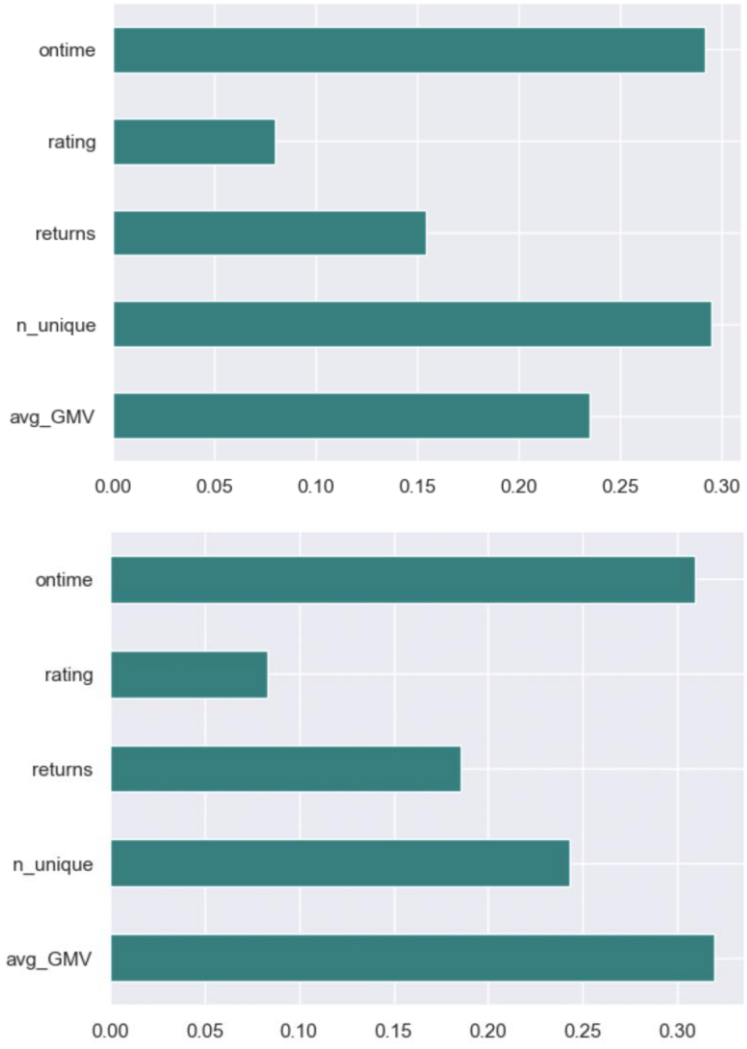

Рис.13 Вклад признаков в модели с ограничением по GMV в 25-ый и 50-ый перцентили соответственно

Источник: составлено автором

Проведем исследование влияния признаков на модель. Для модели с ограничением на 25-ый перцентиль наибольший вклад в модель вносит количество товаров с лучшим прайс индексом, затем по уменьшению влияние следующих факторов: ontime, GMV, доля возвратов и рейтинг. Однако, как было определено ранее в процессе исследования, основной вес должны иметь признаки GMV и ontime, в связи с чем модель с ограничением на 50-ый перцентиль является более подходящей с учетом текущих целей бизнеса маркетплейсов.

Простроим с выбранными ограничениями logit-модель, импортированную из библиотеки statsmodels.api с добавлением свободного члена(const), и проанализируем полученные результаты (рис.14):

```
Optimization terminated successfully.
         Current function value: 0.477781
         Iterations 8
                           Logit Regression Results
==============================================================================
Dep. Variable:              usefulness   No. Observations:               137086
Model:                           Logit   Df Residuals:                   137080
Method:                            MLE   Df Model:                            5
Date:                 Sun, 30 Apr 2023   Pseudo R-squ.:                  0.2334
Time:                         23:31:58   Log-Likelihood:                -65497.
converged:                       True   LL-Null:                       -85444.
Covariance Type:             nonrobust   LLR p-value:                     0.000
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const         -8.9155      0.079   -112.839      0.000      -9.070      -8.761
avg_GMV     2.457e-05   3.12e-07     78.797      0.000     2.4e-05    2.52e-05
n_unique       0.0001   8.56e-06     13.679      0.000       0.000       0.000
returns       -0.0173      0.001    -18.732      0.000      -0.019      -0.016
rating         0.8431      0.011     79.016      0.000       0.822       0.864
ontime         0.0469      0.001     73.882      0.000       0.046       0.048
==============================================================================
```

Рис.14 Сводка логит-модели

По полученной сводке можно сделать вывод, что все коэффициенты регрессии являются статистически значимыми, и кроме того, параметр Df Model показывает, что модель имеет пять степеней свободы: GMV, ontime, товары с PI, доля возвратов и значение рейтинга.

Дополнительно попробуем привести данные к одной шкале измерения с помощью методов StandardScaler() и MinMaxScaler(), но после построения моделей замечаем, что значение Pseudo R-squ не увеличилось, и осталось равным 0.2334 для обеих моделей с проведённой нормализацией, то есть

нормализация не улучшает значительно качество, так как логит-модель в связи с использованием логистической функции для моделирования вероятности бинарного события хорошо обрабатывает величины из разных шкал.

Дополнительно оценим метрики accuracy, precision, recall, f1 score и запишем их в отдельную таблицу для сравнения качества разных моделей.

Для построения следующей модели проанализируем выбросы значений переменной avg_GMV (рис.15)

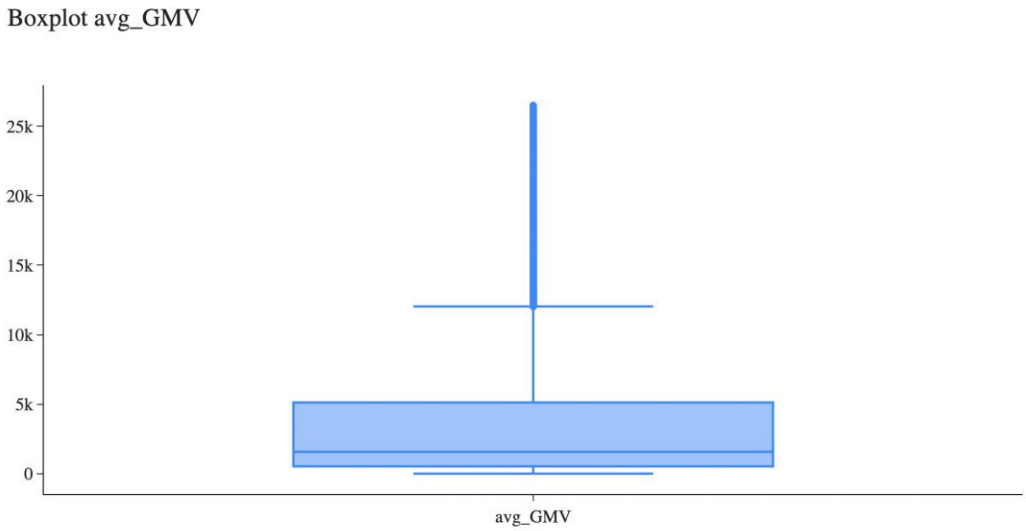

Рис.15 Диаграмма размаха значений avg_GMV

Источник: составлено автором

По диаграмме с усами видно, что столбец средних значений GMV имеет много выбросов. Попробуем исключить их из датасета. Для обнаружения выбросов часто используют межструйный диапазон (IQR), так как согласно работе «Основы статистического анализа данных» [21] величина IQR является очень «устойчивой величиной». IQR по факту обозначает разницу между первым квартилем (Q1 = 25%) и третьим квартилем (Q3 = 75%). Сформируем новую таблицу data, где исключим в столбце avg_GMV значения, лежащие ниже порога Q1 - 1.5 * IQR

и выше порога Q3 + 1.5 * IQR. В таблице data 50-ый перцентиль GMV стал равен 1566 рублям, а 80-ый перцентиль - 6831,807 рублям.

Построим две модели: в первой заменим 80-ый перцентиль значения ограничения GMV из таблицы table на 80-ый перцентиль значения

GMV из таблицы data. Во второй модели дополнительно заменим 50-ый перцениль значения ограничения GMV из таблицы table на 50-ый перцентиль значения GMV из таблицы data. В обеих моделях измерим изменения показателей Pseudo R-squ с использованием метода StandardScaler() и запишем во вспомогательные таблицы значения метрик точности и показателей переменных.

Для последней модели проверим следующую гипотезу: «модель будет показывать лучшие значения точности, если уравнять количество положительных и отрицательных значений весов». Для проверки введем дополнительное значение out_of_ontime, равное разнице между 100% и процентом переменной ontime, которое будет отражать процент постингов, не доставленных вовремя. Для переменной avg_GMV будем использовать такие же показатели, как и в третьей модели, так как она в ходе исследований показала наилучшие результаты (дополнительно пробуем разные ограничения перцентилей GMV из всех датафреймов, однако в ходе исследования убеждаемся, что исходное ограничение показало наилучшие результаты). Аналогично предыдущим шагам запишем метрики качества во вспомогательный датафрейм для сравнения качества моделей (рис.16)

	name	Pseudo R-squ	accuracy	precision	recall	f1 score
0	1 модель	0.2334	0.756310	0.826491	0.282459	0.421028
1	2 модель	0.2401	0.756310	0.826491	0.282459	0.421028
2	3 модель	0.2363	0.746413	0.824232	0.286881	0.425621
3	4 модель	0.2344	0.758650	0.838711	0.295879	0.436348

Рис.16 Сравнение качества моделей

Источник: составлено автором

По сравнительному анализу видим, что метрики accuracy, precision, recall, f1 score в четвертой модели показывают чуть более лучшие результаты по сравнению с другими моделями, однако метрика Pseudo R-squ одна из самых низких. Так как разница по сравнению с другими моделями у accuracy, precision, recall, f1 score незначительна, вторая модель с наилучшим

Pseudo R-squ является наиболее качественной. Дополнительно сравниваем показатели весов и убеждаемся, что разница весов также не изменяется критически, то есть безлайн модели был выбран правильно.

По полученным коэффициентам и формуле вероятности логистической регрессии (1):

$$P = \frac{1}{1+e^{-y}}, = \frac{1}{1+e^{-(b_0+b_1*x_1+b_2*x_2+\cdots+b_n*x_n)}} \quad (1)$$

строим дополнительный столбец prob, который будет отражать вероятность определения продавца как «полезного» и выражаться в шкале от 0 до 1. Это значение будет являться метрикой полезности, то есть, чем выше значение коэффициента prob и ближе к 1, тем более полезен для площадки продавец. И наоборот, чем ближе значение к нулю, тем продавец является менее полезным.

Выводы к главе 2:

1) Разведочный анализ данных позволяет предобработать данные: очистить их от аномальных выбросов и отклонений, выявить основные характеристики датасета, определить вид данных и наиболее значимых переменных, построить допущения для статистических выводов, провести исследование причин и следствий наблюдаемых явлений;

2) Исследуемое решение о полезности продавца масштабируемо, выгодно для бизнеса, а система рекомендаций бесплатна для продавцов;

3) ABC-анализ GMV позволяет выявить наиболее важных продавцов и сконцентрировать на них внимание и ресурсы, чтобы максимизировать выручку и прибыль;

4) Значения весов в модели должны отражать текущие цели бизнеса и могут изменяться в связи со смещением фокуса площадки;

5) В бинарном распределении значения таргета «полезным» продавцом для площадки будет считаться продавец с наилучшими показателями GMV (выше 80-ого перцентиля), наибольшим процентом вовремя доставленных постингов (строго большим 80%), наилучшим

рейтингом (от 4-ёх баллов), наименьшей долей возвратов (20% и ниже) и продавец с высоким количеством товаров с низкими показателями прайс индекса (4 товара и больше);

6) Для оценки биномиальной метрики и создания целевой переменной удобно использовать логистическую регрессию. Также этот метод может быть использован для определения наиболее эффективных показателей, которые влияют на оценку полезности продавца;

7) Индекс цен не является обязательным ограничением полезности. Значение только присваивает дополнительные баллы.

8) В связи с ненормальностью распределения реальных данных для оценки полезности стоит использовать logit-модель, формула вероятности которой отражает приближенность таргета к единице;

Глава 3. Анализ применения полученных результатов и внедрение их в работу маркетплейса

3.1 Принцип разделения продавцов на сегменты в зависимости от значений метрики полезности

В пункте «1.3 Основные признаки и критерии сегментирования продавцов» на основе научных работ Д.Г. Свечкаревой [31], Кирилловой Л.К.[19], Ж.Р. Габбасова [13] был составлен алгоритм разделения пользователей на группы. Подытожим результаты этапов сегментации, проанализированных ранее в работе:

1) Постановка цели. Основными целями сегментации продавцов являются: увеличение прибыли маркетплейса, уменьшение издержек и повышение качества обслуживания за счет внедрения для каждого сегмента совокупности мер и реализации рекомендательной системы. При этом стоит отметить, что приоритет целей также оказывает влияние на веса метрик в модели: например, в случае, когда для площадки важнее увеличение прибыли, больший вес будет иметь метрика GMV; если маркетплейсу приоритетнее повышение качества сервиса даже при условии небольших потерь в выручке – больший вес будет у метрик доли возвратов и доли вовремя доставленных постингов.

2) Выбор критериев. В пункте «2.1 Построение распределения метрик и проведение «разведочного анализа» данных каждого показателя» были проанализированы и подобраны справедливые трешхолды для каждой метрики, а также исследованы возможные модели с разной совокупностью признаков. Подобранная в ходе исследования модель с лучшими показателями качества позволила бинарно разделить продавцов на эталонных и не эталонных.

3) Определение метода сегментации. В зависимости от фокуса групп и приоритетности признаков, определяющих полезность, можно реализовать разные подходы к разделению продавцов. Сегментирование - процесс разделения объекта или явления на более мелкие части с целью детального

анализа и понимания его характеристик. Согласно работе М.М. Борбоедова [9] в качестве признаков, разделяющих продавцов на группы, целесообразно использовать следующие: концептуальные различия между сегментами, целевые размеры сегментов, измеримость и достижимость. Таким образом, информативным и целевым критерием сегментирования для текущего исследования может являться полученное значение признака полезности продавца.

4) Описание каждого сегмента, позволяющее определить особенности целевой аудитории и выбрать стратегию, учитывающую свойства групп. Концептуально для достижения заявленных целей можно разделить пользователей на три группы:

- продавцы, на взаимодействие с которыми следует уменьшать бюджет и принять строгие меры наказания, так как их полезность для маркетплейса очень низкая. Это означает, что затраты на модерацию карточек продавцов, на поддержку менеджеров в чатах, на хранение каточек товаров на серверах площадки выше, чем получаемая прибыль. Кроме того, этот сегмент отличают низкие показатели качества, что может негативно сказываться на репутации маркетплейса в целом;
- продавцы, имеющие средние показатели полезности, для которых можно разработать масштабируемую систему рекомендаций по улучшению взаимодействия с карточкой товара и повышению показателей качества. Кроме того, этот сегмент может являться целевым для продажи селлерам услуг продвижения, рекламы товаров, рассылок и промо-акций, что потенциально увеличит чистую прибыль площадки;
- продавцы, которых необходимо удерживать на площадке, так как они имеют наивысшие показатели полезности. Этот сегмент является целевым для маркетплейса, так как в него входят самые крупные продавцы, приносящие наибольший доход, имеющие лучшие показатели качества и привлекающие большую долю покупателей. Площадке стоит улучшить

взаимодействие с продавцами этого сегмента для удержания их на маркетплейсе;

Таким образом, каждый сегмент имеет объединяющий признак, а кроме того, является достижимым и измеримым.

Определим целевые размеры сегментов. Группы продавцов с наименьшим и наибольшим значением признака полезности нем должны быть многочисленными по следующим причинам:

- комплекс мер, направленный на сегмент продавцов с низким значением полезности, может иметь негативные последствия для репутации площадки и отрицательно повлиять на дальнейшее привлечение продавцов. Однако при этом он не должен быть слишком маленьким, так как для этого сегмента успешность может быть достигнута только за счет масштаба. Таким образом, целевой размер группы с точки зрения бизнес-процессов и особенностей работы не должен превышать 5-10%;
- для целевого сегмента продавцов, имеющего высокие показатели полезности, маркетплейс должен принимать комплекс мер, позволяющих удерживать селлеров на площадке. Это требует дополнительного бюджета, кроме того, для этой группы действует условное правило: «сегмент является эталонным, а значит, нет необходимости развивать его дальше в силу возможного негативного отношения продавцов к пушам и уведомлениям». Таким образом, исходя из текущей ситуации бизнеса, сегмент не должен превышать 5%;

Стоит отметить, что пороговые значения разделения сегментов могут колебаться в связи с изменением фокуса и целей площадки, по причине корректировки допущений бюджета, а также из-за вариативности распределений при ежемесячном пересчете коэффициента полезности. Определим пороговые значения сегментации точнее, для чего построим распределение процента компаний, попадающих в ограничения (таб.3)

Таблица 3.

Распределение процента компаний в зависимости от значений вероятности полезности

Значения вероятности полезности (отрезки вида (a;b])	Процент компаний, попадающих в промежуток
от 0% до 10%	12.34%
от 10% до 20%	14.09%
от 20% до 30%	15.72%
от 30% до 40%	11.4%
от 40% до 50%	12.6%
от 50% до 60%	15.46%
от 60% до 70%	10.1%
от 70% до 80%	2.65%
от 80% до 90%	1.65%
от 90% до 100%	4.01%

Источник: составлено автором

Исходя из полученных значений можно сделать вывод, что для сегмента продавцов с наименьшим значением полезности (prob) необходимо выбрать порог до 10% так, чтобы процент, попадающих в сегмент компаний не превышал 10%, а для группы продавцов с наибольшим значением полезности порог вероятности будет больше 80%, чтобы в группу попадало до 5%. После уточнения подсчетов, выделяем три сегмента:

- от 0% до 5% prob: попадает 8521 компаний - 6.22 %;
- от 6% до 84% prob: попадает 116912 компаний - 85.28 %;
- от 85% prob: попадает 6600 компаний - 4.81 %;

Последним этапом сегментации продавцов является подготовка рекомендаций и комплекса мер со стороны площадки, которые подробнее будут рассмотрены в следующем пункте.

3.2 Построение рекомендаций для работы с разными сегментами со стороны площадки

Исходя из выделенных проблем и потребностей аудитории можно сформировать комплекс мероприятий для достижения поставленных целей.

Для начала сформулируем возможные решения для продавцов с наименьшим значением полезности. Потенциально стоит архивировать карточки продавцов этого сегмента, а затем удалять при несоблюдении условий полезности. Это решение позволит:

- сделать поиск быстрее и наполнить выдачу только актуальными предложениями;
- сэкономить ресурсы на хранение данных на сервере и модерацию карточек;
- улучшить показатели качества за счет блокировки продавцов, нарушающих сроки доставки и предлагающих некачественные продукты и услуги;

Стоит отметить возможный негативный эффект от внедрения выбранного решения: блокировка продавцов может отрицательно сказаться на репутации маркетплейса, увеличить долю оттока с площадки и уменьшить скорость привлечения новых селлеров. Однако, если вводить «мягкое» решение, например, вместо мгновенной блокировки сначала отправлять предупреждения, затем архивировать карточки с временной возможностью восстановления, то сэкономленный бюджет и потенциальное увеличение показателей качества будут превалировать над возможным негативным эффектом. Стоит отметить альтернативное решение: возможно, архивацию карточек стоит рассматривать не только с точки зрения коэффициента полезности продавца, а еще и с точки зрения просмотров и наличия стоков, так как неактуальные карточки могут быть и у успешных продавцов. Однако эффект от этого решения можно оценить только с помощью A/B теста.

Проанализируем возможный комплекс решений для продавцов с наибольшим значением коэффициента полезности. Маркетплейсу важно

удерживать на площадке крупных селлеров, так как по проведенному во второй главе ABC анализу было доказано, что 20% продавцов приносят больше 80% GMV. Таким образом, площадке важно улучшать взаимодействие с продавцами этого сегмента, вводить систему персональных менеджеров и предлагать выгодные «пакетные» предложения.

Для тестирования модели рекомендаций для продавцов со средним значением полезности добавим в датасет дополнительные признаки, которые потенциально могут влиять на приток прибыли и метрики качества:

- num_photos – среднее количество фотографий на карточках товаров;
- has_video – наличие видео на карточках товара (если хотя бы одна карточка имеет видео, значение равно 1, если видео нет ни на одной карточке - 0);
- title_length – средний объем символом в заголовке;
- num_description_words – средний объем слов в описании;
- title_contains_brand - содержание бренда в заголовке объявления (если хотя бы одна карточка в заголовке имеет название бренда, значение равно 1, если видео нет ни на одной карточке – 0);
- promotions - участие в основных акциях (если продавец участвовал хотя бы в одной акции, значение равно 1, если не участвовал ни в одной - 0);

Стоит отметить, что текущей целью работы является построение и исследование возможного решения, которое можно подстроить под цели определенной компании, поэтому искомый набор признаков является не универсальным решением, а безлайном модели. Этот безлайн будет использоваться для тестирования качества и результатов выбранного метода. При масштабировании и реализации модели в рамках реального бизнеса стоит дополнительно использовать ряд признаков, таких, как, например: категория, данные о скорости ответа в чатах, доля не отвеченных сообщений, отклонение цены от рыночной и так далее.

Для обучения модели используем CatBoost. CatBoost это открытая программная библиотека, которая использует уникальный патентованный алгоритм машинного обучения на основе градиентного бустинга. Библиотека была выбрана из-за ряда преимуществ, так как инструмент:

- позволяет получить отличные результаты с параметрами по умолчанию, что сокращает время, необходимое для настройки гиперпараметров;
- обеспечивает повышенную точность за счет уменьшения переобучения;
- имеет возможность быстрого предсказания;
- умеет под капотом обрабатывать пропущенные значения;
- может использоваться для регрессионных и классификационных задач;
- имеет доступную и широкую документацию;

В первую очередь оценим качество выбранной модели и основные метрики precision и recall. Precision (точность) это мера того, насколько точно модель определяет положительный класс (класс, который мы хотим предсказать). Она определяется отношением числа верно определенных положительных примеров к общему числу положительных примеров. Recall (полнота) это мера того, насколько хорошо модель находит все положительные примеры. Она определяется отношением числа верно определенных положительных примеров к общему числу положительных примеров в данных.

В этой модели precision составляет 0.497, recall - 0.952, что является допустимыми значениями. Иногда очень сложно сделать точное предсказание, поскольку часто бывает очень много скрытых от модели внешних факторов, которые трудно заложить в модель. В текущем исследовании важна не столько очень высокая точность предсказания, сколько объяснение, почему модель отнесла продавца к определённому классу, поскольку главная цель

исследования – понять факторы, которые отличают полезных продавцов от неполезных и попытаться повлиять на них.

Для интерпретации модели и анализа результатов используем библиотеку SHAP, которая помогает понять сильные и слабые признаки каждого продавца. Библиотека SHAP позволяет расширить границы применения ML-модели, она не просто ранжирует признаки модели по важности, но и показывает, как именно признаки влияют и насколько сильно смещают таргет. Анализ каждого продавца в отдельности очень полезен, поскольку не очень значимый признак в целом для выборки может для отдельного селлера сильно влиять на его успех, например, если у продавца экстремаль низкое значение признака. В первую очередь оценим вклад каждого признака в модель с помощью метода TreeExplainer(), а затем построим график с помощью shap.summary_plot (рис.17)

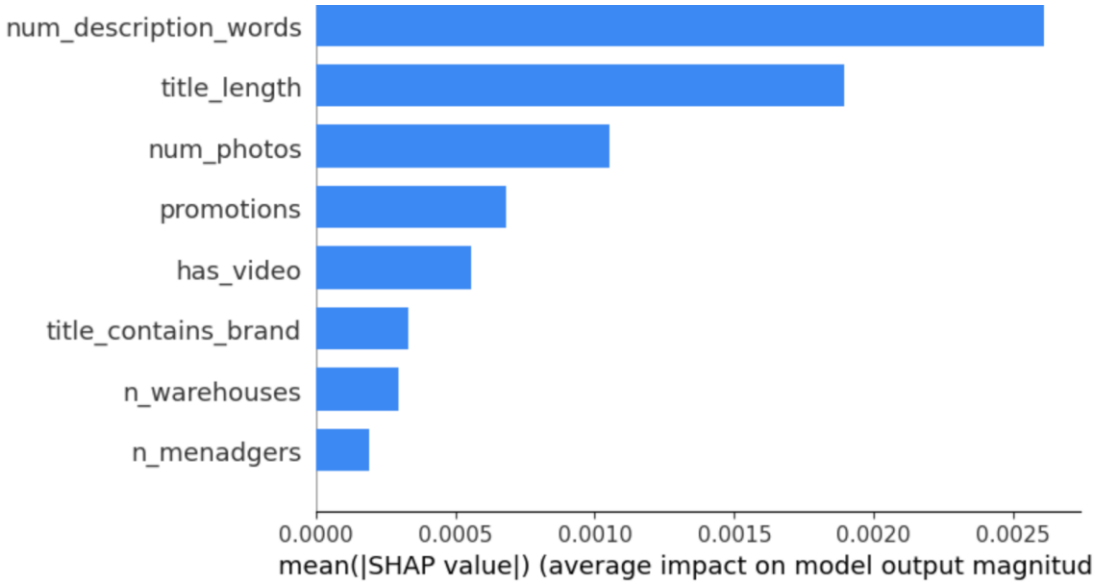

Рис.17 Вклад признаков в модель

По значению графика видно, что наибольшее значение на полезность влияет количество слов в описании. Действительно, чем более подробно и полно описан товар, тем больше доверия он будет вызывать у покупателя, а значит вероятность покупки будет выше. Вторым по приоритетности признаком является длина описания, так как это упрощает систему рекомендаций и по ключевым словам отсылает карточку в выдачу по запросам в поисковой строке. Завершающим топ-3 фактором является количество

фотографий. Этот фактор является менее важным по сравнению с первыми двумя, так как существуют пороговые значения фотографий, до которых обычно при просмотре доходят покупатели и кроме того, товары крупных брендов в силу широкой узнаваемости обычно не имеют большого количества фотографий, что никак оказывает значительного влияния на конверсию из просмотра в покупку. В заключение по убыванию важности расположены следующие признаки: участие в основных акциях, наличие видео на карточке товара, содержание бренда в заголовке, количество складов, количество менеджеров.

Приведем пример использования библиотеки для интерпретации влияния признаков для каждого продавца. Для примера выберем случайный индекс из датасета и построим график (рис.18) (построение графика определяется функцией, поэтому для вывода результатов достаточно передать в функцию соответствующий индекс продавца).

Рис.18 Модель SHAP для случайно выбранного продавца

Источник: составлено автором

Например, выбранный продавец не очень успешен, поскольку у него в среднем всего 20 символов в описании при среднем по датасету в 101 символ, 3 склада, он не имеет видео на карточках, не участвовал в основных акциях и использовал в среднем 6 слов в описании при общем среднем по продавцам в 51 слово. Такому продавцу можно рекомендовать следующее:

- улучшить качество описания: расширить и дополнить его;
- увеличить количество складов, чтобы покрыть большую площадь доставки, улучшить онтайм и скорость доставки;
- добавить видео на карточки товаров;
- принять участие в основных акциях и промо, приобрести услуги продвижения и рекламы;

- увеличить длину заголовка, чтобы повысить вероятность выдачи в поиске;

Стоит отметить, что рекомендации позволяют выдать продавцу набор подсказок и советов, которые потенциально позволили бы ему повысить продажи и улучшить качество сервиса, однако рекомендательная система не предусматривает систему наказаний и штрафов из-за невыполнения предложений площадки.

Потенциальный эффект внедренных изменений будет оценен в следующем пункте главы.

3.3 Анализ потенциальных эффектов от внедрения изменений

Для внедрения в бизнес-процессы метрики полезности продавца, системы рекомендаций и предложенных в работе мер прежде всего необходимо оценить влияние потенциальных изменений и построить ожидаемый прогноз результатов. Для исследования невозможно точно аналитически рассчитать эффект в силу следующих причин:

- переток GMV, из-за которого невозможно ответить на вопрос: «Если пользователь увидит неподходящий продукт менее «полезного» продавца, он сразу перейдет на другую площадку или продолжит искать аналогичный товар на текущем маркетплейсе?». Вполне возможно, что внедренные изменения не столько повысят продажи, сколько повлияют на переток заказов от продавцов с плохим качеством товара к продавцам с хорошими показателями качества, имеющими аналогичные предложения;
- невозможность точной оценки вероятности, с которой продавцы будут реализовывать подсказки площадки. Так как рекомендательная система не предусматривает системы наказаний и штрафов, не все продавцы будут осуществлять предложенный комплекс мер;
- невозможность оценки того, как каждый показатель влияет на GMV в абсолютном выражении;

Точный расчет в таком случае можно получить только с помощью A/B теста. A/B-тест — это методика исследований, которая используется для сравнения двух версий (A и B) продукта. В A/B-тестировании пользователи случайным образом делятся на две группы: контрольную (A), которая не будет получать подсказок и тестовую, которой будут показываться рекомендации. Затем сравнивают результаты двух групп, чтобы определить, какое решение лучше. Цель исследования — повышение качества и рост продаж, таким образом GMV является основной метрикой, а доля вовремя доставленных постингов, доля возвратов, количество товаров с лучшим индексом цен и рейтинг – вспомогательными. Однако в силу невозможности моделирования и проведения A/B на тестовых данных, построим ожидаемый прогноз основной

метрики GMV с помощью регрессионного анализа, оценим влияние каждого из показателей и сравним результаты таргета до и после внедрения рекомендательной системы. При этом важно учитывать, что этот метод не дает такой точной оценки эффекта, как A/B тестирование, и может быть подвержен влиянию таких факторов, как: сезонность, конкуренция, изменения в маркетинговых стратегиях и так далее.

Для расчетов выделим группу продавцов, на которых потенциально может повлиять рекомендательная система. Это продавцы со средним значением метрики полезности. В первую очередь посчитаем веса каждого входящего признака X (рис.19)

	coef	std err	t	P>|t|	[0.025	0.975]
num_photos	43.5308	4.223	10.308	0.000	35.254	51.808
title_length	8.9190	0.848	10.513	0.000	7.256	10.582
num_description_words	19.1868	1.701	11.282	0.000	15.853	22.520
n_warehouses	5369.3671	175.020	30.679	0.000	5026.330	5712.404
n_menadgers	164.2254	104.838	1.566	0.117	-41.256	369.707
has_video	791.6366	102.352	7.734	0.000	591.029	992.244
promotions	782.8914	101.805	7.690	0.000	583.356	982.427
title_contains_brand	470.9666	105.054	4.483	0.000	265.063	676.870

Рис.19 Коэффициенты метрик регрессионной модели

Источник: составлено автором

Из полученных значений следует, что количество менеджеров аккаунта не является значимой переменной, кроме того, для этой метрики ноль лежит в доверительном интервале. Стоит отметить, что на количество менеджеров также трудно влиять, а значит с большой долей вероятности показатель будет изменяться у очень низкого процента продавцов, то есть, ее можно исключить из списка рекомендаций. Определим веса модели после исключения проанализированного признака (рис.20)

	coef	std err	t	P>|t|	[0.025	0.975]
num_photos	43.7096	4.221	10.354	0.000	35.436	51.983
title_length	8.9571	0.848	10.562	0.000	7.295	10.619
num_description_words	19.2610	1.700	11.330	0.000	15.929	22.593
n_warehouses	5528.8354	142.369	38.835	0.000	5249.795	5807.876
has_video	794.5123	102.336	7.764	0.000	593.936	995.089
promotions	786.3234	101.782	7.726	0.000	586.833	985.814
title_contains_brand	472.7879	105.048	4.501	0.000	266.895	678.680

Рис.20 Коэффициенты метрик регрессионной модели после удаления признака «количество менеджеров аккаунта»

Источник: составлено автором

Исходя из полученных значений можно сделать вывод, что все признаки значимы. Свободный член равен нулю, следовательно уравнение таргета Y будет выглядеть следующим образом:

$$Y_1 = 43.709619.2610 * X_{\text{num\_photos}} + 8.9571 * X_{title\_length} + 19.2610 * X_{\text{num\_description\_words}} + 5528.8354 * * X_{\text{n\_warehouses}} + 794.5123 * X_{has\_video} + 786.3234 * X_{\text{promotions}} + 472.7879 * X_{\text{title\_contains\_brand}} \quad (2)$$

При этом Y_1, полученный с помощью формулы, очень близок к реальному Y, отражающему средний GMV: сумма реальных Y составляет 1 115 201 806 рублей, сумма полученных Y_1 – 1 147 421 704 рублей.

Теперь построим таблицу с коэффициентами Шепли библиотеки SHAP (в пункте «3.2 Построение рекомендаций для работы с разными сегментами со стороны площадки» было проанализировано, каким образом библиотека SHAP позволяет понять сильные и слабые признаки каждого продавца). То есть, если коэффициент отрицательный, признак оказывает негативное влияние на модель и смещает таргет влево. В таком случае можно рекомендовать дойти либо до общего среднего среди всех продавцов, либо увеличить признак на N пунктов. Опираясь на соответствующие коэффициенты Шепли, пропишем условие для датасета, получим потенциальные значения X и пересчитаем значение таргета Y_2 по формуле (2) с использованием рассчитанных значений X (рис.21)

Y_target	Y_target_future
25858.4859	26447.629464
10459.4960	10700.395164
16275.7184	21848.263400
8867.7862	10225.216710
20619.7585	26567.635212
...	...
10357.5009	10486.178547
8715.1081	10105.962420
9303.3437	9624.631347
10511.1218	11101.523031
9217.0374	11058.702179

Рис.21 Датасет с текущими и потенциальными значениями таргета

Источник: составлено автором

Исходя из полученных значений таргета для каждого продавца посчитаем возможное процентное изменение GMV, а затем найдем среднее изменение по датасету. Получаем, после внедрения рекомендательной системы метрика GMV в среднем может возрасти на 10,27% (в абсолютных значениях – 3,2 миллиарда рублей ежемесячно). Однако оценка является завышенной, так как ее возможно достичь только при условии, если все продавцы реализуют все предложенные подсказки.

Стоит отметить, что ожидаемые значения были посчитаны для сегмента со средним значением метрики полезности. Кроме того, потенциально возможно получать чистую прибыль за счет продажи акций и услуг продвижения карточек товаров, а также за счет привлечения крупных продавцов в сегменте с высоким значением полезности. При этом такие неучтенные факторы, как, например, сезонность, могут оказывать негативное влияние.

Выводы к главе 3:

1) Основными целями сегментации в исследовании являются: увеличение прибыли маркетплейса, уменьшение издержек и повышение качества обслуживания за счет внедрения для каждого сегмента совокупности мер и реализации рекомендательной системы.

2) Информативный и целевой критерий сегментирования – признак полезности продавца за счет наличия концептуальных различий между группами, измеримости и достижимости.

3) Основные решения для каждого сегмента:

- продавцы с наименьшим значением полезности => строгие меры наказания и архивация карточек;
- продавцы со средними показателями полезности => рекомендательная система и продажа продавцам услуг продвижения, рекламы товаров и промо-акций;

- продавцы с высокими показателями полезности => улучшение взаимодействия;

4) Интерпретировать модель и проанализировать результаты возможно с помощью библиотеки SHAP, которая помогает понять сильные и слабые признаки каждого продавца.

5) Возможный потенциальный рост GMV среднего сегмента составляет 10,27% без учета колебаний, сезонности, акций и маркетинговых стратегий.

ЗАКЛЮЧЕНИЕ

В проведенном исследовании для выпускной квалификационной работы была создана метрика полезности продавцов, основанная на показателях прибыли и качества услуг. Совокупность следующих выбранных показателей: GMV, доля вовремя доставленных постингов, доля возвратов по вине продавца, количество товаров с лучшим индексом цен и рейтинг позволили максимально справедливо сравнить между собой продавцов, имеющих разные характеристики.

Для выбора лучшей модели были построены четыре совокупности признаков, основанных на проанализированных трешхолдах каждой метрики, а также на свойствах и характеристиках входных данных. В бинарном распределении значения таргета «полезным» продавцом для площадки будет считаться продавец с наилучшими показателями GMV (выше 80-ого перцентиля), наибольшим процентом вовремя доставленных постингов (строго большим 80%), наилучшим рейтингом (от 4-ёх баллов), наименьшей долей возвратов (20% и ниже) и продавец с высоким количеством товаров с низкими показателями прайс индекса (4 товара и больше). Однако стоит отметить, что индекс цен не является обязательным ограничением полезности, а лишь присваивает дополнительные баллы.

Для оценки метрики и создания целевой переменной использовалась логистическая регрессия, так как этот метод позволяет определить наиболее эффективные показатели, влияющие на оценку полезности продавца. Сама метрика полезности отражает вероятность получения единицы в бинарном разбиении признака полезности по формуле логит-модели.

В связи с тем, что признак полезности продавца является информативным и целевым критерием сегментирования за счет наличия концептуальных различий между группами, измеримости и достижимости, полученные значения позволили провести сегментацию продавцов, а также разработать рекомендательную систему и совокупность мер, направленных на увеличение прибыли маркетплейса, уменьшение издержек и повышение

качества обслуживания. Например, для продавцов с наименьшим значением полезности были реализованы строгие меры наказания и проведена архивация карточек; для продавцов со средними показателями полезности введена рекомендательная система, а также продвижение карточек и реклама товаров; для продавцов с высокими показателями полезности было предложено улучшить уровень взаимодействия;

Для интерпретации результатов модели использовался функционал библиотеки SHAP, помогающий понять сильные и слабые признаки каждого продавца.

В заключении с помощью регрессионной модели был оценен потенциальный рост GMV от внедрения рекомендательной системы для среднего сегмента продавцов, который в среднем может составить 10,27% без учета колебаний, сезонности, акций и маркетинговых стратегий.

Таким образом, предложенное решение может быть востребовано, актуально и успешно на российском рынке e-commerce.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1) Андреева К.А. Лидогенерация. Маркетинг, который продает. – Санкт-Петербург: Питер, 2014. - 240 с.

2) Андерсон К. Аналитическая культура. От сбора данных до бизнес-результатов / К. Андерсон. – Москва: Манн, Иванов и Фербер, 2017. – 324 с.

3) Андерсон К. Длинный хвост. Эффективная модель бизнеса в Интернете. – Москва: Манн, Иванов и Фербер, 2012. – 304 с.

4) Берг Д.Б., Ульянова Е. А., Добряк П. В. Модели жизненного цикла: учеб. пособие — Екатеринбург: Издательство Уральского университета, 2014. — 74, [2] с.

5) Ковалев А.А. Логистическая регрессия и ROC-анализ — методическое пособие – Москва: 2012 – 103 с.

6) Котлер, Филип Основы маркетинга. Краткий курс / Филип Котлер. – Москва: Вильямс, 2019. – 496 с.

7) Акатьева М.Д. Теоретические аспекты сегментирования — Москва: Международный бухгалтерский учет, 2014, №8 (302), 97 с.

8) Ахраменок, К.А. Построение скоринговой модели с использованием логистической регрессии / К. А. Ахраменок, Н. А. Жиляк // Информационные технологии: Материалы 84-й научно-технической конференции профессорско-преподавательского состава, научных сотрудников и аспирантов (с международным участием), Минск, 03–14 февраля 2020 года / Отв. за издание И.В. Войтов. – Минск: Белорусский государственный технологический университет, 2020. – С. 14-16

9) Борбодоев М.М. Особенность сегментации потребительских рынков — Ош: Актуальные проблемы гуманитарных и естественных наук, 2016, №11-1., 113 с.

10) Васильева А.С., Латыпова А.Э. ABC-анализ - преимущества и недостатки — Ульяновск: Экономика и социум, 2014, №4-2 (13), 173 с.

11) Володько, В.Ф. Сегментирование и позиционирование на рынке / В.Ф. Володько; 2-е изд.– Минск: БНТУ, 2019. – С. 111-113

12) Габалова, Е.Б. Маркетплейс: современный инструмент повышения продаж / Е. Б. Габалова // Modern Science. – 2021. – № 6-2. – С. 35-37.

13) Габбасова, Ж. Р. Анализ проблем метода сегментирования потребителей / Ж. Р. Габбасова // Modern Science. – 2021. – № 3-2. – С. 56-59

14) Гурская С.П. Маркетплейсы – новый сегмент e-commerce — Гомель: Белорусский торгово-экономический университет потребительской кооперации, 2022, С. 26-30

15) Гут, А. В. Современное состояние рынка электронной коммерции в России (на примере маркетплейса «Ozon») / А. В. Гут, А. И. Михайлова // Достижения и перспективы современной науки: материалы Международной (заочной) научно-практической конференции, Нефтекамск, 21 февраля 2022 года. – Нефтекамск: Научно-издательский центр «Мир науки» (ИП Вострецов Александр Ильич), 2022. – С. 66-74.

16) Иванова, Е. К. Маркетплейсы как инструмент развития малого бизнеса — Москва: Инновационная экономика и современный менеджмент, 2021. № 2(33). С. 29-30

17) Изакова, Н. Б. Сегментирование потребителей как ключевой фактор успеха маркетинга взаимоотношений на промышленном рынке / Н. Б. Изакова // Маркетинг и брендинг: вызовы XXI века: Материалы Международной научно-практической конференции, Екатеринбург, 07 ноября 2017 года / Ответственный за выпуск Л. М. Капустина. – Екатеринбург: Уральский государственный экономический университет, 2019. – С. 63-66

18) Карасев А.П. Две стороны понятия «Сегментирование рынка» — Москва: Вестник ГУУ, 2018, №11.

19) Кириллова, Л. К. Сегментация рынка: эволюция и направления развития в условиях цифровизации маркетинга / Л. К. Кириллова // Экономика и предпринимательство. – 2022. – № 1(138), С. 868-871.

20) Козлов, И. Е. Реализация расчета SHAP interaction values для CatBoost / И. Е. Козлов // МНСК-2020: Материалы 58-й Международной научной студенческой конференции, Новосибирск, 2020 –Новосибирский

национальный исследовательский государственный университет, 2020. – С. 29.

21) Конюк А.О. Основы статистического анализа данных –Москва: Актуальные проблемы гуманитарных и естественных наук. 2017. №1-4.

22) Курганова, Н. Ю. Формирование и развитие современных маркетплейсов / Н. Ю. Курганова // Бизнес. Образование. Право. – 2019. – № 4(49). – С. 274-279.

23) Лебедев Б.Д. Рекомендательные системы с применением машинного обучения для интернет-ресурсов — Москва: Национальный исследовательский ядерный университет «МИФИ», 2019, с.265-268

24) Максимова, Н. Б. Исследование рынка маркетплейсов в 2022 году / Н. Б. Максимова // Актуальные вопросы современной экономики. – 2022. – № 7. – С. 408-421.

25) Мешкова, Е. Д. Выход на маркетплейсы как современный тренд в сфере розничной торговли / Е. Д. Мешкова, Е. В. Тинькова // Проблемы развития современного общества : Сборник научных статей 7-й Всероссийской национальной научно-практической конференции, в 5-х томах, Курск, 20–21 января 2022 года. Том 2 Часть 2. – Курск: Юго-Западный государственный университет, 2022. – С. 21-25.

26) Михайлюк М. В. Маркетплейсы как фактор прогрессивной трансформации интернет-торговли в России: логистический аспект — Москва: Экономические науки, 2019, № 172. С. 57-61.

27) Никитин, Н. С. Интерпретирование модели предсказания с помощью SHAP / Н. С. Никитин // Студенческий вестник. – 2022. – № 1-9(193). – С. 67-70.

28) Оболенский, Д. М. Обзор современных методов построения рекомендательных систем - на основе контента и гибридные системы / Д. М. Оболенский, В. И. Шевченко // Мир компьютерных технологий : сборник статей всероссийской научно-технической конференции студентов, аспирантов и молодых ученых,, Севастополь, 05–09 апреля 2021 года /

Министерство науки и высшего образования РФ, Севастопольский государственный университет. – Севастополь: Федеральное государственное автономное образовательное учреждение высшего образования "Севастопольский государственный университет", 2021. – С. 151-156

29) Пекцоркина, И. В. Сравнительный анализ методов ABC И XYZ / И. В. Пекцоркина // Новый путь российской экономики: импортозамещение, инновационность, экономическаяя безопасность: сборник статей Международной научно-практической конференции, Екатеринбург, 23 декабря 2017 года. Том Часть 2. – Екатеринбург: Общество с ограниченной ответственностью "Омега сайнс", 2017. – С. 36-38.

30) Перминов, Н. К. Интерпретация результатов машинного обучения для задачи регрессии / Н. К. Перминов // Информатика: проблемы, методы, технологии: Материалы XXII Международной научно-практической конференции им. Э.К. Алгазинова, Воронеж, 10–12 февраля 2022 года / Под редакцией Д.Н. Борисова. – Воронеж: Общество с ограниченной ответственностью "Вэлборн", 2022. – С. 1185-1196

31) Свечкарева Д. Г. Принципы сегментирования рынка в 21 веке / Д. Г. Свечкарева, Е. О. Краснянская // Modern Science. – 2019. – № 12-3. – С. 105-107.

32) Хоботина Е.А. Российские маркетплейсы: селлеры в 2022 году — Москва: DataInsight», 2022, С. 1-6

33) Чиркин К.Д., Чиркина М.А. Рекомендательные системы — Пенза: ФГБОУ ВО «Пензенский государственный университет архитектуры и строительства», 2020, 134-139

34) Шабаев М.Б., Джангаров А.И. Преимущества и недостатки торговли через маркетплейсы — Чеченя: ФГБОУ ВО «Чеченский государственный университет», 2020, С. 157-160

35) Kizimova, T. A. Using multilomial logistic regression to predict nitrate nitrogen content in soil / T. A. Kizimova // Journal of Agriculture and Environment. – 2022. – No. 7(27).

36) Volik M., Kovaleva M., Btemirova R., Gagloeva I. Methodology of improve-ment of company business processes // Conference: International Conference on Fi-nance, Entrepreneurship and Technologies in Digital Economy. 2021. p. 485-492.

37) Wei J., Lu J., Zhao J. Interactions of competing manufacturers' leader-follower relationship and sales format on online platforms // European Journal of Operational Research, 2020, № 280 (2), P.508-522.

38) Guryanov, A. Efficient Computation of SHAP Values for Piecewise-Linear Decision Trees / A. Guryanov // Proceedings of ITNT 2021 - 7th IEEE International Conference on Information Technology and Nanotechnology:– Samara, 2021

39) The changing world of digital in 2023 [сайт]– 2019. – Текст: электронный. – URL: https://wearesocial.com/us/blog/2023/01/the-changing-world-of-digital-in-2023/ (дата обращения: 17.02.2019).

40) Маркетплейсы заняли почти половину российского рынка электронной торговли [сайт]– Москва, 2019. – Текст: электронный. – URL: https://www.kommersant.ru/doc/5559222 маркетплейсы половина ритейла

```
import pandas as pd
import numpy as np
import random
import matplotlib.pyplot as plt
import math
import seaborn as sns
import statsmodels.api as sm
import sklearn
from copy import deepcopy
from sklearn.model_selection import GridSearchCV
from sklearn.metrics import roc_auc_score
from catboost.utils import get_roc_curve
from sklearn.metrics import roc_curve
from sklearn.metrics import classification_report
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.linear_model import LinearRegression
from sklearn.feature_selection import mutual_info_classif
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_absolute_error, mean_squared_error,
mean_squared_log_error, r2_score
from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import StandardScaler
from sklearn.preprocessing import MinMaxScaler
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score
from catboost import CatBoostClassifier
import xgboost as xgb
import plotly
import plotly.express as px

from sklearn.metrics import precision_score, recall_score
from sklearn.metrics import precision_recall_curve
from sklearn.metrics import precision_recall_curve
from sklearn.metrics import plot_precision_recall_curve
import shap
shap.initjs()
table = pd.read_csv('Desktop/table.csv')
удалим неинформативный столбец
#table.drop(table.columns[[0]], axis = 1, inplace = True)
# посмотрим статистику
table.describe().apply(pd.to_numeric).round(3)
# проверим пустые значения
table.isnull().sum()
plt.figure(figsize=(40, 40))
sns.heatmap(table.corr(), fmt='.1g', annot=True, cmap='coolwarm')
def _color_red_or_green(val):
    color = 'red' if ((val > 0.8)and(val!= 1)) else 'green'
    return 'color: %s' % color

correlation = table.corr()
correlation.style.applymap(_color_red_or_green)
```

```
num_sellers = []
gmv_thresholds = list(range(0, 100000, 1000))
for gmv_threshold in gmv_thresholds:
    num_sellers.append(table[table['avg_GMV']>gmv_threshold].shape[0])

plt.plot(gmv_thresholds, num_sellers)
plt.xlabel('gmv_threshold')
plt.ylabel('num_sellers')
plt.title('Зависимость числа продавцов от порога GMV')
pd.DataFrame(table.nlargest(5, ['avg_GMV'])).apply(pd.to_numeric).round(3)
# распределение GMV
df = pd.DataFrame(columns  = ('quantile', 'GMV_quantile', 'num_companies_higher',
'percent_companies_higher'))
for i in np.arange(0, 1, 0.1):
    quant = table.avg_GMV.quantile(i)
    companies = table[table['avg_GMV']>=quant].n_unique.count()
    df = df.append({'quantile':round(i, 1), 'GMV_quantile':round(quant,
2),'num_companies_higher':companies,
                    'percent_companies_higher':round(companies/table.shape[0]*100, 2)},
ignore_index=True)
# проставим успешность по GMV (по ABC-анализу)
table['usefulness'] = 0
table.loc[table['avg_GMV'] >= table['avg_GMV'].quantile(0.8), 'usefulness'] = 1
table
usefulness_gmv = table[table['usefulness'] == 1].sum().avg_GMV
usefulness_comp = table[table['usefulness'] == 1].count().avg_GMV
uselessess_gmv = table[table['usefulness'] == 0].sum().avg_GMV
uselessess_comp = table[table['usefulness'] == 0].count().avg_GMV
print('Сумма GMV полезных продавцов =', round(usefulness_gmv, 3))
print('Сумма GMV бесполезных продавцов =', round(uselessess_gmv, 3))
print('Процент GMV полезных продавцов от общего GMV =',
round(usefulness_gmv/(usefulness_gmv+uselessess_gmv)*100, 2))
print('Процент GMV полезных продавцов от общего GMV =',
usefulness_comp/(usefulness_comp+uselessess_comp)*100)
sns.set_theme(style="darkgrid")
sns.countplot(table, x='usefulness', hue='usefulness')ё

df = pd.DataFrame(columns  = ('quantile', 'ontime_quantile', 'num_companies_higher',
'percent_companies_higher'))
for i in np.arange(0, 1, 0.1):
    quant = table.ontime.quantile(i)
    companies = table[table['ontime']>=quant].n_unique.count()
    df = df.append({'quantile':round(i, 1), 'ontime_quantile':round(quant,
2),'num_companies_higher':companies,
                    'percent_companies_higher':round(companies/table.shape[0]*100, 2)},
ignore_index=True)
num_sellers = []
ontime_thresholds = list(range(0, 100, 10))
for ontime_threshold in ontime_thresholds:
    num_sellers.append(table[table['ontime']>ontime_threshold].shape[0])

sns.set_theme(style="darkgrid")
```

```
plt.plot(ontime_thresholds, num_sellers)
plt.xlabel('ontime_threshold')
plt.ylabel('num_sellers')
plt.title('Зависимость количества продавцов от порога ontime')
print('от 0% до 10% ontime =',
round(len(table[(table["ontime"]>0)&(table["ontime"]<=10)])/137086*100, 2),
                    '% companies')
print('от 10% до 20% ontime =',
round(len(table[(table["ontime"]>10)&(table["ontime"]<=20)])/137086*100, 2),
                    '% companies')
print('от 20% до 30% ontime =',
round(len(table[(table["ontime"]>20)&(table["ontime"]<=30)])/137086*100, 2),
                    '% companies')
print('от 30% до 40% ontime =',
round(len(table[(table["ontime"]>30)&(table["ontime"]<=40)])/137086*100, 2),
                    '% companies')
print('от 40% до 50% ontime =',
round(len(table[(table["ontime"]>40)&(table["ontime"]<=50)])/137086*100, 2),
                    '% companies')
print('от 50% до 60% ontime =',
round(len(table[(table["ontime"]>50)&(table["ontime"]<=60)])/137086*100, 2),
                    '% companies')
print('от 60% до 70% ontime =',
round(len(table[(table["ontime"]>60)&(table["ontime"]<=70)])/137086*100, 2),
                    '% companies')
print('от 70% до 80% ontime =',
round(len(table[(table["ontime"]>70)&(table["ontime"]<=80)])/137086*100, 2),
                    '% companies')
print('от 80% до 90% ontime =',
round(len(table[(table["ontime"]>80)&(table["ontime"]<=90)])/137086*100, 2),
                    '% companies')
print('от 90% до 100% ontime =',
round(len(table[(table["ontime"]>90)&(table["ontime"]<=100)])/137086*100, 2),
                    '% companies')
plt.title('Зависимость количества продавцов от группах ontime')
table["ontime"].hist(bins = 10)

num_sellers = []
rating_thresholds = list(range(0, 6, 1))
for rating_threshold in rating_thresholds:
   num_sellers.append(table[table['rating']>rating_threshold].shape[0])

sns.set_theme(style="darkgrid")
plt.plot(rating_thresholds, num_sellers)
plt.xlabel('rating_threshold')
plt.ylabel('num_sellers')
plt.title('Зависимость количества продавцов от порога рейтинга')
round(len(table[table['rating']>=4])/len(table)*100, 2)

num_sellers = []
gmv_thresholds = list(range(0, 100, 10))
for gmv_threshold in gmv_thresholds:
```

```
    num_sellers.append(table[table['returns']>gmv_threshold].shape[0])

plt.plot(gmv_thresholds, num_sellers)
plt.xlabel('returns_threshold')
plt.ylabel('num_sellers')
plt.title('Зависимость числа селлеров от порога возвратов')
print('от 0% до 10% =',
round(len(table[(table["returns"]>=0)&(table["returns"]<=10)])/137086*100, 2), '% companies')
print('от 10% до 20% =',
round(len(table[(table["returns"]>10)&(table["returns"]<=20)])/137086*100, 2), '% companies')
print('от 20% до 30% =',
round(len(table[(table["returns"]>20)&(table["returns"]<=30)])/137086*100, 2), '% companies')
print('от 30% до 40% =',
round(len(table[(table["returns"]>30)&(table["returns"]<=40)])/137086*100, 2), '% companies')
print('от 40% до 50% =',
round(len(table[(table["returns"]>40)&(table["returns"]<=50)])/137086*100, 2), '% companies')
print('от 50% до 60% =',
round(len(table[(table["returns"]>50)&(table["returns"]<=60)])/137086*100, 2), '% companies')
print('от 60% до 70% =',
round(len(table[(table["returns"]>60)&(table["returns"]<=70)])/137086*100, 2), '% companies')
print('от 70% до 80% =',
round(len(table[(table["returns"]>70)&(table["returns"]<=80)])/137086*100, 2), '% companies')
print('от 80% до 90% =',
round(len(table[(table["returns"]>80)&(table["returns"]<=90)])/137086*100, 2), '% companies')
print('от 90% до 100% =',
round(len(table[(table["returns"]>90)&(table["returns"]<=100)])/137086*100, 2), '%
companies')

num_sellers = []
n_unique = list(range(0, int(table.n_unique.max()), 5))
for gmv_threshold in gmv_thresholds:
    num_sellers.append(table[table['n_unique']>gmv_threshold].shape[0])

plt.plot(gmv_thresholds, num_sellers)
plt.xlabel('n_unique_threshold')
plt.ylabel('num_sellers')
plt.title('Зависимость числа селлеров от PI')
df = pd.DataFrame(columns  = ('quantile', 'n_unique_quantile', 'num_companies_higher',
'percent_companies_higher'))
for i in np.arange(0, 1, 0.1):
    quant = table.ontime.quantile(i)
    companies = table[table['n_unique']>=quant].n_unique.count()
    df = df.append({'quantile':round(i, 1), 'n_unique_quantile':round(quant,
2),'num_companies_higher':companies,
                    'percent_companies_higher':round(companies/table.shape[0]*100, 2)},
ignore_index=True)
for i in np.arange(0, 10, 1):
    print(i, table[table['n_unique']>=i].n_unique.count(), \
          round(table[table['n_unique']>=i].n_unique.count()/len(table)*100, 2))
print('>=4 ->', round(len(table[table['n_unique']>=4])/len(table)*100, 2), '% companies')
print('< 4 ->', round(len(table[table['n_unique']<4])/len(table)*100, 2), '% companies')
```

```
table['usefulness'] = 0

table.loc[(
        (table['avg_GMV'] >= table['avg_GMV'].quantile(0.8))
        &(table['returns']<=20)
        &(table['rating']>=4)
        &(table['ontime']>80))
        |(
         (table['avg_GMV'] > 0)
         &(table['n_unique']>=4)
         &(table['returns']<=20)
         &(table['ontime']>80)), 'usefulness'] = 1
X = table.drop(['usefulness'],axis=1)
X_scal = StandardScaler().fit_transform(X)
y = table['usefulness']
# разделим данные на тренировочные и тестовые (70% к 30%):
x_train, x_test, y_train, y_test = train_test_split(X_scal, y, test_size = 0.3, random_state=42)

logreg = LogisticRegression()
logreg.fit(x_train, y_train)
model = LogisticRegression(class_weight='balanced')
parameters = {'solver': ['newton-cg', 'lbfgs', 'liblinear', 'sag', 'saga']}

grid_clf = GridSearchCV(model, parameters, cv=5, scoring='accuracy')
grid_clf.fit(x_train, y_train)
print(grid_clf.best_params_)
print(grid_clf.best_score_)
logreg = LogisticRegression(class_weight='balanced', solver='newton-cg')
logreg.fit(x_train, y_train)

logit_roc_auc = roc_auc_score(y_test, logreg.predict(x_test))
fpr, tpr, thresholds = roc_curve(y_test, logreg.predict_proba(x_test)[:,1])
plt.figure()
plt.plot(fpr, tpr, label='Logistic Regression (area = %0.2f)' % logit_roc_auc)
plt.plot([0, 1], [0, 1],'r--')
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('False Positive')
plt.ylabel('True Positive')
plt.title('ROC-кривая Logistic Regression')
plt.legend(loc="lower right")
plt.show()
y_pred = logreg.predict(x_test)
print(classification_report(y_test, y_pred))
# Создаем три графика рядом
fig, axs = plt.subplots(ncols=3, figsize=(13, 4))

# Построение распределения для ontime
sns.distplot(table["ontime"], ax=axs[0])
axs[0].set_title('Распределение ontime')
# Построение распределения для возвратов
sns.distplot(table["returns"], ax=axs[1])
```

```
axs[1].set_title('Распределение возвратов')
# Построение распределения для рейтинга
sns.distplot(table["rating"], ax=axs[2])
axs[2].set_title('Распределение рейтинга')
# Отображение графиков
plt.show()

table['usefulness'] = 0

table.loc[(
        (table['avg_GMV'] >= table['avg_GMV'].quantile(0.8))
        &(table['returns']<=20)
        &(table['rating']>=4)
        &(table['ontime']>80))
        |(
         (table['avg_GMV'] > 689) # 25 перцениль
         &(table['n_unique']>=4)
         &(table['returns']<=20)
         &(table['ontime']>80)), 'usefulness'] = 1
X = table.drop(['usefulness'],axis=1)
y = table['usefulness']
importances = mutual_info_classif(X, y)
feature_importances = pd.Series(importances, table.columns[0:len(table.columns)-1])
feature_importances.plot(kind='barh', color='teal')
plt.show()
table['usefulness'] = 0

table.loc[(
        (table['avg_GMV'] >= table['avg_GMV'].quantile(0.8))
        &(table['returns']<=20)
        &(table['rating']>=4)
        &(table['ontime']>80))
        |(
         (table['avg_GMV'] > 2060.645) #50 перцентиль
         &(table['n_unique']>=4)
         &(table['returns']<=20)
         &(table['ontime']>80)), 'usefulness'] = 1
X = table.drop(['usefulness'],axis=1)
y = table['usefulness']
importances = mutual_info_classif(X, y)
feature_importances = pd.Series(importances, table.columns[0:len(table.columns)-1])
feature_importances.plot(kind='barh', color='teal')
plt.show()
df = pd.DataFrame(columns  = ('name', 'accuracy', 'precision', 'recall', 'f1 score'))
df1 = pd.DataFrame(columns  = ('name', 'Pseudo R-squ', 'avg_GMV', 'n_unique', 'returns',
'rating', 'ontime'))
X = table.drop(['usefulness'],axis=1)
y = table['usefulness']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)

model = sm.Logit(y_train, X_train)
```

```
result = model.fit()
y_pred = result.predict(X_test)

print("Accuracy:", round(accuracy_score(y_test, y_pred.round()), 6))
print("Precision:", round(precision_score(y_test, y_pred.round()), 6))
print("Recall:", round(recall_score(y_test, y_pred.round()), 6))
print("F1 score:", round(f1_score(y_test, y_pred.round()), 6))
df = df.append({'name':'1 модель', 'accuracy':0.75631,'precision':0.826491, 'recall':0.282459, 'f1
score':0.421028}
               ,ignore_index=True)
X = table.drop(['usefulness'],axis=1)
X1 = sm.add_constant(X)
y = table['usefulness']
logit_model=sm.Logit(y, X1)
result=logit_model.fit()
print(result.summary())
X = table.drop(['usefulness'],axis=1)
X_scal = StandardScaler().fit_transform(X)
X1 = sm.add_constant(X_scal)
y = table['usefulness']
logit_model=sm.Logit(y, X1)
result=logit_model.fit()
print(result.summary())
X = table.drop(['usefulness'],axis=1)
X_scal = MinMaxScaler().fit_transform(X)
X1 = sm.add_constant(X_scal)
y = table['usefulness']
logit_model=sm.Logit(y, X1)
result=logit_model.fit()
print(result.summary())
df1 = df1.append({'name':'1 модель', 'Pseudo R-squ':0.2334,'avg_GMV':2.457e-05,
'n_unique':0.0001,
                  'returns':-0.0173, 'rating':0.8431, 'ontime':0.0469}
                 ,ignore_index=True)
table["y1"] = -8.9155+table["avg_GMV"]*2.457e-05+table["n_unique"]*0.0001+\
              table["returns"]*(-0.0173)+table["rating"]*(0.8431)+table["ontime"]*(0.0469)
table['prob'] = 1/(1+np.exp(-table["y1"]))
table[['avg_GMV', 'n_unique', 'returns', 'rating', 'ontime', 'prob']]

table = table[['avg_GMV', 'n_unique', 'returns', 'rating', 'ontime']]
Q1 = table["avg_GMV"].quantile(q=0.25)
Q3= table["avg_GMV"].quantile(q=0.75)
IQR = Q3 - Q1

data = table[table["avg_GMV"] < (Q3+1.5*IQR)]
data.describe()
table = table[['avg_GMV', 'n_unique', 'returns', 'rating', 'ontime']]
table['usefulness'] = 0

table.loc[(
        (table['avg_GMV'] >= gmv_treshold)
        &(table['returns']<=20)
```

```
        &(table['rating']>=4)
        &(table['ontime']>80))
        |(
         (table['avg_GMV'] > 2060.645)
         &(table['n_unique']>=4)
         &(table['returns']<=20)
         &(table['ontime']>80)), 'usefulness'] = 1
X = table.drop(['usefulness'],axis=1)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
y = table['usefulness']
model = sm.Logit(y_train, X_train)
result = model.fit()
y_pred = result.predict(X_test)

print("Accuracy:", round(accuracy_score(y_test, y_pred.round()), 6))
print("Precision:", round(precision_score(y_test, y_pred.round()), 6))
print("Recall:", round(recall_score(y_test, y_pred.round()), 6))
print("F1 score:", round(f1_score(y_test, y_pred.round()), 6))
df = df.append({'name':'2 модель', 'accuracy':0.75631,'precision':0.826491, 'recall':0.282459, 'f1
score':0.421028}
               ,ignore_index=True)
X = table.drop(['usefulness'],axis=1)
X1 = sm.add_constant(X)
y = table['usefulness']
logit_model=sm.Logit(y, X1)
result=logit_model.fit()
print(result.summary())
X = table.drop(['usefulness'],axis=1)
X_scal = StandardScaler().fit_transform(X)
X1 = sm.add_constant(X_scal)
y = table['usefulness']
logit_model=sm.Logit(y, X1)
result=logit_model.fit()
print(result.summary())
df1 = df1.append({'name':'2 модель', 'Pseudo R-squ':0.2401,'avg_GMV':2.408e-05,
'n_unique':0.0001,
                  'returns':-0.0183, 'rating':0.9302, 'ontime':0.0476}
                 ,ignore_index=True)
table["y1"] = -9.2736+table["avg_GMV"]*2.408e-05+table["n_unique"]*0.0001+\
              table["returns"]*(-0.0183)+table["rating"]*(0.9246)+table["ontime"]*(0.0476)
table['prob'] = 1/(1+np.exp(-table["y1"]))
table = table[['avg_GMV', 'n_unique', 'returns', 'rating', 'ontime', 'prob']]
# table.avg_GMV = table.avg_GMV.apply(pd.to_numeric).round(9)
table.avg_GMV = table.avg_GMV.astype('int64')
table.sort_values(by='prob', ascending=False)

import plotly.graph_objects as go
sns.boxplot(y=data['avg_GMV'], color='blue', width=0.7, linewidth = 2, fliersize = 9)
columns = table.columns.tolist()

fig = go.Figure()
fig.add_trace(go.Box(y=data[columns[0]], name=columns[0], marker_color = '#1589FF'))
```

```
fig.update_layout(
   font=dict(size=15,family="Franklin Gothic"),
   template='simple_white',
   title = 'Boxplot avg_GMV')
fig.show()
table = table[['avg_GMV', 'n_unique', 'returns', 'rating', 'ontime']]
Q1 = table["avg_GMV"].quantile(q=0.25)
Q3= table["avg_GMV"].quantile(q=0.75)
IQR = Q3 - Q1

data = table[table["avg_GMV"] < (Q3+1.5*IQR)]
data.describe()
gmv_treshold = data['avg_GMV'].quantile(0.8)
table = table[['avg_GMV', 'n_unique', 'returns', 'rating', 'ontime']]
table['usefulness'] = 0

table.loc[(
        (table['avg_GMV'] >= gmv_treshold)
        &(table['returns']<=20)
        &(table['rating']>=4)
        &(table['ontime']>80))
        |(
         (table['avg_GMV'] > 1566.1616)
         &(table['n_unique']>=4)
         &(table['returns']<=20)
         &(table['ontime']>80)), 'usefulness'] = 1
X = table.drop(['usefulness'],axis=1)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
y = table['usefulness']
model = sm.Logit(y_train, X_train)
result = model.fit()
y_pred = result.predict(X_test)

print("Accuracy:", round(accuracy_score(y_test, y_pred.round()), 6))
print("Precision:", round(precision_score(y_test, y_pred.round()), 6))
print("Recall:", round(recall_score(y_test, y_pred.round()), 6))
print("F1 score:", round(f1_score(y_test, y_pred.round()), 6))
df = df.append({'name':'3 модель', 'accuracy':0.746413,'precision':0.824232, 'recall':0.286881,
'f1 score':0.425621}
              ,ignore_index=True)
X = table.drop(['usefulness'],axis=1)
X1 = sm.add_constant(X)
y = table['usefulness']
logit_model=sm.Logit(y, X1)
result=logit_model.fit()
print(result.summary())
X = table.drop(['usefulness'],axis=1)
X_scal = StandardScaler().fit_transform(X)
X1 = sm.add_constant(X_scal)
y = table['usefulness']
logit_model=sm.Logit(y, X1)
```

```
result=logit_model.fit()
print(result.summary())
df1 = df1.append({'name':'3 модель', 'Pseudo R-squ':0.2363,'avg_GMV':2.228e-5,
'n_unique':0.0001,
                  'returns':-0.0190, 'rating':0.9302, 'ontime':0.0487}, ignore_index=True)
table['out_of_ontime'] = 100 - table['ontime']
table = table[['avg_GMV', 'n_unique', 'returns', 'rating', 'out_of_ontime']]

table['usefulness'] = 0

table.loc[(
        (table['avg_GMV'] >= gmv_treshold)
        &(table['returns']<=20)
        &(table['rating']>=4)
        &(table['out_of_ontime']<=20))
        |(
         (table['avg_GMV'] > 1566.645)
         &(table['n_unique']>=4)
         &(table['returns']<=20)
         &(table['out_of_ontime']<=20)), 'usefulness'] = 1
X = table.drop(['usefulness'],axis=1)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
y = table['usefulness']
model = sm.Logit(y_train, X_train)
result = model.fit()
y_pred = result.predict(X_test)

print("Accuracy:", round(accuracy_score(y_test, y_pred.round()), 6))
print("Precision:", round(precision_score(y_test, y_pred.round()), 6))
print("Recall:", round(recall_score(y_test, y_pred.round()), 6))
print("F1 score:", round(f1_score(y_test, y_pred.round()), 6))
df = df.append({'name':'4 модель', 'accuracy':0.75865,'precision':0.838711, 'recall':0.295879, 'f1
score':0.436348}
               ,ignore_index=True)
X = table.drop(['usefulness'],axis=1)
X1 = sm.add_constant(X)
y = table['usefulness']
logit_model=sm.Logit(y, X1)
result=logit_model.fit()
print(result.summary())
X = table.drop(['usefulness'],axis=1)
X_scal = StandardScaler().fit_transform(X)
X1 = sm.add_constant(X_scal)
y = table['usefulness']
logit_model=sm.Logit(y, X1)
result=logit_model.fit()
print(result.summary())
df.merge(df1, how='left', on='name')[['name', 'Pseudo R-squ', 'accuracy', 'precision', 'recall', 'f1
score']]
table["y1"] = -9.2736+table["avg_GMV"]*2.408e-05+table["n_unique"]*0.0001+\
              table["returns"]*(-0.0183)+table["rating"]*(0.9302)+table["ontime"]*(0.0476)
table['prob'] = 1/(1+np.exp(-table["y1"]))
```

```
table[['avg_GMV', 'n_unique', 'returns', 'rating', 'ontime', 'prob']]
print('от 0% до 10% prob =', len(table[(table["prob"]>=0)&(table["prob"]<=0.1)]), '-',
                    round(len(table[(table["prob"]>=0)&(table["prob"]<=0.1)])/len(table)*100, 2),
'%')
print('от 10% до 20% prob =', len(table[(table["prob"]>0.1)&(table["prob"]<=0.2)]), '-',
                    round(len(table[(table["prob"]>=0.1)&(table["prob"]<=0.2)])/len(table)*100,
2), '%')
print('от 20% до 30% prob =', len(table[(table["prob"]>0.2)&(table["prob"]<=0.3)]), '-',
                    round(len(table[(table["prob"]>=0.2)&(table["prob"]<=0.3)])/len(table)*100,
2), '%')
print('от 30% до 40% prob =', len(table[(table["prob"]>0.3)&(table["prob"]<=0.4)]), '-',
                    round(len(table[(table["prob"]>=0.3)&(table["prob"]<=0.4)])/len(table)*100,
2), '%')
print('от 40% до 50% prob =', len(table[(table["prob"]>0.4)&(table["prob"]<=0.5)]), '-',
                    round(len(table[(table["prob"]>=0.4)&(table["prob"]<=0.5)])/len(table)*100,
2), '%')
print('от 50% до 60% prob =', len(table[(table["prob"]>0.5)&(table["prob"]<=0.6)]), '-',
                    round(len(table[(table["prob"]>=0.5)&(table["prob"]<=0.6)])/len(table)*100,
2), '%')
print('от 60% до 70% prob =', len(table[(table["prob"]>0.6)&(table["prob"]<=0.7)]), '-',
                    round(len(table[(table["prob"]>=0.6)&(table["prob"]<=0.7)])/len(table)*100,
2), '%')
print('от 70% до 80% prob =', len(table[(table["prob"]>0.7)&(table["prob"]<=0.8)]), '-',
                    round(len(table[(table["prob"]>=0.7)&(table["prob"]<=0.8)])/len(table)*100,
2), '%')
print('от 80% до 90% prob =', len(table[(table["prob"]>0.8)&(table["prob"]<=0.9)]), '-',
                    round(len(table[(table["prob"]>=0.8)&(table["prob"]<=0.9)])/len(table)*100,
2), '%')
print('от 90% prob =', len(table[(table["prob"]>0.9)]), '-',
                    round(len(table[table["prob"]>0.9])/len(table)*100, 2), '%')
table = pd.read_csv('Desktop/table.csv')
table = table[['avg_GMV', 'n_unique', 'returns', 'rating', 'ontime']]
table['usefulness'] = 0

table.loc[(
        (table['avg_GMV'] >= gmv_treshold)
        &(table['returns']<=20)
        &(table['rating']>=4)
        &(table['ontime']>80))
        |(
         (table['avg_GMV'] > 2060.645)
         &(table['n_unique']>=4)
         &(table['returns']<=20)
         &(table['ontime']>80)), 'usefulness'] = 1
table["y1"] = -9.2736+table["avg_GMV"]*2.408e-05+table["n_unique"]*0.0001+\
            table["returns"]*(-0.0183)+table["rating"]*(0.9302)+table["ontime"]*(0.0476)
table['prob'] = 1/(1+np.exp(-table["y1"]))
X = table[['num_photos', 'title_length', 'num_description_words', 'n_warehouses', 'n_menadgers',
'has_video', 'promotions', 'title_contains_brand']]
y = table['usefulness']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
```

```
model = CatBoostClassifier(iterations=300,
                           learning_rate=0.1,
                           depth=5, auto_class_weights='Balanced',
                           random_seed=79)
# Fit model
model.fit(X_train[features], y_train)

print(list(sorted(zip(model.get_feature_importance(),model.feature_names_))))
explainer = shap.TreeExplainer(model)
shap_values = explainer.shap_values(X[features])
shap.summary_plot(shap_values, X_test[features], plot_type="bar")
## shap
features = ['num_photos', 'title_length', 'num_description_words', 'n_warehouses', 'n_menadgers',
'has_video', 'promotions', 'title_contains_brand']
X = table[['num_photos', 'title_length', 'num_description_words', 'n_warehouses', 'n_menadgers',
'has_video', 'promotions', 'title_contains_brand']]
y = table['usefulness']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)

model = CatBoostClassifier(iterations=300,
                           learning_rate=0.1,
                           depth=5, auto_class_weights='Balanced',
                           random_seed=79)
# Fit model
model.fit(X_train[features], y_train)
explainer = shap.TreeExplainer(model)
shap_values = explainer.shap_values(X[features])
shap_df = pd.DataFrame(shap_values, columns=X_test.columns)
def shap_plot(j):
    explainerModel = shap.TreeExplainer(model)
    shap_values_Model = explainerModel.shap_values(X[features])
    p = shap.force_plot(explainerModel.expected_value, shap_values_Model[j],
X[features].iloc[[j]])
    return(p)
shap.force_plot(shap.TreeExplainer(model).expected_value, shap_plot(2)[2],
X[features].iloc[[2]])
shap_plot(2)
thr = 0.4
predicts = predicts_proba[:, 1] > thr

print(precision_score(y_test,predicts))
print(recall_score(y_test, predicts))
print('---'*25)
print('Средний GMV неполезных продавцов =', round(table[table['usefulness'] ==
1].avg_GMV.mean(), 2))
print('Средний GMV полезных продавцов =', round(table[table['usefulness'] ==
0].avg_GMV.mean(), 2))
print('---'*25)
print('Средний ontime неполезных продавцов =', round(table[table['usefulness'] ==
1].ontime.mean(), 2))
print('Средний ontime полезных продавцов =', round(table[table['usefulness'] ==
0].ontime.mean(), 2))
```

```
print('---'*25)
print('Среднее количество товаров с лучшим индексом цен у неполезных продавцов =',
round(table[table['usefulness'] == 1].n_unique.mean(), 2))
print('Среднее количество товаров с лучшим индексом цен у полезных продавцов =',
round(table[table['usefulness'] == 0].n_unique.mean(), 2))
print('---'*25)
print('Средняя доли возвратов у неполезных продавцов =', round(table[table['usefulness'] ==
1].returns.mean(), 2))
print('Средняя доли возвратов у полезных продавцов =', round(table[table['usefulness'] ==
0].returns.mean(), 2))
print('---'*25)
print('Средний рейтинг неполезных продавцов =', round(table[table['usefulness'] ==
1].rating.mean(), 2))
print('Средний рейтинг полезных продавцов =', round(table[table['usefulness'] ==
0].rating.mean(), 2))
print('---'*25)
## shap
table = table[(table["prob"]>=0.06)&(table["prob"]<=0.84)]
features = ['num_photos', 'title_length', 'num_description_words', 'n_warehouses', 'n_menadgers',
'has_video', 'promotions', 'title_contains_brand']
X = table[['num_photos', 'title_length', 'num_description_words', 'n_warehouses', 'n_menadgers',
'has_video', 'promotions', 'title_contains_brand']]
y = table['usefulness']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)

model = CatBoostClassifier(iterations=300,
                           learning_rate=0.1,
                           depth=5, auto_class_weights='Balanced',
                           random_seed=79)
# Fit model
model.fit(X_train[features], y_train)

explainer = shap.TreeExplainer(model)
shap_values = explainer.shap_values(X[features])

shap_df = pd.DataFrame(shap_values, columns=X_test.columns)
shap_means = shap_df.abs().mean(axis=0).sort_values(ascending=False)
## пример
X[features].iloc[[2]]
shap_plot(2)[2]
## поиск весов
table1 = table.copy()
table1 = table1.drop(columns = ['n_menadgers'])
X = table1[['num_photos', 'title_length', 'num_description_words', 'n_warehouses', 'has_video',
'promotions', 'title_contains_brand']]
y = table1['avg_GMV']
X = table[['num_photos', 'title_length', 'num_description_words', 'n_warehouses', 'n_menadgers',
'has_video', 'promotions', 'title_contains_brand']]
# Создаем модель линейной регрессии
model = sm.OLS(y, X)
# Обучаем модель
results = model.fit()
```

```
# Выводим summary модели
print(results.summary())
X = table1[['num_photos', 'title_length', 'num_description_words', 'n_warehouses', 'has_video',
'promotions', 'title_contains_brand']]
# Создаем модель линейной регрессии
model = sm.OLS(y, X)
# Обучаем модель
results = model.fit()
# Выводим summary модели
print(results.summary())
x = table[['num_photos', 'promotions', 'has_video', 'title_length', 'num_description_words',
'title_contains_brand', 'n_warehouses']]
y = table[['avg_GMV']]
model = LinearRegression().fit(x, y)
print("Intercept:", model.intercept_)
shap_df = shap_df[['num_photos', 'title_length', 'num_description_words',
'n_warehouses','has_video', 'promotions', 'title_contains_brand']]
shap_df = shap_df.rename(columns={'num_photos':'num_photos_shap',
'title_length':'title_length_shap', 'num_description_words':'num_description_words_shap',
'n_warehouses':'n_warehouses_shap', 'has_video':'has_video_shap',
'promotions':'promotions_shap', 'title_contains_brand':'title_contains_brand_shap'})
shap_df = shap_df.reset_index()
table = table.reset_index()
num_photos_mean = tt1[['num_photos']].mean()
promotions_mean = tt1[['promotions']].mean()
has_video_mean = tt1[['has_video']].mean()
title_length_mean = tt1[['title_length']].mean()
num_description_words_mean = tt1[['num_description_words']].mean()
title_contains_brand_mean = tt1[['title_contains_brand']].mean()
n_warehouses_mean = tt1[['n_warehouses']].mean()
tt1=pd.concat([table, shap_df], axis=1)
def choose_value(row):
    if row['num_photos_shap'] > 0:
        return row['num_photos']
    elif row['num_photos'] > num_photos_mean[0]:
        return row['num_photos']+1
    else: return num_photos_mean[0]

# Применяем функцию к каждой строке
tt1['num_photos_shap1'] = tt1.apply(choose_value, axis=1)
def choose_value(row):
    if row['promotions_shap'] > 0:
        return row['promotions']
    elif row['promotions'] > promotions_mean[0]:
        return 1 #row['promotions']+1
    else: return promotions_mean[0]
# Применяем функцию к каждой строке
tt1['promotions_shap1'] = tt1.apply(choose_value, axis=1)

def choose_value(row):
    if row['title_length_shap'] > 0:
        return row['title_length']
```

```
    elif row['title_length'] > title_length_mean[0]:
        return row['title_length']+10
    else: return title_length_mean[0]
# Применяем функцию к каждой строке
tt1['title_length_shap1'] = tt1.apply(choose_value, axis=1)


def choose_value(row):
    if row['num_description_words_shap'] > 0:
        return row['num_description_words']
    elif row['num_description_words']>num_description_words_mean[0]:
        return row['num_description_words']+10
    elif (row['num_description_words_shap'] <=
0)&(row['num_description_words_shap']<num_description_words_mean[0]):
        return num_description_words_mean[0]
# Применяем функцию к каждой строке
tt1['num_description_word_shap1'] = tt1.apply(choose_value, axis=1)

def choose_value(row):
    if row['title_contains_brand_shap'] > 0:
        return row['title_contains_brand']
    elif row['title_contains_brand'] > title_contains_brand_mean[0]:
        return 1 #row['title_contains_brand']+1
    else: return title_contains_brand_mean[0]
# Применяем функцию к каждой строке
tt1['title_contains_brand_shap1'] = tt1.apply(choose_value, axis=1)

def choose_value(row):
    if row['n_warehouses_shap'] > 0:
        return row['n_warehouses']
    elif row['n_warehouses'] > n_warehouses_mean[0]:
        return row['n_warehouses']+1
    else: return n_warehouses_mean[0]
# Применяем функцию к каждой строке
tt1['n_warehouses_shap1'] = tt1.apply(choose_value, axis=1)
def choose_value(row):
    if row['has_video_shap'] > 0:
        return row['has_video']
    elif row['has_video'] > has_video_mean[0]:
        return 1 #row['has_video']+1
    else: return has_video_mean[0]
# Применяем функцию к каждой строке
tt1['has_video_shap1'] = tt1.apply(choose_value, axis=1)
tt1[['Y_target' , 'Y_target_future']]
tt1["Y_target_future"] =
19.2610*tt1["num_description_words_shap1"]+8.9571*tt1["title_length_shap1"]+43.7096*tt1["
num_photos_shap1"]+786.3234*tt1["promotions_shap1"]+794*tt1["has_video_shap1"]+5528.8
354*tt1["n_warehouses_shap1"]+472.7879*tt1["title_contains_brand_shap1"]
tt1.Y_target_future.sum()
(tt1["Y_target_future"].sum()-tt1["Y_target"].sum())*30
((tt1["Y_target_future"].sum()-tt1["Y_target"].sum())/tt1["Y_target"].sum())*100
tt2['percent'] = ((tt2["Y_target_future"]-tt2["Y_target"])/tt1["Y_target"])*100
```