

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

Факультет информатики, математики и компьютерных наук

**Программа подготовки бакалавров по направлению
38.03.05 Бизнес-информатика**

Кононова Екатерина Сергеевна

КУРСОВАЯ РАБОТА

**Многокритериальная система автоматической корректировки наценок
при онлайн торговле**

Научный руководитель:

канд. техн. наук, доцент

Крылов Сергей Владимирович

Нижний Новгород, 2020

Оглавление

Введение.....	3
Глава 1. Понятия наценки и Интернет-торговли.	6
1.1. Цена и ее основные характеристики.	6
1.2. Методы расчета наценок и скидок.	9
1.3. Особенности онлайн продаж и управление ценообразованием.	12
1.4. Анализ онлайн рынка в России и в мире.	18
Глава 2. Разработка системы автоматической корректировки наценок.....	21
2.1. Описание текущей ситуации и существующих алгоритмов.	21
2.2. Выявление и анализ требований к будущей системе.....	24
2.3. Реализация системы наценок.	33
2.4. Тестирование системы и сравнительный анализ.....	41
Заключение	46
Список литературы	49
Приложения.	
Приложение 1. Диаграмма классов.	
Приложение 2. Диаграмма активности.....	
Приложение 3. Диаграмма последовательности.....	
Приложение 4.Диаграмма состояний системы.	
Приложение 5. Многокритериальная система автоматизированной корректировки наценки.	

Введение

Актуальность темы исследования обусловлена активным развитием онлайн торговли и необходимостью изменения наценок в зависимости от различных факторов, влияющих на уровень продаж. В современном мире всё больше и больше людей пользуются услугами интернет магазинов и приобретают вещи, не выходя из дома. Спрос порождает предложение, а как следствие и необходимость контроля наценок на товары во избежание работы предприятия в убыток. Во многом эффективность и прибыльность интернет магазина зависит от правильной и своевременной корректировки цены товара, поэтому руководство несет достаточно большие умственные, а иногда и финансовые затраты для того, чтобы вручную посчитать оптимальную цену на текущий период. Большое количество конкурирующих фирм, магазинов и предприятий в целом приводят к тому, что приходится считаться с ними и для того, чтобы выжить в условиях такой конкуренции, необходима особая выработанная стратегия продвижения предприятия и его совершенствования, расширения ассортимента, а вместе с тем и продуманное ценообразование.

Так, например, можно назвать CRM-систему как 1С.Битирикс, где в разделе «Наценки» можно рассчитать автоматически цену на товар с желаемой наценкой. Данный модуль пользуется популярностью, так как есть возможности объединить товары в группы, учесть затраты на перевозку, хранение и саму себестоимость товара. Из этого можно сделать вывод, что разработка в области наценок и скидок является достаточно востребованной, в первую очередь для предприятий с большим объемом продаж и ассортиментом. Возвращаясь к примеру, стоит сказать, что какие-либо факторы, которые могут влиять на величину наценки, программа не высчитывает сама. Данная функция отдается пользователю, а значит, мы приходим к выводу о том, что тема разработки программы, которая могла бы рассчитать и порекомендовать желательную наценку при определенных

условиях, основываясь на конкретных данных, является своевременной и достаточно злободневной, особенно, когда идет речь об онлайн торговле.

Актуальность, поставленной в работе проблемы, позволяет определить объект, предмет, цель и задачи исследования.

Целью работы является разработка программного продукта для автоматизации корректировок наценки на товар при онлайн торговле.

В связи с этой целью решаются следующие **задачи**:

- рассмотреть понятие и теоретические аспекты торговой наценки
- изучить особенности онлайн торговли
- выявить факторы, влияющие на изменение торговой наценки
- разработать программу автоматизированной корректировки наценок

Объект исследования - торговая наценка.

Предмет исследования – факторы, влияющие на изменение наценки на товар.

Теоретической и методологической основой исследования послужили труды отечественных и зарубежных ученых в области исследования понятия торговой наценки, её видов, а также автоматизированных систем, созданных для предоставления рекомендаций пользователю.

В качестве основных **методов** исследования были использованы общенаучные методы анализа, синтеза, систематизация, классификация и обобщение теоретических данных, представленных в литературе. Кроме того, вся систематизированная информация была применена на практике, и был разработан алгоритм, основываясь на ней, позволяющий автоматически корректировать торговые наценки, исходя из нескольких критериев.

Практическая значимость обуславливается необходимостью своевременного изменения цен на товары в условиях онлайн торговли. Разработанный алгоритм может применяться любыми по своим масштабам предприятиями, которые реализуют свою продукцию на интернет площадке. Данный программный продукт значительно сократит время, которое уходит у персонала на исследование и корректировку цен, а также поможет

увеличить доходность предприятия за счет своевременного изменения наценок на товары. Особенно актуальным этот алгоритм будет для небольших магазинов, так как поможет владельцам сосредоточить все свои силы и возможности на расширении ассортимента вместо того, чтобы вручную пересчитывать цены продукции.

Курсовая работа состоит из введения и 2 глав – теоретической и практической, а также заключения.

Глава 1. Понятия наценки и Интернет-торговли.

1.1. Цена и ее основные характеристики.

Цена – это денежное выражение стоимости товара, то есть сумма, за которую покупатель готов приобрести данный товар, а продавец готов его продать[3].

При этом по характеру обслуживаемого оборота цены можно разделить на следующие группы:

- 1) на оптовые (отпускные) цены;
- 2) розничные цены;
- 3) закупочные цены;
- 4) цены на строительную продукцию;
- 5) цены на услуги (тарифы).

Если говорить о том, по какой цене организация реализует свои товары, то именно по отпускной (оптовой) цене предприятие продает свою продукцию другим предприятиям и организациям. Розничные цены же в большинстве своём предназначены для населения, продажи по таким ценам производятся в розничных магазинах и торговых точках, реже бывают и продажи небольшим компаниям и организациям. Рисунок 1.1. показывает схему образования розничной цены на товар.

Себестоимость продукции	Прибыль предприятия	Акциз	НДС	Оптовая или снабженческо- сбытовая над- бавка	Торговая надбавка
Оптовая (отпускная) цена предприятия					
Оптовая цена промышленности (цена оптовой организации, сбытовая цена)					
Розничная цена					

Рисунок 1.1. - Схема формирования розничной цены товара[3]

Как строится розничная цена, можно увидеть на схеме выше, и из этого вытекает определение розничной цены, однако, даже она может иметь

некоторые вариации. В некоторых случаях в оптовую цену уже включены некоторые надбавки или дополнительные затраты, которые потерпело предприятие из-за повышения стоимости комплектующих, например.

Основу цены продукции составляют издержки предприятия, затраты на производство конкретного товара. Именно себестоимость является определяющей в образовании цены, она становится основой, относительно чего можно уже двигаться дальше. Например, в промышленности доля издержек, т.е. себестоимости товара, составляет более 75% отпускной цены, а когда речь идет о железнодорожном транспорте, то более 87% [1].

Существуют зависящие от предприятия издержки, которые несет именно само предприятие, но также и внешние издержки, те, которые организация не контролирует и не может изменять самостоятельно. Поэтому идеальная цена продукции всегда должна быть больше минимальной, но меньше максимальной цены, для того, чтобы, во-первых, она была привлекательна для покупателей, а, во-вторых, покрывала всевозможные издержки предприятия. На рисунке 1.2 представлена схема образования цены товара.



Рисунок 1.2 - Схема формирования розничной цены [3]

Существует несколько основных методов расчёта цены товара предприятия, и они зависят в основном от условий рынка и целей самой компании или предприятия. Основными являются:

- 1) затратные методы;
- 2) рыночные методы;

3) эконометрические методы.

Метод, который включает в себя следующие понятия - «издержки плюс прибыль», метод целевого ценообразования, а также метод стоимости изготовления товара – является наиболее часто используемым и он называется затратный метод.

В свою очередь «издержки плюс прибыль» можно также разделить на 3 следующих метода:

- метод полных издержек;
- метод прямых затрат;
- метод предельных издержек.

Метод полных издержек основывается на сумме затрат постоянных и переменных, а также прибыли, на которую рассчитывает компания. Когда же мы говорим о методе прям затрат, стоит понимать, что он подразумевает также учет добавочной стоимости к товару, надбавки на товар, для получения прибыли. Постоянные же расходы в таком случае не распределяются по каждому товару, а возвращаются исходя из разницы между ценой продажи и переменными издержками предприятия. Это и называется добавленной или маржинальной прибылью.

Ещё один метод, основанный на себестоимости – метод расчёта цен на основе предельных издержек, он является чуть более сложным по своей сути и менее используемым, однако, он достаточно эффективен[14]. К тому же при таком образовании цены надбавка осуществляется только к предельно высокой себестоимости производства каждой последующей единицы уже освоенного товара или услуги. Такой метод будет эффективен в тех ситуациях, когда у нас есть гарантированный объём продаж по высокой цене, который покроет наши расходы.

1.2. Методы расчета наценок и скидок.

В условиях рыночной экономики внешняя среда часто меняется, а она в свою очередь оказывает непосредственное влияние на установление цены на товар. На это влияют различные факторы: изменения цены на сырье, сопутствующих товаров, а также условия поставки и размер заработной платы. Если же производитель хочет изменить цену на товар, то в договоре о сбыте продукции должны быть прописаны оговорки об увеличении цены на продукцию. Если такие имеются, то, основываясь на этих оговорках, мы можем определить новую цену товара по формуле 1.1.:

$$Ц_{п} = \frac{Ц_{к}}{100\%} \cdot \left(A \cdot \frac{З_{п}}{З_{к}} + B \cdot \frac{С_{с}}{С_{к}} + C \cdot \frac{Э_{п}}{Э_{к}} + D \cdot \frac{П_{п}}{П_{к}} \right),$$

Формула 1.1. [3]

где Ц_п – это цена товара во время поставки, Ц_к – цена изделия во время подписания контракта, З_п – заработная плата соответственно во время поставки товара, З_к – во время подписания контракта, С_п и С_к – цена материалов и сырья в те же сроки, Э_п и Э_к – стоимость электроэнергии, П_п и П_к – это постоянные затраты, которые учитываются в цене товара (затраты на электроэнергию, например), А, В, С, Д – удельный вес в структуре себестоимости товара соответственно затрат на заработную плату, материалы, электроэнергию, постоянные расходы[3].

Также в ценообразовании большую роль играет система скиду и надбавок. Цены, которые устанавливает производитель, существенно отличаются от реальных цен, по которым население приобретает тот или иной товар.

На данный момент существует более 20 различных видов скидок, которые широко применяются не только в Российской Федерации, но и во всем мире. Далее представлены самые основные из них:

1. Скидки за оплату товаров наличными. В таком случае размер скидки является индивидуальным и зависит от конкретных условий, но по различным данным она может достигать до 5% от изначальной цены. Такой тип скидок еще называют общим (или простым).
2. Оптовая скидка – такую скидку можно получить, если вы покупаете крупную партию товара (например, если покупаете более 2000 штук). В таком случае продавец экономит не только на хранении этой продукции, но и на транспортировке. С такими скидками стоит быть осторожным, так как нужно всегда внимательно следить за тем, чтобы сумма предоставленной скидки не превысила затраты на единицу продукции.
3. Дилерская скидка она предназначена для посредников и предоставляется за то, что они продают и распространяют информацию о нем непосредственно конечному потребителю. Такие скидки широко используются, когда продают машины и оборудование. Они обычно могут достигать 25% от розничной цены, в зависимости от марки автомобиля.
4. Специальная скидка предназначена особой группе лиц, в которой заинтересован продавец. Также сюда можно отнести скидки на пробные партии товаров, когда производитель хочет заинтересовать потребителя.
5. Сезонная скидка обычно предоставляется покупателю, если он покупает товар, который является несезонным в данный период времени (шорты в январе, например), и она может достигать 80% от розничной цены.

6. Скидка за оборот предназначена постоянным покупателям и только том случае, если он в течение определенного периода приобретают товар. Такие скидки могут составлять как несколько процентов на сельскохозяйственные товары, например, так и достигать 25%, когда речь идет об оборудовании.
7. Скрытая скидка она выражается в предоставлении клиенту каких-либо бесплатных услуг (например, бесплатная сборка шкафа для одежды).
8. Скидка за качество (Например, товар был некачественно расфасован или укомплектован).
9. Скидка при продаже оборудования, бывшего в употреблении. Такое оборудование обычно еще достаточно надежно и может прослужить еще долгие годы, при этом цена на него составляет не более 50% от цены аналогичного нового.

Наценка – это разница между, ценой, по которой определенный товар был продан, и его себестоимостью, она, как правило, выражается в процентном соотношении к цене продажи[14].

Чтобы рассчитать наценку на товар, можно воспользоваться формулой 1.2.:

$$\text{Markup Percentage} = \frac{\text{Sales Price} - \text{Unit Cost}}{\text{Unit Cost}} \times 100$$

Формула 1.2. [18]

Знание понятия наценки и умелое его использование является одним из основных знаний для работы в сфере бизнеса. Для того чтобы не уйти в убыток предприятию, наценки должны быть такими, чтобы они покрывали

издержки на произведенный товар. Другими словами, наценки нужны для того, чтобы производитель мог получить прибыль со своей продукции.

1.3. Особенности онлайн продаж и управление ценообразованием.

Интернет меняет устройство всего окружающего нас мира. Уже сейчас многие не могут представить свою жизнь без него. За несколько лет появилось и много компаний, которые предоставляют свои услуги и товары онлайн. Поэтому можно с уверенностью утверждать, что интернет-экономика имеет свои особенности, которые отличаются от традиционной торговли. Вот некоторые из них:

Избыточность. В условиях онлайн продаж очень часто можно замечать широкий ассортимент того или иного товара. Предложение товара становится избыточным на онлайн рынке, но также и его распространение уходит далеко за пределы каких-либо географических рамок. Чем больше становится того или иного товара, тем более ценным он становится, так как является наиболее доступным для потенциальных покупателей.

Прибыль. Чем больше новых предпринимателей приходит в сферу онлайн торговли, тем больше она растёт и развивается. Кроме того, обороты продаж увеличиваются, а как следствие и прибыль самих предпринимателей становится больше. Невысокие затраты в сравнении с традиционным способом продаж также помогают увеличить прибыль компаний.

Снижение цен. Как правило, после некоторого количества времени цены имеют тенденцию снижаться, если товар находится слишком долго на рынке. Спрос на этот товар со временем становится всё ниже, что и обуславливает снижение цен. Такая ситуация характерная, например, для смартфонов, ноутбуков и прочей техники. Именно поэтому приходится искать новые товары и продавать уже их.

Бесплатный товар. Из-за большой распространённости товаров на рынке появилось правило «бесплатный товар». Таким образом, больше потенциальных покупателей заинтересуется товаром, ведь чаще всего, даже если это не нужно, но это бесплатно, человек в силу своей психологии захочет попробовать. А в последующем сопутствующие товары и обновления принесут немалый доход.

Глобальная конкуренция. Помимо возможности продавать свой товар по всему миру, у Интернет-торговли есть и свой минус. За счёт того, что площадки продаж связаны между собой во всемирной паутине, намного легче становится выходить на этап – продажи за границу. Однако именно этот плюс сети и порождает большую конкуренцию. Так как насколько легко выходить на мировую площадку продаж, настолько же быстро появляются всё новые конкуренты и скорость их появления намного выше, чем в условиях традиционной торговли.

Подобные отличия онлайн продаж приводят к тому, что человеку нужно перестраивать свой бизнес, начиная от ценообразования, заканчивая базовыми принципами торговли. Можно выделить несколько категорий, по которым бизнес в Интернете и бизнес в офлайн режиме отличаются.

1. *Привлечение потребителей.* Так, если рассматривать офлайн торговлю, то можно выделить несколько этапов привлечения внимания потребителя:

- Реклама; фактор – расположение;
- Оформление витрин и визуальной составляющей магазина;
- Удобство торгового зала, расположение прилавков и комфортная организация пространства;
- Решение покупателя о будущей покупке будет зависеть от консультанта в магазине.

Если рассматривать онлайн магазины, то можно сказать, что визуальная составляющая является единой вместе с витринами, реклама расположена непосредственно на сайте. Она краткая и понятная, чаще всего сделана так, чтобы покупатель при покупке или простом посещении сайта запомнил эту информацию, причем не только общие характеристики товара, но и где, когда можно его приобрести. Чаще всего покупатель ищет именно низкие цены в интернет магазине, поэтому в электронной торговле не самую последнюю роль играют цены. Они явно указаны и являются одним из основных факторов.

2. *Воздействие на потребителя.* Так как в условиях онлайн продаж отсутствуют консультанты в торговом зале, то и психологического воздействия нет. То есть, такого фактора в Интернет-торговле вообще нет.

3. *Предоставление информации о продукте.* Снова возвращаясь к консультантам, можно понять, что знакомство с товаром в традиционных магазинах происходит лично. Всю информацию, как правило, человек получает от консультанта. В условиях Интернет-торговли потребитель самостоятельно изучает характеристики товара, описание и всю предоставленную информацию на сайте.

4. *Приобретение товаров.* В офлайн магазинах покупатель может самостоятельно изучить приобретаемый товар, осмотреть, проверить качество и приобрести именно тот, который ему по душе. В условиях же Интернета, потребители вынуждены полагаться на описание товара и отзывы, так как они сами проверить не могут товар. Поэтому, можно сказать, что потребитель идёт на осознанный риск получения не того товара или же товара не надлежащего качества. Однако этот риск чаще всего оправдывается относительно низкой ценой.

5. *Объемы затрат.* В условиях Интернет-торговли можно существенно экономить на аренде складов, помещений, рекламы. За счёт этого затраты компании снижаются. Часто магазины не имеют складов вообще, а следовательно и затрат на это не имеют. Большие организации арендуют складские помещения, но всё равно это несравнимо с затратами в условиях традиционной торговли.

6. *Посещаемость.* Очень часто в поисках нужного товара необходимо обойти несколько точек, чтобы найти именно тот, что нужен. Смотреть из дома на сайте существенно легче и удобнее. Во-первых, меньше физических нагрузок на организм. А, во-вторых, экономия времени.

7. *Расположение.* Расположение компании или магазина то же имеет особое значение в традиционной торговле в отличие от онлайн продаж. В этом случае магазин доступен из любой точки мира, и потребитель может совершить покупку в любой момент времени, совершенно не напрягаясь и не тратя усилий на разъезды.

8. *Компетентность консультанта.* В обычных магазинах, как правило, всю необходимую информацию рассказывает работник магазина. Именно от него в большей степени зависит решение потребителя о покупке, так как много решает осведомленность продавца о товаре, его отношение к работе, а так же количество времени, которое он способен уделить потребителю. Если говорить об онлайн сфере, то вся информация указывается на самом сайте. Там же есть и отзывы, то есть потенциальный покупатель не зависит от консультанта магазина.

9. *Зарождение бизнеса.* Намного легче создать онлайн компанию и предоставлять услуги, чем такую же в традиционном понимании. Более того, не только материальные затраты снижаются, но и временные рамки тоже.

Гораздо быстрее создать что-то в Интернете, чем реальный магазин с помещением, витринами и обслуживающим персоналом.

10. *Ассортимент.* В онлайн магазинах более широкий выбор товаров, чем в традиционных точках. Таким образом, человеку придется обойти несколько точек, прежде чем он найдет то, что он искал, а в условиях онлайн торговли, он может найти всё за пару поисковых запросов, возможно, даже в одном магазине.

Итак, можно сделать вывод, что благодаря развитию интернет площадок, масштабируемости бизнеса и большой распространенности товаров, становится особенно актуальным изучения ценообразования в таких условиях. Ниже приведена таблица с данными о влиянии на прибыль изменения разных факторов.

Параметр	Степень влияния на прибыль, %
Цена	11,19
Переменные издержки	7,8
Объем продаж	3,3
Постоянные издержки	2,3

Таблица 1.1. – Влияние 1% изменения некоторых факторов на прибыль. [7]

Исходя из этих данных (Табл. 1.1.) можно сделать вывод, что цену на товар, который реализуется с помощью онлайн торговли, можно представить функцией от нескольких параметров.

$$P_i = f(K_i; C_i; D_i; K_j; S_i),$$

Формула 1.3. [7]

где P_i — цена продукта; K_i — качественные характеристики товара; C_i — себестоимость; D_i — параметры спроса на товар; $K_j, j \in [1; n]$ — характеристики товаров-аналогов; S_i — параметр конкуренции[7].

Как правило, используют следующие формулы для расчёта цен.

Для начала стоит определить эффективность торговли, и поэтому нужно воспользоваться формулой 1.4., которая помогает понять по грубым подсчётам, на какую прибыль можно рассчитывать.

$$\text{Прибыль} = \text{Спрос} * (\text{Выручка} - \text{Расходы})$$

Формула 1.4. [12]

Допустим, ежемесячно товар ищут примерно 15 000 человек. Эти данные можно узнать по статистике запросов в Google или Яндекс. Если предположить, что хотя бы треть компания сможет привлечь, то есть 5 000 человек. Тогда, если отталкиваться от средней конверсии продаж в 1%, то это уже примерно 50 продаж ежемесячно. Если средний чек покупателя будет около 6 000 рублей и наценка будет 40%, то прибыль составит 240 000 рублей в месяц.

Если организация отталкивается от конкретного процента рентабельности изготовления товаров, то расчет реализуемой стоимости может быть выполнен по формуле 1.5.:

$$P = C (1 + R/100)$$

Формула 1.5. [7]

где P — цена продажи; C — расходы на 1 продукта; R — ожидаемая рентабельность[7].

По следующей формуле 1.6. можно рассчитать точку безубыточность компании:

$$BSV = FC / TR$$

Формула 1.6. [7]

где BSV — точка безубыточности; FC — постоянные издержки; TR — валовая прибыль[7].

Если компания реализует несколько товаров, то можно использовать усложненный вариант расчёта точки безубыточности (Формула 1.7.):

$$BSV = FC / (S \times TR)^A + (S \times TR)^B + (S \times TR)^C + (S \times TR)^D$$

Формула 1.7. [7]

где S — процент количества продаж для товаров А, В, С и D[7].

1.4. Анализ онлайн рынка в России и в мире.

В 2014 году было проведено исследование о то, какая же возрастная группа является основной, когда дело касается покупок в интернет сети. По итогам исследованием были сделаны выводы, что самая платежеспособная и активная группа это юноши и девушки в возрасте от 22 до 44 лет[7].

Мужчины (46%)		Предпочтения онлайн покупателей	Женщины (54%)	
9%	18-24	Самая молодая группа Интернет-покупателей насчитывает 19% от общего числа онлайн покупателей. В Интернете их привлекает приобретение купонов и товаров для хобби	10%	18-24
16%	25-34	Более трети (34%) клиентов Интернет-магазинов принадлежат данной категории, которая является самой активной. Наиболее популярными покупками являются товары для детей, товары для дома и одежда и обувь.	18%	25-34
10%	35-44	Почти четверть покупателей (23%) наиболее часто приобретают различные товары в Интернет-супермаркетах и Интернет магазинах электроники.	13%	35-44
8%	45-54	Одинаковое число мужчин и женщин данного возраста (16% от общего числа онлайн покупателей) являются частыми посетителями Интернет-магазинов одежды и сервисов по бронированию отелей.	8%	45-54
3%	55-64	Представители данного возрастного сегмента – самая малочисленная группа покупателей в Интернете (8%). Области их интересов распространяются на приобретение в Интернете товаров для дома и книг.	5%	55-64

Таблица 1.2. – Статистика предпочтений онлайн пользователей[1]

Согласно проведенному исследованию самыми популярными товарами на онлайн-рынке являются товары для дома и детские товары[1]. Также были сделаны выводы о том, что данная категория лиц, производящих покупки в интернет магазинах, имеют ежемесячный доход до 30 тыс. руб. Не выявлено особых предпочтений у тех, кто имеет доход чуть выше, так как они покупают всё. Меньше всего людей делают покупки в онлайн-магазинах те, кто зарабатывают более 65 тыс. в месяц, их цель в основном это авиабилеты и бронирование отелей.

Рынок Интернет-торговли активно развивается не только в России, но и за рубежом, так некоторые компании говорят о том, что уж к 2020 году товарооборот достигнет доходности в 4 трлн. долл.

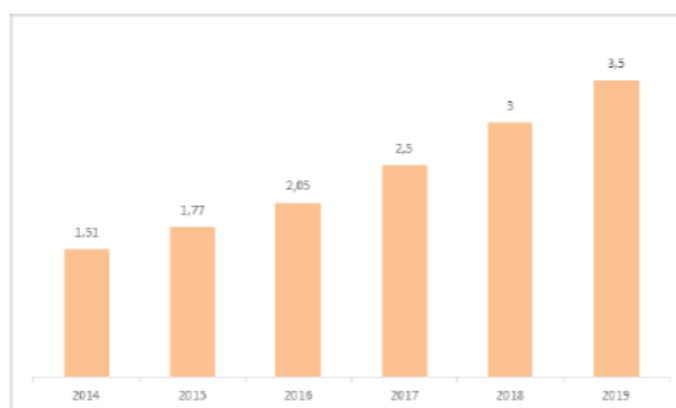


Рисунок 1.3. – Рост мирового рынка розничной интернет торговли, трлн. долл. США[1]

Все больше и больше будет развиваться сфера Интернет-торговли, в основном за счёт притока новых клиентов, которые будут пользоваться смартфонами для того, чтобы заказать себе тот или иной товар.

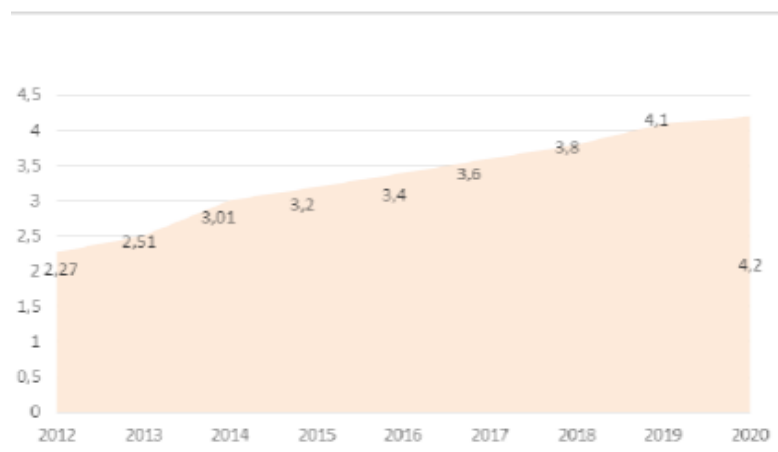


Рисунок 1.4. – Рост числа пользователей интернета в мире[7]

Более того, все также прогнозируется приток продавцов на рынок онлайн торговли и развитие все большей конкуренции, так как пользователи имеют возможность быстро сравнивать цены разных производителей и выбирать более дешевый товар. Если же говорить про крупных игроков, то они по-прежнему будут на высоте, все более и более усиливая свои позиции при помощи новых проектов.

Глава 2. Разработка системы автоматической корректировки наценок.

2.1. Описание текущей ситуации и существующих алгоритмов.

В настоящее время уже существуют системы, которые помогают не только установить наценку, исходя из предпочтений пользователя, но также и отследить изменение продаж.

Примером такой программы является 1С:Управление торговлей 8, где есть возможность установления наценки пользователем, в зависимости от желаемой цены в итоге. Он также может менять наценку и изменения могут применяться не только к одному конкретному товару, а сразу к нескольким, в зависимости от категории.

Сразу стоит сказать, что система удобная, но для того, чтобы разобраться с полным функционалом потребуется время, хотя та часть которая отвечает за корректировку наценки на товар понятна. Программа позволит посчитать надбавку в 3 разных случаях: в зависимости от цены поступления, от другого вида цен и также есть вариант ручного назначения цены. Кроме того, также можно учитывать НДС сразу или ставить флажок уже непосредственно при назначении наценки, но так или иначе НДС будет входить в конечную стоимость продукции.

Однако даже принимая во внимание все эти плюсы, все равно можно сделать вывод, что основная часть работы все еще остается на человеке. Именно от пользователя зависит конкретная наценка на товар, программа не рассчитывает ее исходя из полученных изменений, эта задача возлагается на человека.

Более того, нужно еще принимать во внимание специфику онлайн магазинов. Так на сегодняшний день уже существуют программы, которые позволяют отслеживать цены конкурентов, поставщиков автоматически. И при

назначении наценки продавец может на них ориентироваться без особых затрат времени на сравнение и исследования предложения конкурентов.

Такие программы как E-Trade PriceList Importer или Quad-CRM и не только, их достаточно много, предоставляют возможность отслеживать изменения цен конкурентов и поставщиков без нагрузки на человека. Конечно же, на этом не заканчивается их функционал, спектр возможностей достаточно обширен. Например, существует обработка прайс-листов, их составление и еще несколько возможностей, которые помогают автоматизировать работу интернет-магазинов.

Однако даже такие обширные возможности не дают вычислить подходящую наценку автоматически. Как правило, большинство продавцов ее исходя из того, что она должна быть около 50%, а дальше основываясь на отчетах о продажах приходят к своей идеальной комбинации.

Алгоритм прогнозирования наценки является одним из наиболее успешных примеров автоматизированного назначения наценки [6]. То есть имея данные о количестве продаж и наценке за 2 определенных интервала времени мы можем воспользоваться этим алгоритмом и он предложит оптимальную наценку исходя из полученных данных.

В этом алгоритме существует переменная γ , которая и отвечает за степень изменения наценки. Другими словами, если пользователю хочется увидеть существенные изменения в наценке, то стоит установить большое значение и также наоборот.

Я решила смоделировать этот алгоритм, используя язык программирования python, чтобы увидеть наглядные данные. В результате можно увидеть, что при некотором задании первичных данных можно наблюдать рост прибыли.

Если установить функцию $y = A/(1+x^{**2})$, где y = количество продаж, x – наценка на товар, A – коэффициент, в зависимости от которого происходит

изменение в функции. $A = A + 6 * \text{np.random.rand}()$. Тогда можно будет увидеть следующую ситуацию. Изначально, пусть, $A = 100$, а $x = 0,1$.

Наценка на товар постепенно увеличивается (Рис. 2.1.). Повышение происходит плавно, нет никаких больших разбросов.

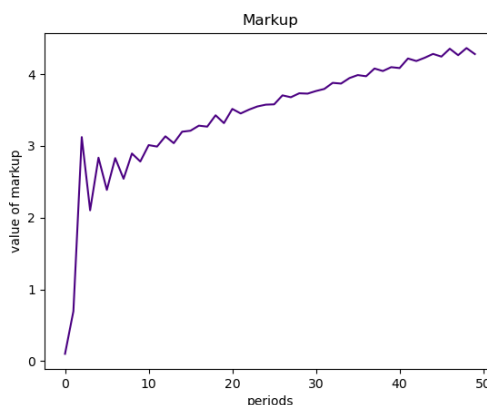


Рисунок 2.1. – График наценки

Далее, стоит обратить внимание на значения количества продаж (Рис. 2.2.). Оно также изменилось, но в отличие от наценки, это значение остается примерно одинаковым, слегка меняясь за счет коэффициента A . В данном случае большой спад продаж в начале обусловлен исключительно тем, что значение 100 было слишком высоким в силу низкой наценки. Поэтому как только наценка поднялась все встало на свои места и никаких резких скачков мы уже не наблюдаем.

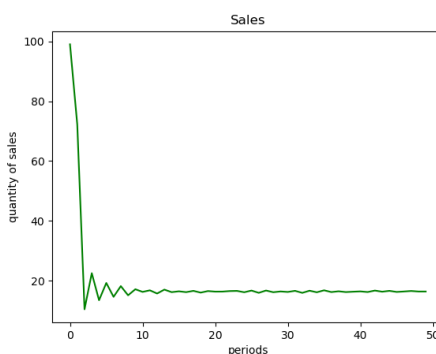


Рисунок 2.2. – График объёма продаж

График прибыли выглядит следующим образом (Рис. 2.3.). Здесь мы видим ее увеличение в основном за счет того, что спрос относительно стабильный, а наценка небольшими шагами повышается.

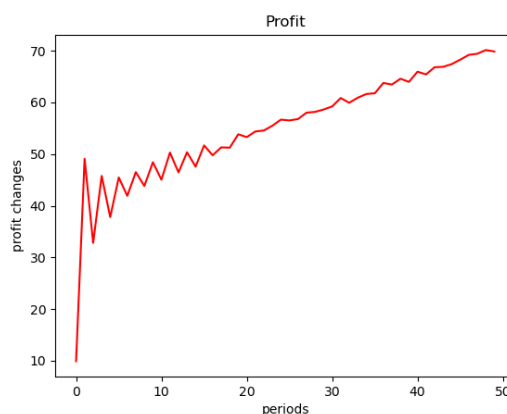


Рисунок 2.3. – График прибыли

Таким образом, цель моей работы – разработать систему наценки, учитывающую несколько факторов. Задачами являются:

1. Анализ факторов, влияющих на наценку в условиях интернет торговли.
2. Выделение основных факторов.
3. Разработка алгоритма.
4. Построение модели готовой системы и ее анализ.

2.2. Выявление и анализ требований к будущей системе.

Для того чтобы выстроить требования к системе, первоначально мне нужно было проанализировать факторы, которые влияют на наценку, и, основываясь на этих данных, понять, какие из них стоит учесть в условиях онлайн торговли. В ходе своей работы я пришла к следующим выводам относительно этих факторов.

Во-первых, одним из основных критериев успешной наценки на товар является учет расходов организации на конкретный товар (транспортировка товара, аренда склада, отправка получателю, заработная плата персонала,

который обслуживает сайт). Чаще всего предприниматели, которые начинают рассчитывать наценку на товар думают, что их расходы в сравнении с расходами обычных магазинов намного меньше. И либо не учитывают их совсем, делая наценку ориентируясь только на цены конкурентов, что потом приводит к банкротству магазина и его закрытию, либо учитывают, но не полностью (только заработную плату рабочим, не учитывая траты на доступ в интернет, например), тем самым ухудшая свое финансовое положение.

Расходы магазина являются основой для построения наценки, так как при любых обстоятельствах наценка на товар не должна быть ниже этого значения. В противном случае, магазин уже будет работать в убыток.

Во-вторых, также стоит ориентироваться на цены конкурентов. Так как в Интернете покупателю стало еще проще сравнивать цены, и он может быстро найти тот магазин, где будет дешевле, и купить товар там. Однако такая ситуация происходит постоянно, даже если цены магазина очень схожи с ценами конкурентов, так как на выбор покупателей влияет не только цена товара, но и многие другие факторы (надежность продавца, удобство доставки).

Такой критерий, как цена конкурента стоит учитывать для того, чтобы не завышать цену в своем магазине, так как в этом случае продажи сильно снизятся. Можно сказать, что такая наценка – это верхний рекомендуемый предел, который стоит устанавливать, для того, чтобы не терять прибыль.

Следующий фактор, который стоит учитывать при назначении наценки на определенный товар – это эластичность спроса. Существуют понятия эластичности и неэластичности спроса.

Он считается эластичным, если отношение изменения объема спроса к изменению цены больше 1 (Рис. 2.4). Если же ситуация обратная и при

снижении цены объем спроса увеличивается незначительно, то такой спрос называется неэластичным (Рис. 2.4.).

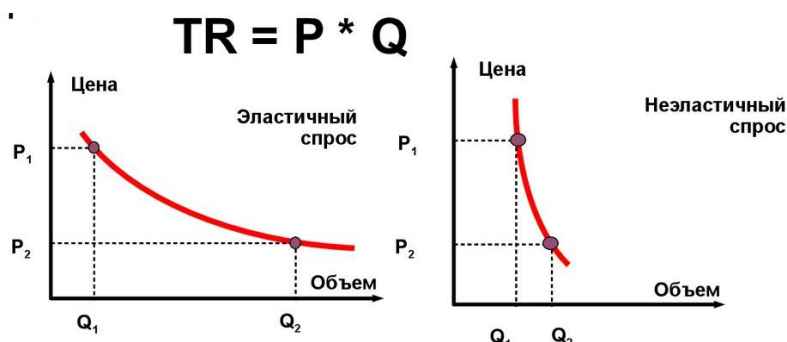


Рисунок 2.4. – Эластичность спроса по цене [14]

Также если рассмотреть эти два понятия с точки зрения выручки, то можно увидеть, что если спрос неэластичен, то при снижении цены на товар, выручка уменьшится и наоборот, если спрос эластичный, то понижение цены приведет к росту выручки.

Можно выделить несколько групп товаров, на которые спрос неэластичен. Такими товарами являются товары первой необходимости (лекарства, электричество), трудно заменяемые товары (электрические лампочки, бензин), и те товары, стоимость которых незначительная для бюджета (карандаши, зубные щётки)[14]. При назначении любой наценки на эти товары объем спроса будет примерно одинаковым, так как независимо от цены покупатели все равно будут приобретать их.

Однако в условиях интернет торговли покупатели могут выбирать товар часами, у них есть время на сравнение цен, отслеживание их изменения в течение некоторого периода времени и, таким образом, покупатели все равно чаще всего приобретают больше, когда цена снижается, т.е. спрос становится эластичным. В данном случае, особенность интернет продаж для магазинов заключается в том, что практически на все товары спрос эластичен и при снижении цены неизбежно будет увеличиваться объем спроса. А, значит,

фактор неэластичности спроса можно не учитывать в условиях интернет торговли, так как вероятность того, что спрос на товар будет неэластичным крайне мала.

Помимо расходов магазина, цен конкурентов и эластичности спроса, стоит также учитывать и «сезонность» некоторых товаров. Это необходимо для того, чтобы продавец не терял потенциальную прибыль. Основываясь на расходах и ценах конкурента, он сделает нужную наценку и скорее всего продажи будут достаточно стабильными. Его магазин будет приносить прибыль, но она могла бы быть больше, если бы вовремя делалась правильная наценка на тот товар, за который в определенный период времени покупатель готов заплатить больше обычного, так как этот товар является «сезонным».



Рисунок 2.5. – Статистика увеличения объёмов спроса в «сезонные» периоды [14]

По статистике на 2019 год в зависимости от сезона или же в преддверии праздника спрос на определенные товары может существенно повышаться (Рис. 2.5.) и даже, если наценка будет 100-200%, то спрос на товар все равно существенно не упадет, так как в период «сезонности» покупатель готов заплатить больше обычного. Так, например, можно видеть (Рис. 2.5), что спрос на товары, относящиеся к категории «Новый год и Рождество» (новогодние игрушки, сувениры, гирлянды), повышается на 48%. В период

Пасхи спрос увеличивается на 9%, поэтому и цены на яйца, краску для яиц и творог взлетают значительно.

Итак, этот фактор не является основным в построении наценки на товар, но помогает продавцу существенно увеличить прибыль в определенные периоды времени. У каждого продавца одна из целей – это максимизация прибыли магазина, и как раз за счет учета фактора «сезонности» это цель и будет выполняться в случае с наценкой на товар.

Таким образом, проанализировав факторы, влияющие на наценку товара в условиях онлайн торговли и выделив главные из них, я пришла к выводу, что стоит учитывать расходы магазина, цены конкурентов, а также «сезонность» товара для того, чтобы наценка была максимально оправданной со стороны продавца, и цена оставалась привлекательной для потребителя.

Кроме того, существуют ситуации, когда не так важны расходы предприятия, но нужно обращать особое внимание на изменение спроса и вовремя реагировать с целью получения максимальной прибыли. Так, например, если рассмотреть продажу онлайн билетов, то можно заметить, что в данном случае крайне важно отслеживать динамику изменения спроса и подбирать максимально выгодную наценку.

Поэтому после анализа факторов, влияющих на наценку товара, можно сказать, что будущая система должна:

1. Рассчитывать рекомендуемую наценку, учитывая расходы магазина.
2. Рассчитывать наценку, учитывая «сезонность» товара.
3. Рассчитывать наценку, учитывая цены конкурентов.
4. Рассчитывать наценку исходя из данных за прошлые периоды.
5. Рассчитывать наценку, максимизирующую прибыль, путем отслеживания динамики изменения объема спроса.
6. Получить результирующую наценку, учитывая все факторы.

7. Пользователь должен иметь возможность просматривать историю рекомендуемых наценок и объёмов продаж с этой наценкой.

Исходя из технического задания и требований к системе, я решила сделать use-case diagram с помощью языка моделирования UML. Для построения диаграммы была выбрана программная платформа моделирования StarUML.

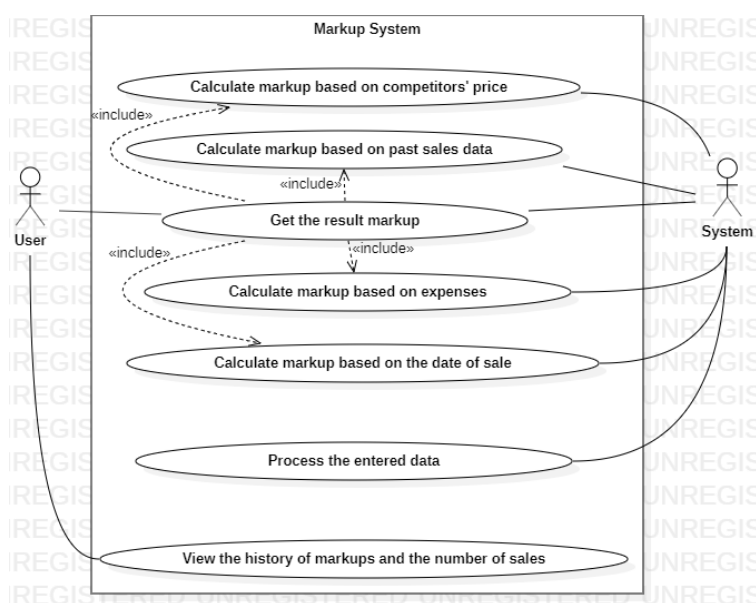


Рисунок 2.6. – Use case diagram для системы многокритериальной наценки.

Итак, пользователь системы должен получить конечный результат рекомендуемой наценки, а также иметь возможность просматривать историю рекомендованных наценок и объёмов продаж (Рис. 2.6.). Главная задача системы – посчитать итоговую наценку на товар. Это в свою очередь включает в себя подсчет наценки по расходам, наценки по предполагаемой дате продажи (проверка «сезонности»), наценки относительно цен конкурентов, а также расчёт наценки исходя из данных по количеству продаж и наценок за предыдущие периоды для максимизации прибыльности предприятия с помощью отслеживания изменений в спросе.

Далее были определены следующие классы в системе (Приложение 1).

ExpensesMarkup – класс наценки по доходам. Имеет следующие атрибуты:

- int costPrice – себестоимость товара.
- int totalExpenses – сумма всех расходов.
- int implementationNumber – количество продукта реализованного.

Также класс имеет функции:

- getCostPrice()
- getTotalExpenses()
- getImplementationNumber()
- getExpenseMarkup() – подсчитывает наценку исходя из расходов.

Competitor'sPriceMarkup – класс наценки по цене конкурента. Атрибуты:

- int costPrice – себестоимость.
- int competitor'sPrice – цена, на которую ориентируется пользователь.

Функции класса:

- getCostPrice()
- getCompetitor'sPrice()
- getCompetitor'sPriceMarkup() – подсчитывается наценка на основе данных о цене конкурента.

SeasonalMarkup – класс наценки по «сезонности» товара. Атрибуты класса:

- string productName – название товара.
- string date – дата продажи товара.

Функции:

- getProductName()
- getDate()
- getSeasonalMarkup() – подсчитывает наценку на основе «сезонности».

Этот класс также использует таблицу с распределением товаров по категориям и в зависимости от этого определяет, стоит ли делать наценку по этому критерию.

ResultMarkup – класс, вычисляющий итоговую наценку. Атрибуты класса:

- int expensesMarkup
- int seasonalMarkup
- int competitor'sPriceMarkup
- int firstMarkup – наценка за позапрошлый период реализации.
- int secondMarkup – наценка за прошлый период реализации.
- firstNumberOfSales – количество продаж за позапрошлый период.
- secondNumberOfSales – количество продаж за прошлый период.

Функции класса:

- int getMarkupForFirstPeriod() – подсчёт наценки на товар, который продается впервые.
- int getRecomMarkup() – подсчёт наценки на товар, который имеет историю продаж.
- void showResultMarkup()

Помимо 4 классов существуют также класс Main, откуда запускается программа.

После того, как диаграммы вариантов использования и классов были составлены была создана диаграмма активности, чтобы представлять, каким образом будут взаимодействовать пользователь с системой (Приложение 2).

1. Пользователь вводит себестоимость товара.
2. Пользователь вводит примерные сроки реализации товара.
3. Пользователь вводит расходы по каждому пункту.
4. Система суммирует все затраты.
5. Система подсчитывает наценку по расходам.

6. Пользователь вводит цену конкурента, на которую он ориентируется.
7. Система подсчитывает наценку относительно этой цены.
8. Пользователь вводит название товара.
9. Пользователь вводит дату продажи товара.
10. Система проверяет товар на сезонность.
11. Если товар уже продавался, пользователь вводит наценки за прошлые периоды.
12. Система считает рекомендованную наценку.
13. Если товар не продавался, то подсчитывается наценка на первый период времени.
14. После этого система сохраняет полученные данные.

С помощью построенной диаграммы последовательности (Приложение 3) можно понять жизненный цикл объектов и их взаимодействие.

На диаграмме представлены:

- Actor User – пользователь системы.
 - Объект ExpenseMarkup – класса наценки по расходам.
 - Объект Competitor'sPriceMarkup – класса наценки по цене конкурента.
 - Объект SeasonalMarkup – класса наценки по «сезонности».
 - Объект ResultMurkup – класса результирующей наценки.
1. Пользователь последовательно вводит данные о себестоимости, расходах и сроках реализации.
 - 1.1. Подсчитывается наценка по расходам (функция getExpenseMarkup()).
 2. Пользователь вводит данные о цене конкурента.
 - 2.1. Подсчитывается наценка по цене конкурента, на которую ориентируется пользователь (функция getCompetior'sPriceMarkup()).
 3. Пользователь вводит последовательно данные о названии товара и дате продаж товара.

3.1.Подсчитывается наценка по «сезонности» (функция `getSeasonalMarkup()`).

4. Выводится рекомендуемая наценка на товар согласно введенным данным (функция `showresultMarkup()`).

Далее была составлена диаграмма состояний системы (Приложение 4) в течение всего взаимодействия и работы самой системы. Система может находиться в следующих состояниях:

1. Ожидание данных от пользователя.
2. Введенные данные обработаны.
3. Итоговая рекомендованная наценка посчитана.
4. Полученные данные сохранены.

Все переходы между состояниями отражены на диаграмме. Существует лишь один триггерный переход, когда были введены не все данные и система возвращается в состояние «Ожидание данных от пользователя». Во всех остальных случаях система последовательно в процессе работы переходит из одного состояния в другое.

После того, как стало ясно, какой функционал должен быть, можно приступить к созданию будущей системы многокритериальной наценки.

2.3. Реализация системы наценок.

Принимая во внимание, все учитываемые факторы, которые влияют на расчет рекомендуемой наценки, во время разработки алгоритма, стало очевидно, что существуют две ситуации, в которых данные, введенные пользователем, будут обрабатываться по-разному.

Так как предполагается, что система должна рассчитывать наценку не только по предыдущим периодам, но и в тех ситуациях, когда таких данных нет. Например, в магазин поступил новый товар, который ранее не продавался, и

нужно решить, какую первоначально сделать наценку. Итак, существует 2 ситуации:

1. Товар уже продавался, и существуют данные о предыдущих наценках и количествах продаж.
2. Данных о продаже за прошлые периоды нет.

В первом случае алгоритм будет короче и станет основой для ситуации, когда данных о продажах товара еще нет. После того, как пользователь введет все необходимые данные, наценка будет рассчитываться следующим образом:

1. Рассчитать наценку по расходам
2. Рассчитать наценку по данным о предыдущих продажах на следующий период
3. Проверить товар на «сезонность»
4. Получить результирующую наценку и сохранить ее.
5. Сохранить данные о количестве продаж после установления наценки

В этом алгоритме требуются данные о расходах и предыдущих продажах, а также даты продаж для того, чтобы проверить на «сезонность» товар. В данном случае нам не требуются данные о ценах конкурентов, потому что основой служит история продаж магазина, и, когда количество продаж будет уменьшаться, система сама предложит сделать наценку ниже. Для того чтобы не уйти в убыток тут рассчитывается наценка по расходам.

Как только будет получен результат на будущий период, а затем введены данные о продаже, алгоритм повторится, начиная с 1 пункта, и так каждый раз заново. Таким образом, сохраняя данные о наценках и продажах, пользователь также сможет просматривать историю и делать выводы о том, стоит ли закупать товар снова и продавать его, приносит ли он желаемую прибыль.

Во втором случае, когда пользователь введет все необходимые данные о товаре, алгоритм подсчета наценки будет следующим:

1. Подсчитать наценку исходя из расходов магазина.
2. Подсчитать наценку исходя из цен конкурентов.
3. Проверить товар на «сезонность».
4. Получить итоговую результирующую наценку на первый период времени.
5. Сохранить первоначальную наценку и объем продаж при условии такой наценки на товар.
6. Рассчитать наценку исходя из данных о наценке и продажах за прошлый период.
7. Скорректировать наценку, учитывая расходы организации и дату продажи.
8. Сохранить полученные результаты о наценке и данных о количестве продаж.

Пункты с 1 по 5 являются обязательными только на этапе расчета самой первой наценки на товар, когда не имеется истории продаж о нем. Далее после ввода пользователя система будет автоматически рассчитывать наценку, повторяя пункты с 6 по 8.

В данном случае, когда первый раз будет рассчитываться наценка, основным критерием для получения результирующей будет цена конкурента. Она является верхней границей наценки. Но также будет рассчитываться и наценка по расходам для того, чтобы магазин не работал в убыток. Она станет нижней границей рекомендуемой наценки. Проверка товара на «сезон» будет лишь корректирующим фактором в случае, если на период продажи обычно бывает повышенный спрос на данный товар.

После этого результат будет основываться на данных о продажах и рекомендованной наценке. Суть алгоритма будет такой же, как и в первом

случае, просто пользователь не вводит прошлые данные, а система сама рассчитывает наценку и сохраняет ее.

После того, как был составлен алгоритм поведения системы, можно приступить к написанию кода программы. Будет несколько функций: наценка по расходам, наценка относительно цены конкурента, наценка в случае «сезонности» товара.

Функции расчет наценки по расходам и относительно цены конкурента достаточно просты, так как существуют формулы, по которым можно быстро рассчитать. Главное, знать об их существовании, программа же позволит быстро и без усилий самого человека рассчитать нужные значения.

Сначала стоит разобрать, как сделать такую наценку, чтобы покрыть расходы магазина на этот товар. Такие расчеты могут быть достаточно сложными с точки зрения внедрения их в программу, потому что иногда продавец покупает одно количество товара, потом другое, а значит и период времени, за который будет реализован весь товар тоже разное.

Например, если у нас есть 120 единиц товара, которые предположительно будет распродан за месяц и затраты магазина на поставку этого товара, отправку получателю, а также постоянные расходы, то можно рассчитать наценку по формуле. В функции программы это выглядит следующим образом:

```
def get_expense_markup(self):  
    rec_markup = self.total_expenses / (self.implementation_number * self.cost_price)  
    * 100  
    return rec_markup
```

Здесь в классе имеется информация о сумме расходов, а также стоимость товара за ту партию, которая будет распродана, то есть `cost*implementation_number` - это себестоимость 1 штуки умноженная на количество товара.

Однако становится сложно рассчитывать, если период, за который будет продано данное количество товара не 1 месяц, а 2 недели или 1,5 месяца. Поэтому для упрощения программы и расчетов, пользователю предлагается ввести не то количество, которое он закупил, а сколько единиц будет продано за месяц и себестоимость товара. Таким образом, намного проще учесть постоянные расходы магазина, которые, как правило, ежемесячные.

При вводе расходов пользователю также предлагается перечень пунктов, который позволит не забыть пользователю какие-либо траты. В этот перечень входят:

- Хостинг
- Доступ в интернет
- Бухгалтерское сопровождение
- Затраты на рекламу
- Администрирование сайта
- Расходы на доставку товара
- Расходы на отправку товара получателю
- Дополнительные расходы

Когда пользователь закончит ввод данных, программа посчитает сумму.

Далее нужно рассчитать наценку исходя из цен конкурента. Цены конкурентов очень удобно смотреть, особенно в интернете. Например, с помощью Яндекс.Маркет и других подобных сервисов, где можно отсортировать товар по цене и увидеть наименьшую и наибольшую цены. Выбирая цену, пользователь вписывает в поле нужное значение.

Для этих расчетов тоже существует специальная формула и код выглядит следующим образом:

```
def get_competitor_price_markup(self):  
    rec_markup = (self.comp_price - self.cost_price) / self.cost_price * 100  
    return rec_markup
```

То есть для того чтобы получить рекомендуемую наценку, нужно посчитать разницу цены конкурента и себестоимости товара, поделить все на себестоимость.

Дальше нужно прописать функцию учета «сезона». Для этого первоначально нужно создать списки. В таблице Excel мы делим товары на группы, чтобы потом при сравнении, можно было понять «сезонный» это товар или нет.

	23 февраля (20 - 25 февраля)	8 марта (1-10 марта)	Пасха (апрель)	1 мая (28 апреля - 7 мая)
Алкоголь		Набор шоколадных конфет	Кулич	Набор для пикника
Брелок в виде солдата		Шампанское	Набор для украшения яиц	Мангал
Сувенир в виде солдата		Вино	Посыпка для куличей	Шампура для мангала
Брелок в виде звезды		Мармелад	Глазурь для кулича	Гамак
Брелок в виде георгиевской ленты		Цукаты в коробочке	Краска для яиц	Уголь древесный
Брелок в виде оружия		Подарочный набор сладостей	Сувенир в виде цыпленка	Пластиковая посуда
Сувенир в виде звезды		Открытка	Термонаклейки	Лопата
Сувенир в виде георгиевской ленты		Подарочный пакет	Сувенир в виде кролика	Грабли
Сувенир в виде оружия		Свеча	Сувенир в виде яйца	
Сувенирная кружка "С днем Защитника Отечества"		Парфюмерно-косметический набор	Декорация в виде яйца	
Игрушечная машина			Мягкий кролик	
Игрушка Солдатик			Игрушечный цыпленок	
Игрушечное оружие			Корзина для украшения на Пасху	
Открытка "С 23 февраля"				
Фляжка				
Флешка				
Футболка мужская				
Носки мужские				

Таблица 2.1. – Таблица «сезонных» товаров

Пример того, как могут быть сгруппированы товары (Табл. 2.1.). В зависимости от специализации магазина и его ассортимента списки могут корректироваться. После того, как список составлен все, что остается сделать программе – это пробежаться по ним и, если товар пользуется повышенным спросом в данный период времени, то можно смело делать наценку в 100%, это самое минимальное значение. Если же товар не проходит по критериям, то функция вернет 0, что будет означать, что при расчете наценки стоит обратить внимание на другие факторы.

Фрагмент функции:

```
def get_seasonal_markup(self):
    wb = openpyxl.load_workbook('./Seasonal.xlsx')
```

```

sheet = wb.active

number_date = int(self.date[0:2].strip())
month_date = self.date[2:len(self.date)].strip()
rec_markup = 0

if month_date == "ноября" or month_date == "декабря" or month_date ==
"января":
    for row in sheet.rows:
        if row[0].value == self.name:
            rec_markup = 100

```

В конструктор сезонной наценки придет дата, введенная пользователем и название товара. После небольшой обработки можно будет с помощью функции класса просчитать «сезонность».

Расчет наценки на основе прошлых данных происходит по предиктивному алгоритму назначения наценки [6, с 2].

$$d[k+1]=d[k]+\gamma*(d[k]*N(d[k])-d[k-1]*N(d[k-1]))$$

Формула 2.1 [6]

Функция в программе выглядит следующим образом:

```

def get_rec_markup(self, gamma):
    rec_markup = self.last_markup + gamma * (self.last_markup * self.last_num -
self.pre_last_markup * self.pre_last_num)

    if rec_markup < self.exp_markup:
        rec_markup = self.exp_markup
    if self.seasonal_markup > rec_markup and self.last_num >= self.pre_last_num:
        rec_markup = self.seasonal_markup

    return rec_markup

```

Расчет наценки по расходам вызывается для того, чтобы определить граничную точку, до которой может опуститься наценка. А также проверяем «сезонность» нашего товара, чтобы не упустить прибыль в случае, если объем спроса увеличивается.

В итоге, будет считаться наценка исходя из прошлой наценки и количества продаж, дальше проверяется, не меньше ли она нижней границы, рассчитанной по расходам. Если меньше, то назначается наценка по расходам, если больше, то проверяется «сезонность». Если товар не «сезонный», то просто сохраняем это значение, если же это период повышенного спроса на этот товар, то ставим наценку 100% - минимально возможную (см. Приложение 1).

Данная функция особенно полезна тем, что она позволяет максимизировать прибыль. В некоторых случаях не так важны расходы, зато особенно ценным является своевременное изменение наценки на товар для того, чтобы максимизировать прибыль. Хотя в реальности мы не можем знать функцию спроса, нам доступны только отдельные точки, но с помощью формулы 2.1. алгоритм функции будет каждый раз близок к максимально возможному доходу. Так как изначально он построен так (Формула 2.2):

$$\text{Доход} = \text{Наценка} * \text{Количество продаж} \rightarrow \max$$

Формула 2.2

Так как при онлайн продажах практически все товары имеют эластичный спрос, то мы получим, что при увеличении наценки, количество продаж будет снижаться и наоборот. Алгоритм позволит отслеживать колебания в спросе и будет предлагать наиболее выгодную наценку на следующий период.

Еще одна функция в классе ResultMarkup – это подсчёт наценки, когда данных нет. То есть считается самая первая наценка основываясь на данных о затратах, сезоне продаж и цене конкурента. Логика схожа с предыдущей функцией за исключением того, что основой является цена конкурента, а не максимизация прибыли.

```
def get_first_markup(self):
```



```

recom_markup = self.comp_markup

if self.exp_markup > recom_markup:

    recom_markup = self.exp_markup

elif self.seasonal_markup > recom_markup:

    recom_markup = self.seasonal_markup

return recom_markup

```

Итак, спустя время сформируются сохраненные данные, и пользователь сможет просмотреть историю рекомендованных наценок. Полный код программы можно посмотреть в Приложении 5.

2.4. Тестирование системы и сравнительный анализ.

После составления алгоритма и написания кода программы, нужно смоделировать ситуацию, чтобы увидеть правильно ли ведет себя система.

Итак, пусть расходы магазина составляют 12600 с учетом доставки и отправки товара, а также месячных расходов. Товар уже продавался ранее и наценка была 40%. Пусть функция объема спроса будет $y = A/(1+x^{**2})$, где начальное значение $A = 30$ и $A = A + 6*nr.random.uniform(0, 1)$. Тогда график изменения наценки будет выглядеть следующим образом (Рис 2.7.):

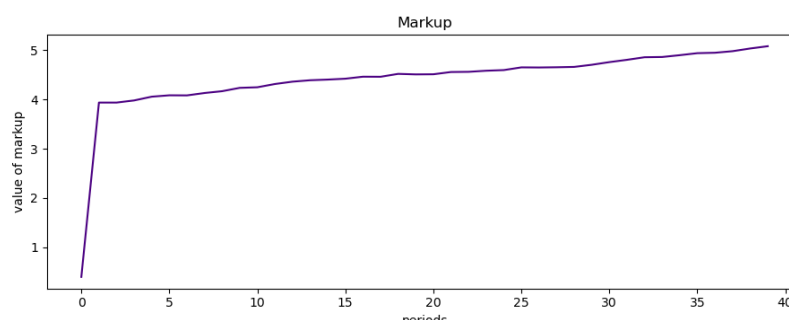


Рисунок 2.7. – График наценки

Наценка повышается, так как объем спроса увеличивается постепенно (Рис 2.8.). Резкий скачок вниз обусловлен резким повышением наценки на товар.

Однако можно увидеть, что даже после резкого снижения, спустя время объем спроса начал увеличиваться, что дает возможность повышать наценку.

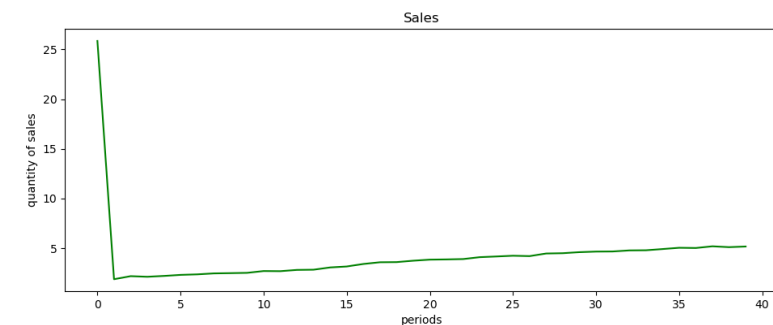


Рисунок 2.8. – График объема спроса

Благодаря благоприятным трендам на предыдущих двух графиках, можно заметить, что линия прибыли идет стремительно вверх, что говорит о том об эффективности системы (Рис. 2.9.). Хотя по первому тестированию и не заметны отличия этой системы от однокритериальной.

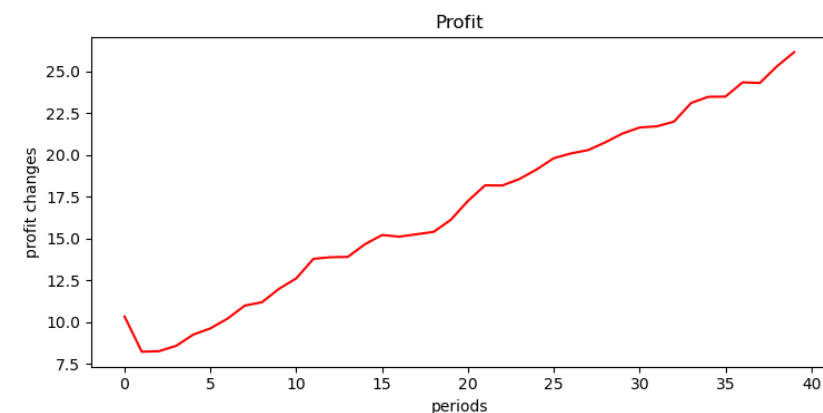


Рисунок 2.9. – График прибыли

Для того чтобы увидеть различие систем, нужно провести сравнительное тестирование и посмотреть, как каждая из них отреагирует и к каким результатам приведет.

Первым отличием многокритериальной системы является не только подсчет наценки при учете нескольких влияющих факторов, но и то, что она

позволяет посчитать рекомендуемую наценку без данных за прошлые периоды.

Рассмотрим, как будет выглядеть график наценки, если пользователь в первый раз получил свой товар и хочет рассчитать наценку на него. Итак, пусть затраты магазина все также 12600, себестоимость товара 400 и цена конкурента, на которую ориентируется пользователь равно 582. Согласно расчетам программы при таких затратах минимальная наценка должна быть 30% для того, чтобы покрыть расходы. Однако рекомендуемая системой наценка – 45%, что обуславливается ценой конкурента. Тогда получаем, что 15% будет чистая прибыль.

Какие будут даваться дальше рекомендации пользователю, если положим, что функция объема спроса равно $y = A/(1+x^{**2})$, где начальное значение

$A = 30$ и $A = A + 6*nr.random.uniform(0, 1)$. График наценки будет примерно таким же, как и при первом тестировании, разница лишь в том, что первое значение тоже было рассчитано системой, а не введено пользователем (Рис 2.10.).

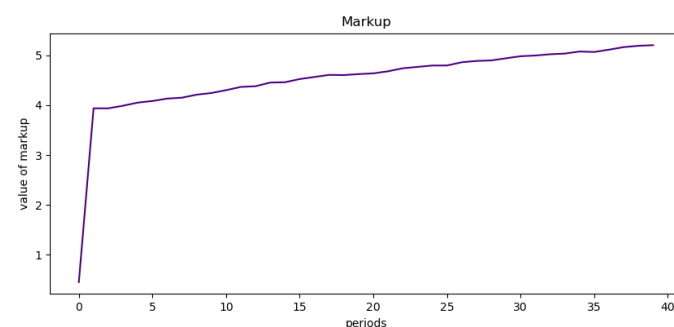


Рисунок 2.10. – График наценки

Далее проведем сравнительный анализ и выясним, насколько действительно многокритериальная система эффективнее.

Если установить функцию $y = A*x^{(-1.3)}$, где y – количество продаж, x – наценка на товар, A – коэффициент функции. $A = A + 6*nr.random.rand()$.

Изначально, пусть, $A = 50$, $x = 0,4$. И пусть расходы предприятия 16600, тогда получим следующее.

В данном случае стоит обратить внимание, на графике системы без учета расходов (график справа) наценка опускается ниже до значения 2, когда же в случае много критериальной системы, минимальное значение составляет 2,88 и ниже этого значения наценка не опускается (Рис 2.11.).

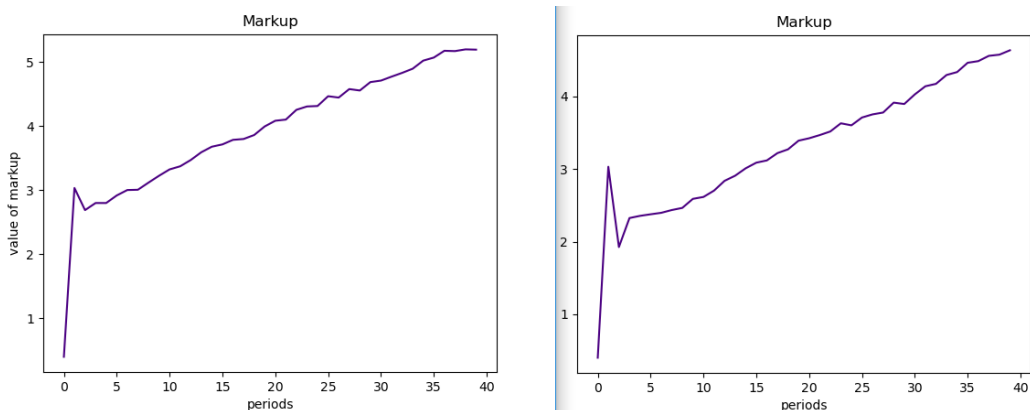


Рисунок 2.11. – Графики формирования наценок разных систем

Такая ситуация обусловлена тем, что была рассчитана минимальная наценка и ниже нее система не позволяет опускать рекомендованную наценку.

Проанализировав эти данные, можно прийти к выводу, что магазин мог бы понести убытки и не покрыть полностью свои расходы. Таким образом, можно видеть, что расходы один из важнейших факторов, который нужно учитывать.

Теперь стоит посмотреть, как система работает с «сезонностью». Здесь (Рис 2.12.) мы можем заметить, что слева график ломаный, и это обусловлено повышением цен в периоды «сезонности» товара. Вначале графики схожи и только потом на том, что слева, можно наблюдать резкий рывок вверх, после того, как закончился период сезонности идет снова спад.

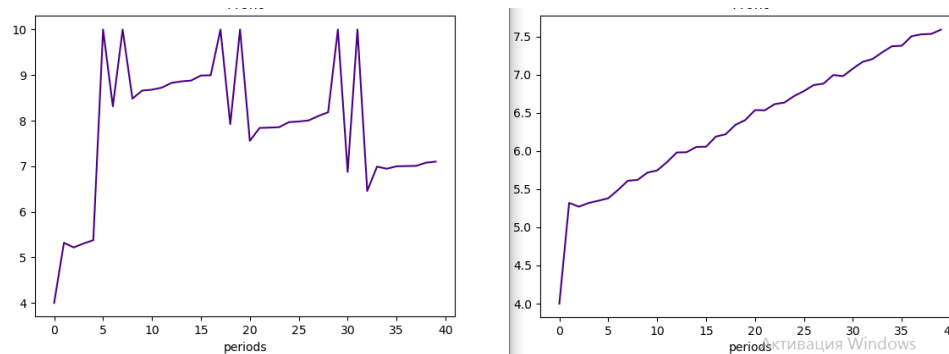


Рисунок 2.12. – График формирования наценки в условиях «сезонности» товара

Кроме того, стоит также рассмотреть различие в прибыли магазина. На этих графиках (Рис. 2.13.) слева снова можно видеть прибыль при использовании системы многокритериальной наценки. Разница очевидна. Не нужно даже делать специальные расчеты, для того, чтобы убедиться, что в первом случае магазин получит в несколько раз больше прибыли в периоды «сезонности». Если внимательно посмотреть, то можно увидеть, что графики схожи, если не обращать внимания на скачки.

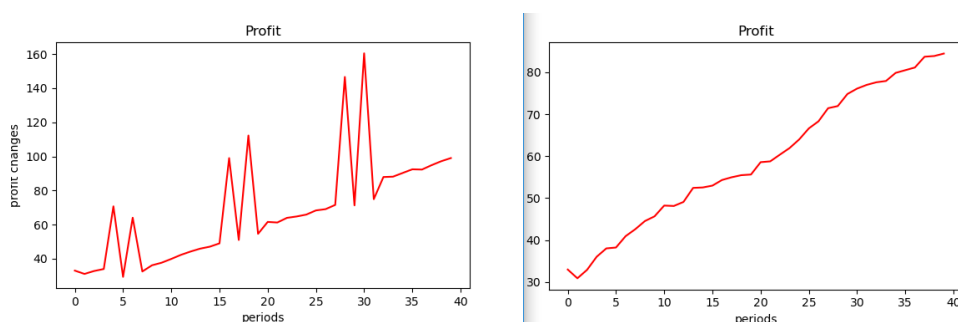


Рисунок 2.13. – Графики прибыли в периоды «сезонности» товара

Таким образом, можно наглядно убедиться в эффективности данной системы. Действительно, с помощью учета нескольких факторов можно не только прекратить работать в убыток, но и повысить прибыльность магазина.

Заключение

Цель работы заключалась в разработке программного продукта корректировки торговой наценки в зависимости от нескольких факторов, влияющих на её величину. Для достижения данной цели были поставлены следующие задачи:

1. Рассмотреть понятие и теоретические аспекты торговой наценки.
2. Изучить особенности онлайн торговли.
3. Выявить факторы, влияющие на изменение торговой наценки.
4. Разработать программу автоматизированной корректировки наценок.

В ходе работы были рассмотрены понятия цены и интернет торговли. При рассмотрении понятия цены и ее характеристик было выяснено, что до формирования итоговой наценки у розничного продавца, существует еще как минимум несколько аспектов, которые учитываются и входят в конечную стоимость продукции: себестоимость продукции, прибыль предприятия, акциз, НДС, оптовая надбавка и только потом розничная наценка. При формировании розничной наценки стоит учитывать несколько основных факторов: себестоимость продукции, цену конкурентов, а также уникальность товара.

Если учесть все эти факторы и товар будет действительно уникальным, то организация может рассчитывать на достаточно большую прибыль, ввиду низкой конкуренции. Если же не придать особого значения первому или второму фактору, тогда это в свою очередь может привести либо к снижению прибыльности магазина, либо же к банкротству. Поэтому так важно грамотно рассчитывать наценку на товар, если магазин хочет существовать и развиваться на рынке.

Также были рассмотрены различные системы скидок и надбавок. Грамотное использование этого инструмента гарантирует продавцу прибыльность. Система скидок особенно хорошо помогает стимулировать покупателя

приобретать продукцию. Таким образом, можно привлечь новых клиентов, а потом нужно удерживать своих клиентов путем правильного назначения наценок.

Кроме того, при формировании наценок стоит учитывать и особенности условий рынка. Так, например, онлайн торговля также имеет свои особенности, которые были рассмотрены в ходе работы.

Интернет торговля становится особенно актуальной и очень возможно, что в ближайшем будущем онлайн магазины смогут потеснить обычные торговые центры. Можно также сказать, что автоматизированная система наценки была бы особенно актуальной при торговле во всемирной паутине.

В рамках разработки данной системы был проведен анализ существующих программ, которые позволяют делать наценку, а также критериев, которые необходимо учитывать при разработке системы, если речь идет об онлайн торговле.

На сегодняшний день существует множество программ, которые позволяют делать наценку на товар и получать отчетность по продажам. Однако человек сам все еще должен высчитывать наценку исходя из тех знаний, которыми он обладает. Для того чтобы автоматизировать этот процесс, было проведено исследование факторов, влияющих на величину наценки.

Были выделены основные критерии, которые необходимо учесть: расходы магазина, цена конкурента на данный товар, а также «сезонность» товара. Более того, элемент прогнозирования наценки также является одним из важных аспектов системы, так как в случаях, когда перестройка наценки зависит не столько от расходов, сколько от динамики изменения спроса, этот функционал позволяет максимизировать прибыль. Именно в данных случаях особенно важно быстро реагировать на изменения и корректировать наценку своевременно.

Используя данные о тратах организации, можно посчитать минимальную наценку. Расчёты по ценам конкурента дают возможность узнать верхний предел наценки. «Сезонность» товара – это фактор, при учете которого магазины могут увеличивать свою прибыль в несколько раз в определенные периоды времени.

При разработке алгоритма и написании кода будущей системы были учтены две возможные ситуации: данные о наценке и количестве продаж за прошлые периоды существуют, и наценка на товар делается впервые.

Система может обрабатывать обе ситуации успешно. При разработке функции учета «сезонности» было выяснено, что все-таки остается достаточно трудно решить, какую наценку следует сделать в такой ситуации. Поэтому было решено, что функция возвращает минимально возможную, а далее, если спрос будет расти при такой большой наценке, то система скорректирует наценку с помощью элементов прогнозирования.

Согласно проведенному тестированию, а также построению моделей, можно сказать, что система работает успешно. Более того, благодаря учету нескольких критериев она позволяет избегать убыточности, максимизировать прибыль, а также точнее подбирать наценку, используя данные о ценах конкурента.

Все выше перечисленные задачи были выполнены в ходе работы, и была разработана система наценки. Данный вариант системы все еще может быть улучшен путем более сложного алгоритма учета «сезонности», например, а также внедрением автоматического контроля и сравнения цен конкурентов. Подводя итог, можно сказать, что благодаря последовательному анализу и построению алгоритмов цель работы удалось выполнить - разработать систему корректировки наценки.

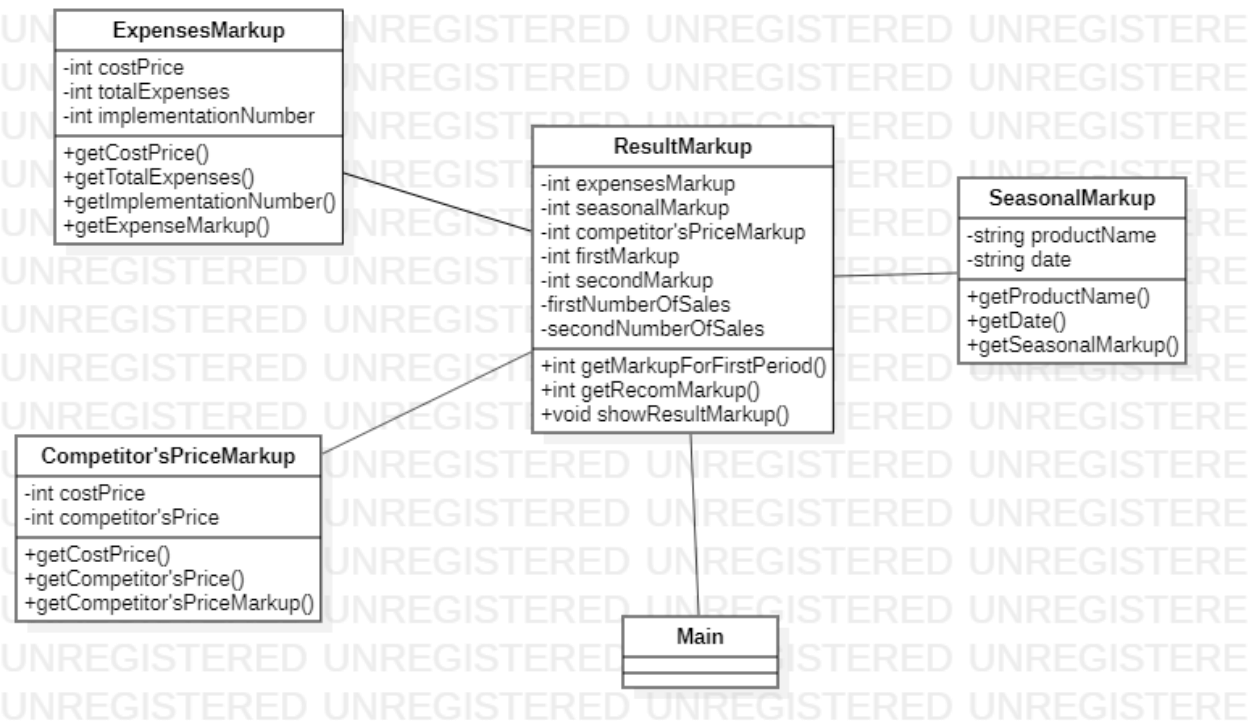
Список литературы

1. Анализ развития интернет-торговли на российском рынке, А.В. Коняшова, В.А. Яшагина, 2018
2. Инновационная экономика: перспективы развития и совершенствования, Гамидова, Н.Г. Cost-volume-profit-анализ в системе управления / Н.Г. Гамидова. 2015
3. Основы ценообразования, А.Н. Крестьянинов, Ю.Н. Жулькова, А.А. Сазонов, 2016
4. Основные тренды развития российской розничной торговли, Красюк И.Н., Интернет-журнал Науковедение, 2014
5. Предиктивный алгоритм назначения наценки, Крылов В.В., Крылов С.В., 2019
6. Развитие систем управленческого учета в торговой деятельности и их особенности в сфере розничной торговли, Зуева И.А., Гребеник В.В., Жидков А.С., Иванова Е.В., 2017
7. Рынок интернет-торговли, Седых И.А, 2016
8. Competition-Based Dynamic Pricing in Online Retailing: A Methodology Validated with Field Experiments, Marshall Fisher , Santiago Gallino , Jun Li, 2017
9. Consumer price search and platform design in internet commerce, Dinerstein, M., Einav, L., Levin, J., Sundaresan, N., 2018
10. Design intelligence of web application for Internet direct consumer-to-consumer trading, Chan, V., Wu, T.H., Pang, G., 2015
11. Georgia Koutrika Modern Recommender Systems: from Computing Matrices to Thinking with Neurons Proceeding of the 2018 International Conference on Management of Data Pages 1651-1654 Houston, TX, USA — June 10 - 15, 2018
12. Internet regulation, two-sided pricing, and sponsored data, Bruno Julliena, Wilfried Sand-Zantman, 2018
13. Margin squeeze in the Internet backbone interconnection market: a case study of Korea, Choi, S.M., Libaque-Saenz, C.F., Lee, S.-W., Park, M.-C., 2015
14. Marketing and pricing, Valery Bondarenko, 2017
15. Past, present and future of Industry 4.0 - a systematic literature review and research agenda proposal, Liao, Y. aEmail Author, Deschamps, F. a, b, Lour es, E. F. R. a, c, Ramos, L. F. P. a, 2017
16. Predicting consumer product demands via Big Data: the roles of online promotional marketing and online reviews, Chong, A.Y.L., Ch'ng, E., Liu, M.J., Li, B., 2017
17. Pricing in the online invoice trading market: First empirical evidence, Gregor Dorfleitner, Jacqueline Rad, Martina Weber, 2017

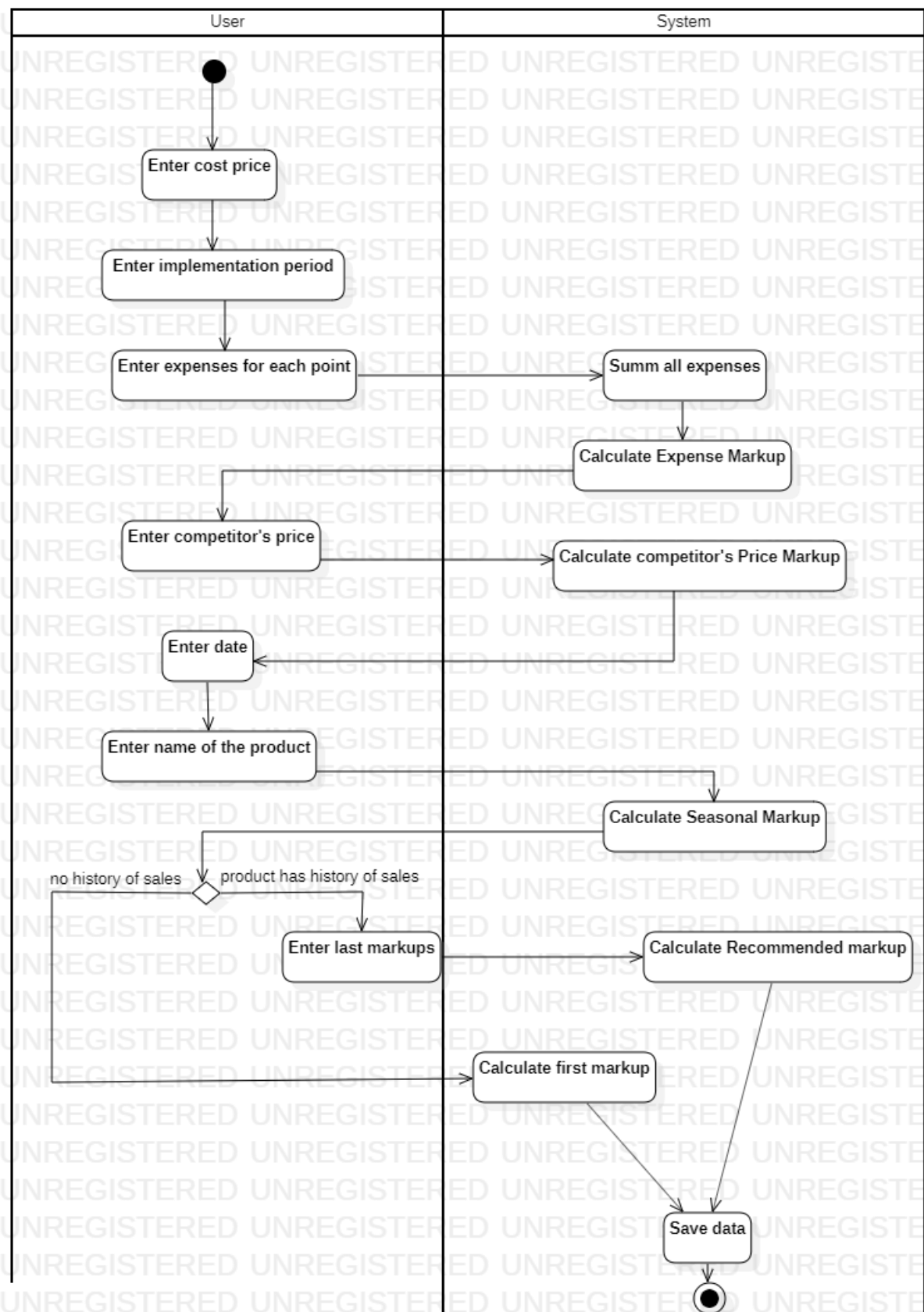
18. Pricing Strategy, Stephen J. Skripak, Richard Parsons, Anastasia Cortes, Anita Walz, 2016
19. The Elements of Statistical Learning. Springer, Hastie T., Tibshirani R., Friedman J. r, 2015
20. Trading financial indices with reinforcement learning agents, Pendharkar, P.C., Cusatis, P., 2018

Приложения.

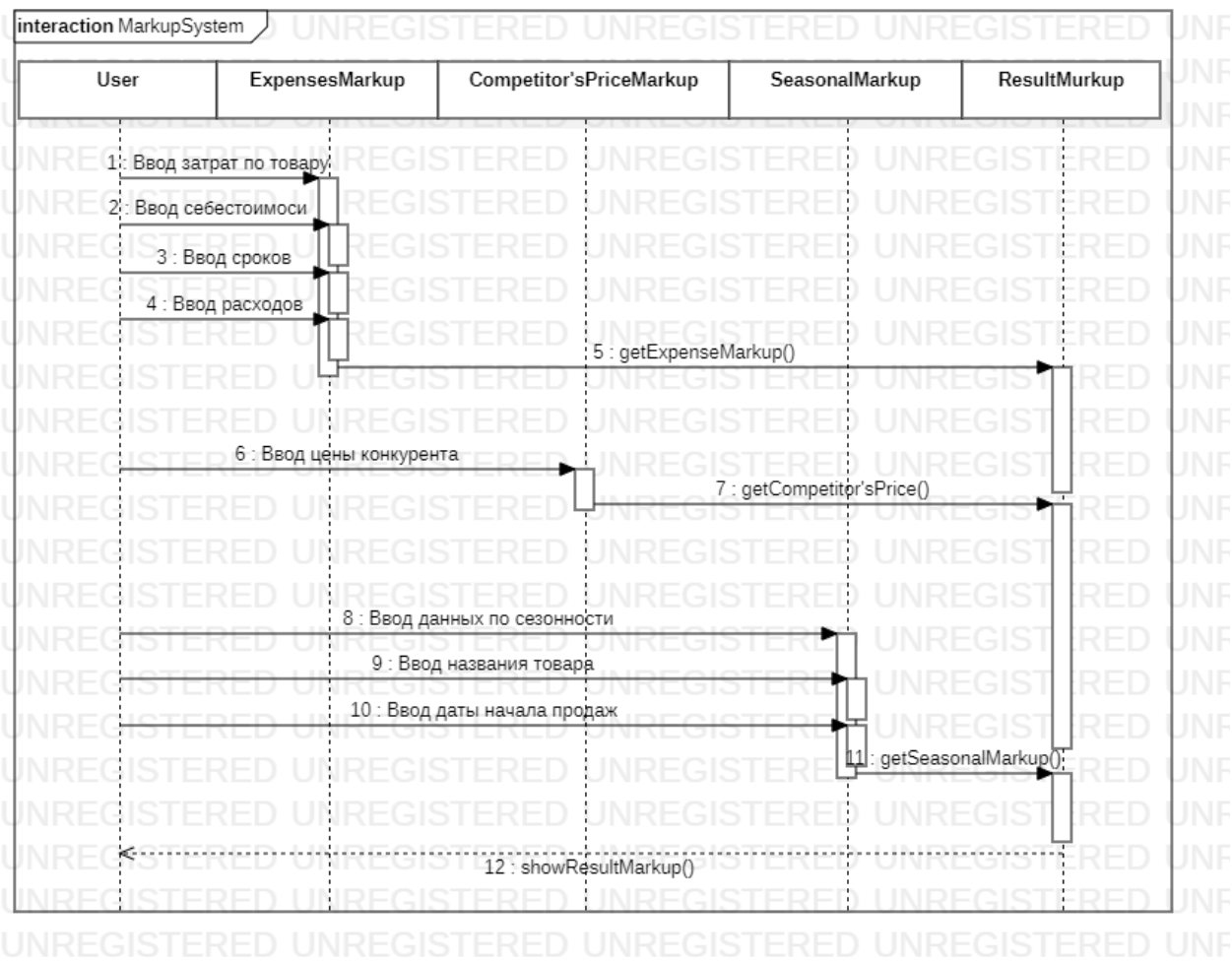
Приложение 1. Диаграмма классов.



Приложение 2. Диаграмма активности.



Приложение 3. Диаграмма последовательности.



Приложение 4.Диаграмма состояний системы.



Приложение 5. Многокритериальная система автоматизированной корректировки наценки.

```
import numpy as np

class CompetitorsPriceMarkup:

    def __init__(self, cost_price, competitor_price):
        self.cost_price = cost_price
        self.comp_price = competitor_price

    def get_cost(self):
        return self.cost_price

    def get_competitor_price(self):
        return self.comp_price

    def get_competitor_price_markup(self):
        rec_markup = (self.comp_price - self.cost_price) / self.cost_price * 100
        return rec_markup

class ExpenseMarkup:

    def __init__(self, cost_price, total_expense, implementation_num):
        self.cost_price = cost_price
        self.total_expense = total_expense
        self.implementation_num = implementation_num

    def get_cost(self):
        return self.cost_price

    def get_cost(self):
        return self.cost_price

    def get_expenses(self):
        return self.total_expense

    def get_implementation_period(self):
        return self.implementation_num

    def get_expense_markup(self):
        rec_markup = self.total_expense / (self.implementation_num * self.cost_price) * 100
        return rec_markup

class SeasonalMarkup:

    def __init__(self, name, date):
        self.name = name
        self.date = date

    def get_name(self):
        return self.name

    def get_date(self):
        return self.date

    def get_seasonal_markup(self):
        wb = openpyxl.load_workbook('./Seasonal.xlsx')
```

Продолжение приложения 5.

```
def get_seasonal_markup(self):
    wb = openpyxl.load_workbook('./Seasonal.xlsx')
    sheet = wb.active

    number_date = int(self.date[0:2].strip())
    month_date = self.date[2:len(self.date)].strip()
    rec_markup = 0

    if month_date == "ноября" or month_date == "декабря" or month_date == "января":
        for row in sheet.rows:
            if row[0].value == self.name:
                rec_markup = 100

    if number_date > 6 or number_date < 16 and month_date == "февраля":
        for row in sheet.rows:
            if row[1].value == self.name:
                rec_markup = 100

    if number_date > 19 or number_date < 26 and month_date == "февраля":
        for row in sheet.rows:
            if row[2].value == self.name:
                rec_markup = 100

    if number_date >= 1 or number_date < 11 and month_date == "марта":
        for row in sheet.rows:
            if row[3].value == self.name:
                rec_markup = 100

    if month_date == "апреля":
        for row in sheet.rows:
            if row[4].value == self.name:
                rec_markup = 100

    if number_date >= 28 and month_date == "апреля" or number_date < 8 and month_date == "мая":
        for row in sheet.rows:
            if row[5].value == self.name:
                rec_markup = 100

    if number_date > 6 or number_date < 10 and month_date == "мая":
        for row in sheet.rows:
            if row[6].value == self.name:
                rec_markup = 100

    if month_date == "мая" or month_date == "июня" or month_date == "июля" or month_date == "августа":
        for row in sheet.rows:
            if row[7].value == self.name:
                rec_markup = 100

    if number_date >= 15 and month_date == "августа" or number_date <= 15 and month_date == "сентября":
        for row in sheet.rows:
            if row[6].value == self.name:
                rec_markup = 100

    return rec_markup
```

Продолжение приложения 5.

```
class ResultMarkup:

    def __init__(self, last_markup, pre_last_markup, last_num, pre_last_num, seasonal_markup, exp_markup, comp_markup):
        self.last_markup = last_markup
        self.pre_last_markup = pre_last_markup
        self.last_num = last_num
        self.pre_last_num = pre_last_num
        self.seasonal_markup = seasonal_markup
        self.exp_markup = exp_markup
        self.comp_markup = comp_markup

    def get_rec_markup(self, gamma):
        rec_markup = self.last_markup + gamma * (self.last_markup * self.last_num - self.pre_last_markup * self.pre_last_num)

        if rec_markup < self.exp_markup:
            rec_markup = self.exp_markup
        if self.seasonal_markup > rec_markup and self.last_num >= self.pre_last_num:
            rec_markup = self.seasonal_markup

        return rec_markup

    def get_first_markup(self):
        recom_markup = self.comp_markup
        if self.exp_markup > recom_markup:
            recom_markup = self.exp_markup
        elif self.seasonal_markup > recom_markup:
            recom_markup = self.seasonal_markup

print("Сколько штук товара предположительно будет продано за месяц?")
quantity = int(input())
print("Введите стоимость товара за штуку")
cost = int(input())
print("Введите ваши расходы по пунктам: ")
print("Хостинг")
hosting = int(input())
print("Доступ в Интернет: ")
internet = int(input())
print("Бухгалтерское сопровождение")
accountant = int(input())
print("Затраты на рекламу")
advertisement = int(input())
print("Администрирование сайта")
site_administration = int(input())
print("Введите доп. расходы, если такие есть или же просто 0 ")
dop_expenses = int(input())
total_expenses = hosting + internet + accountant + advertisement + site_administration + dop_expenses

exp_markup = ExpenseMarkup(cost, total_expenses, quantity).get_expense_markup()

print("Введите цену конкурента, на которую вы ориентируетесь")
co_price = int(input())

comp_mark = CompetitorsPriceMarkup(cost, co_price).get_competitor_price_markup()
```


Продолжение приложения 5.

```
print("Введите дату начала продажи (формат ввода - 15 марта) : ")
data = str(input())
print("Введите название товара: ")
product_Name = str(input())

season_mark = SeasonalMarkup(product_Name, data).get_seasonal_markup()

print("Вы уже продавали этот товар?")
answer = str(input())

amount = 40
mark_array = np.zeros(amount) # массив наценок
num_array = np.zeros(amount) # массив количества продаж
A = np.zeros(amount) # коэффициент функции спроса
date_month = ["января", "февраля", "марта", "апреля", "мая", "июня", "июля", "августа", "сентября", "октября", "ноября",
              "декабря"]

if answer == "нет":
    rec_mark = ResultMarkup(0, 0, 0, 0, season_mark, exp_markup, comp_mark).get_first_markup()
    print("Рекомендуемая наценка на первый период времени ", rec_mark)
    print("Расчет последующих наценок из имеющихся данных ")

    mark_array[0] = 0
    mark_array[1] = rec_mark
    A[0] = 50
    A[1] = A[0] + 6 * np.random.uniform(0, 1)

    mark_array[0] = 0
    mark_array[1] = rec_mark
    A[0] = 50
    A[1] = A[0] + 6 * np.random.uniform(0, 1)
    num_array[0] = 0
    num_array[1] = A[1] * mark_array[1] ** (-1.3)
    indicator = 0
    month = data[2:len(data)].strip()
    number_of_month = int(data[0:2].strip())

    for i in range(len(date_month) - 1):
        if date_month[i] == month:
            indicator = i
            month = date_month[i]
            break

    for i in range(2, len(mark_array)):
        c_date = str(number_of_month) + " " + month
        season_mark = SeasonalMarkup(product_Name, c_date).get_seasonal_markup()

        mark_array[i] = ResultMarkup(mark_array[i - 1], mark_array[i - 2], num_array[i - 1], num_array[i - 2],
                                      season_mark, exp_markup, comp_mark).get_rec_markup(0.04)
        A[i] = A[i - 1] + 6 * np.random.uniform(0, 1)
        num_array[i] = A[i] * mark_array[i] ** (-1.3)

    if indicator + 1 == len(date_month):
        indicator = -1
```

Продолжение приложения 5.

```
        indicator = indicator + 1
        month = date_month[indicator]

    print(mark_array)

else:
    print("Введите последнюю наценку, которая была на данный товар в % (например 40) ")
    markup = int(input())
    print("Введите предыдущую наценку в % (например 40) ")
    pre_markup = int(input())

    mark_array[0] = pre_markup
    mark_array[1] = markup
    A[0] = 50
    A[1] = A[0] + 6 * np.random.uniform(0, 1)
    num_array[0] = A[0] * mark_array[0]**(-1.3)
    num_array[1] = A[1] * mark_array[1]**(-1.3)

    indicator = 0
    month = data[2:len(data)].strip()
    number_of_month = int(data[0:2].strip())

    for i in range(len(date_month) - 1):
        if date_month[i] == month:
            indicator = i
            month = date_month[i]
            break
```

Акти

```
indicator = 0
month = data[2:len(data)].strip()
number_of_month = int(data[0:2].strip())

for i in range(len(date_month) - 1):
    if date_month[i] == month:
        indicator = i
        month = date_month[i]
        break

for i in range(2, len(mark_array)):
    c_date = str(number_of_month) + " " + month
    season_mark = SeasonalMarkup(product_Name, c_date).get_seasonal_markup()

    mark_array[i] = ResultMarkup(mark_array[i-1], mark_array[i-2], num_array[i-1], num_array[i-2], season_mark, exp_markup, com
    A[i] = A[i-1] + 6 * np.random.uniform(0, 1)
    num_array[i] = A[i] * mark_array[i]**(-1.3)

    if indicator + 1 == len(date_month):
        indicator = -1

    indicator = indicator + 1
    month = date_month[indicator]

print(mark_array)
```