

Федеральное государственное автономное образовательное
учреждение высшего образования

Национальный исследовательский университет
«Высшая школа экономики»

Факультет компьютерных наук
Основная образовательная программа
Прикладная математика и информатика

КУРСОВАЯ РАБОТА

Программный проект на тему

**«Оптимизация запасов в ритейл сети с помощью
алгоритмов обучения с подкреплением»**

Выполнил студент группы 184, 3 курса,
Ширнин Александр Андреевич

Руководитель КР:
Старший преподаватель, Романенко Алексей Александрович,
базовая кафедра компании SAS

Москва 2021

Содержание

Содержание	1
Аннотация	2
1 Введение	4
2 Цели и задачи проекта	5
3 Планируемые результаты проекта	5
4 Обзор литературы и существующих решений	6
5 Методы восстановления спроса	8
6 Сравнение результатов алгоритмов	18
7 Заключение	19
Список литературы	20

Аннотация

Одной из проблем во время создания различных моделей для управления запасами в цепочки поставок для ритейл сетей является отсутствие знания истинного спроса на товары в течение различного периода времени, потому что спрос требуется для расчёта различных метрик моделей. Истинный спрос известен лишь в те дни, когда количество предложенного товара было больше количества проданного, в остальные дни нельзя судить о количестве спроса по зафиксированным продажам в магазине, будем называть их дефицитными. Требуется как можно точнее смоделировать спрос в дефицитные дни, чтобы обучить агента для модели обучения с подкреплением (RL). От среды (environment) требуется в качестве одного из составляющих ответа возвращать объём спроса, зафиксированный в конкретный день, для дней с неизвестным спросом или в дни, в которые товар отсутствовал или было продано столько единиц товара, сколько было в наличии, эта величина как раз требует моделирования. Более того, для обучения различных моделей нужно учитывать нестационарный характер спроса для более точного моделирования реального поведения покупателей в жизни. Перспективными моделями для моделирования спроса являются модели машинного обучения и методы с использованием расчета скользящего среднего. За основу проекта взята последняя версия фреймворка, разработанная нашими коллегами для построения алгоритма оптимизации политик пополнения запасов ритейл сети, где отклик среды моделируется на основе реальных данных за 3 года в розничной сети. На основе их работы, в проекте будет развита часть новой системы для обучения RL агента. Отличием этого варианта системы от известных аналогов является моделирование спроса с учетом наличия нестационарной компоненты, считается, что спрос по товару в магазине описывается стохастическим процессом.

One of the problems during the creation of various models for inventory management in the supply chain for retail chains is the lack of knowledge of the true demand for goods for different periods of time, because demand is required to calculate different metrics of the models. The true demand is known only on those days when the quantity of the offered product was more than the quantity sold; on other days it is impossible to judge the quantity of demand by the fixed sales in the store, we will call them scarce. It is required to model the demand

on scarcity days as accurately as possible in order to train the agent for a reinforcement learning (RL) model. Environment is required as one of the components of the answer to return the volume of demand recorded on a specific day, for days with unknown demand or on days when the product was absent or as many units of the product were sold as were in stock, this value requires modeling. Moreover, to train various models, it is necessary to take into account the non-stationary nature of demand for more accurate modeling of the real behavior of buyers in life. Promising models for demand modeling are machine learning models and methods using moving average calculation. The project is based on the latest version of the framework developed by our colleagues to build an algorithm for optimizing the replenishment policies of a retail network, where the response of the environment is modeled on the basis of real data for 3 years in the retail network. Based on their work, a part of the new system for training the RL agent will be developed in the project. The difference between this version of the system and analogs in the world is the modeling of demand, taking into account the presence of a non-stationary component, as it is believed that the demand for a product in a store is described by a stochastic process.

Ключевые слова

Машинное обучение, градиентный бустинг, метод скользящего среднего, генерация признаков, распределение Пуассона, обучение с подкреплением, восстановление спроса.

1 Введение

Одной из ключевых проблем для ритейл сети является оптимальное пополнение магазинов запасами из центрального распределительного склада (ЦРС). В бизнес компаниях для решения данной задачи обычно требуется большое количество людей, так бизнес процесс содержит несколько этапов:

- 1) прогнозирование спроса в ритейл сети;
- 2) планирование спроса;
- 3) расчет перемещений со склада в магазины;
- 4) формирование заказов для пополнения склада (от поставщиков).

Это классический метод решения задачи, в течение этого процесса может применяться множество различных алгоритмов, но это усложняет понимание и проверку того, являются ли финальные рекомендации о размерах пополнений оптимальными.

В задаче оптимизации политик пополнения в научных публикациях широко исследуются методы обучения с подкреплением. В данных моделях используется среда и агент, где среда и её отклик проектируются на основе данных о продажах, перемещениях, запасах. Для обучения агента также используется истинный спрос на товары, он нужен для расчета метрики reward, которую использует агент для оптимизации запасов. На практике истинный спрос можно узнать из данных о продажах, например, чековых данных, в те дни, когда продажах не превышали остаток товара в магазине. А в те дни, когда случился дефицит, то есть остаток товара в магазине равен продажам за день, истинный спрос неизвестен. Поэтому в дефицитные дни, а также для тех дней, когда нет

данных о продажах, например, из-за ошибки системы учета, требуется смоделировать спрос. Задача моделирования спроса заключается в определении наиболее точной модели, описывающей величину спроса по товару магазину на историческом периоде данных. Восстановление значений спроса в дефицитные дни будем называть задачей восстановления спроса. В данном проекте будут рассмотрены и реализованы различные методы восстановления спроса за требуемый период времени.

2 Цели и задачи проекта

Требуется реализовать восстановление спроса различными методами для предложенных данных, чтобы эти результаты могли использоваться в среде модели обучения с подкреплением. а была разработана простая версия восстановления константного спроса, поэтому основной задачей является реализация перехода к нестационарности, а также разработка улучшенных способов восстановления спроса.

3 Планируемые результаты проекта

Основными целями проекта являются:

1. Развитие метода восстановления спроса с учётом нестационарности процесса, описывающего модель спроса.
2. Реализация новых методов восстановления спроса.
3. Сравнение методов восстановления спроса с помощью различных метрик на реальных данных.

4. Создать единый интерфейс, с которым будет работать фреймворк gum и среда модели, реализованная в нём.

4 Обзор литературы и существующих решений

Для проекта были рассмотрены решения похожих задач и проанализированы различные статьи, чтобы понять, можно ли сразу применить готовое решение или использовать аналогичные методы. Применить готовое решение оказалось невозможно, так как у наших данных своя структура и признаки, под которые не подходят реализованные алгоритмы, более того, нужно, чтобы код работал с фреймворком gum [1]. Однако можно найти некие алгоритмы, взять идеи из источников и уже их имплементировать. Также основная проблема заключается в том, что многие решают задачу предсказаний будущего спроса по известному, в то время как нам требуется восстановить большие промежутки пропущенных данных, которые могут быть и в самом начале временного ряда.

Вначале было рассмотрено решение, выложенное пользователем на сайте kaggle.com [2]. В данном решении были предложены различные варианты усреднения последних известных величин (концов интервала), в котором восстанавливаются значения, также предлагалось использовать константное значение, например, моду или медиану всех значений временного ряда. Такой вариант решения не подходил для моего проекта из-за плохой точности восстановленных значений, а также в данном случае отсутствует переход к нестационарности. Было также рассмотрен вариант решения практически такой же задачей, где использовалась модель обучения с подкреплением с фреймворком gum, для которой восстанавливался спрос [3]. Однако автор использовал функцию

синусоиды с предположением, что среднее и разброс спроса не меняются. Этот вариант тоже не подходит, так как нам требуется более высокая точность предсказаний, более того, в нашем случае за 3 года среднее или разброс спроса могли измениться.

Далее было рассмотрено решение соревнования M5 forecasting на сайте kaggle, которое заняло 1 место [4], это решение получило высокую точность предсказаний. В соревновании требовалось предсказывать спрос для будущего, но те же методики можно использовать и в нашем случае, так как, во-первых, мы тоже в некоторых ситуациях прогнозируем будущий спрос, во-вторых, эти алгоритмы также работают на данных с пропусками. Автор решения предлагает использовать генерацию признаков, а именно он предлагает использовать методы скользящего среднего для создания новых признаков, то есть, для каждого дня считать средний спрос на товар в магазине в ближайшем прошлом и будущем периоде времени. Также автор использует модель градиентного бустинга, так как данные модели машинного обучения обычно самые эффективные для работы с табличными данными, эти модели используют ансамблирование простых моделей, которые обучаются на производных ошибок прошлых моделей. Поэтому было решено, что в нашем проекте будет реализована такая модель с данными признаками, которые также называют “lag features”. В качестве библиотеки для градиентного бустинга будем использовать XGBoost [5], так как авторы библиотеки используют вторые производные в своей имплементации алгоритма, что улучшает качество предсказаний. Перед реализацией основной модели будут также реализованы более простые модели для сравнения качества: подсчёт среднего, а также скользящее среднее, но без градиентного бустинга.

5 Методы восстановления спроса

5.1 Описание данных для работы

В качестве данных для восстановления спроса использовались различные данные за три года о продажах товаров в различных магазинах. Датафреймы обладали следующей структурой:

- 1) `product_id` - уникальный `id` каждого товара.
- 2) `store_id` - уникальный `id` каждого магазина в ритейл-сети.
- 3) `flg_spromo` - бинарный признак, означающий наличие определённой акции или скидки на товар в указанную дату.
- 4) `s_qty` - количество проданного товара за день.
- 5) `stock` - количество товара, которое было на полках на начало дня.
- 6) `curr_date` - дата дня, за которую были записаны данные.

Предложенные данные иногда содержат аномалии и пропуски, поэтому необходима предобработка перед их использованием. В качестве аномалий было иногда несоответствие между продажами и количеством товара на начала дня, были ситуации, когда количество товара на начало дня было отрицательными. Такие даты было решено считать некорректными и не использовать в качестве данных для обучения моделей, которые использовались для восстановления спроса. Если же модели требуется использовать все дни (например, `moving average`), то считаем, что в данный день продажи были нулевыми. Также одной из особенностью данных был частый пропуск данных за целый день, в этом случае требовалось переформировать датафрейм, сделав индексом дату, добавить недостающие даты в датафрейм. Пропуски в этом случае решено заполнять методом `ffill`, будем считать, что данные не менялись за пропущенные дни.

5.2 Описание алгоритма восстановления спроса

Для перехода к нестационарности, чтобы предсказанный спрос не был одинаковым за день, что требуется для модели обучения с подкреплением, будем использовать распределение Пуассона для моделирования спроса. Это распределение подходит для нашей задачи, так как имеет параметр лямбда, которая для нашей задачи является интенсивностью спроса, которую мы и будем рассчитывать. Более того, в аналогичных работах с моделями обучения с подкреплением для ритейл-сетей также обычно используется данное распределение. Таким образом, вначале выбирается метод расчёта интенсивности спроса, т.е. того самого параметра лямбды, а затем с помощью Пуассоновской функции сэмплирования мы получаем значение спроса за каждый день.

Будем использовать следующую модель, которая основана на Пуассоновском распределении:

$$P_m(x = k) = \begin{cases} 1, k = m = 0 \\ 0, k > m \text{ или } k < 0 \\ \frac{(\lambda)^k}{k!} e^{-\lambda}, 0 \leq k < m \\ 1 - \sum_{l=0}^{m-1} \frac{(\lambda)^l}{l!} e^{-\lambda}, k = m > 0 \end{cases}$$

Иллюстрация 1 - формулы для модели

В данном случае, $P_m(k)$ - вероятность продать k товаров за день, если на начало дня было, m товаров в магазине где m - кол-во товаров в магазине на начало дня, а k - кол-во проданного товара за день. Параметр лямбда находится с помощью метода максимального правдоподобия:

$$\lambda = \frac{\sum_{i=1}^T x_i [i - \text{недефицитный}]}{\sum_{i=1}^T [i - \text{недефицитный}]}, \text{ где } T - \text{номер}$$

недефицитного дня, x_i - количество проданного товара за i -ый день.

Здесь рассчитывается количество проданного товара, делённый на количество дней, когда не было дефицита в магазине, то есть в наличии было товара больше или столько же, сколько продали. То есть считаются средние продажи по все недефицитные дни.

В этой работе, чтобы добавить элемент нестационарности мы позволяем интенсивности меняться со временем. Таким образом мы реализовываем подсчёт лямбды отдельно для каждой пары “товар - магазин” для различных наборов дней, которые имеют разное распределение спроса.

$$\lambda(t) = \frac{\sum_{i=t-w}^{t+w} x_i [i - \text{недефицитный}]}{\sum_{i=t-w}^{t+w} [i - \text{недефицитный}]},$$

где t - номер дня, для которого рассчитывается интенсивность спроса, w - размер окна, то есть количество дней, которые учитываются для подсчёта интенсивности спроса, x_i - количество проданного товара за i -ый день.

Отдельно также был реализован подсчёт интенсивности спроса с помощью градиентного бустинга, где лямбда рассчитывается следующим способом:

$\lambda(t) = XGB.predict(X)$, где X - вектор признаков для набора объектов, то есть дней, для которых нужно предсказать интенсивность спроса.

Для более точного восстановления спроса требуется уйти от константного спроса (использовался в прошлой версии работы). Поэтому

нужно отдельно сэмплировать спрос для разных участков временного ряда, где одинаковая интенсивность спроса. Это также нужно для более удачного обучения модели обучения с подкреплением. Для сэмплирования спроса с учётом интенсивности спроса используем следующий код:

```
min(max_sales, max(stats.poisson(lambda), sales_oracle_day)  
), где max_sales - максимальное количество продаж за весь временной ряд по данному товару, sales_oracle_day - количество продаж товара в день, в который восстанавливается спрос.
```

Таким образом, мы добиваемся того, что спрос за день не может быть меньше реальных продаж. Но при этом мы ограничиваем спрос, который выдаётся функцией сэмплирования, он не будет больше максимального количества продаж за всю историю продаж. Важно заметить, что всё это считается отдельно для каждой пары “уникальный товар” - ”уникальный магазин”.

5.3 Базовый метод восстановления спроса

В начале моей работы было замечено, что метод из прошлой версии работы не работал верно, а именно не верно был устроен подсчёт лямбды, поэтому при константной лямбде на каждой итерации получался константный спрос для каждого дня. Это было неверное поведение системы, так как мы желали получить сэмплированные величины с помощью распределения Пуассона для дней, в которые не известен истинный спрос. Ошибка была в реализации алгоритма. Поэтому вначале требовалось написать реализацию базового способа без ошибки.

В качестве базового способа было решено разделить отдельно для каждой пары “уникальный товар” - “уникальный магазин” дни с промо акцией и на дни без него, для каждого такого набора дней отдельно

посчитаем средний спрос по ним. Таким образом был совершён переход к нестационарности спроса, временной ряд был разделён на участки дней, имеющих разное распределение спроса.

В качестве дополнительного параметра функции в данном методе был реализован статистический тест t-критерий Стьюдента. После того как посчитана средняя интенсивность спроса для этих наборов дней, проведём тестирование гипотезы, что средняя интенсивность отличается значительно, то есть имеет смысл использовать различную интенсивность для промо и не промо дней.

На практике оказалось, что, несмотря на данные теста, лучше использовать разные лямбды для промо и не промо дней, таким образом происходит более точное восстановление спроса.

5.4 Интерактивные графики для визуализации результатов

Для оценки качества результатов вначале было решено создать интерактивные графики, на которых были показаны следующие величины:

- 1) Сумма реальных продаж по всем товарам за день
- 2) Суммарный сэмплированный спрос только по товарам, по которым за день не было дефицита
- 3) Суммарный сэмплированный спрос, включая те товары, по которым был дефицит за день
- 4) Суммарное количество дефицитных товаров за день
- 5) Суммарное количество товаров с промо акциями за день

Ниже приведён пример такого графика:

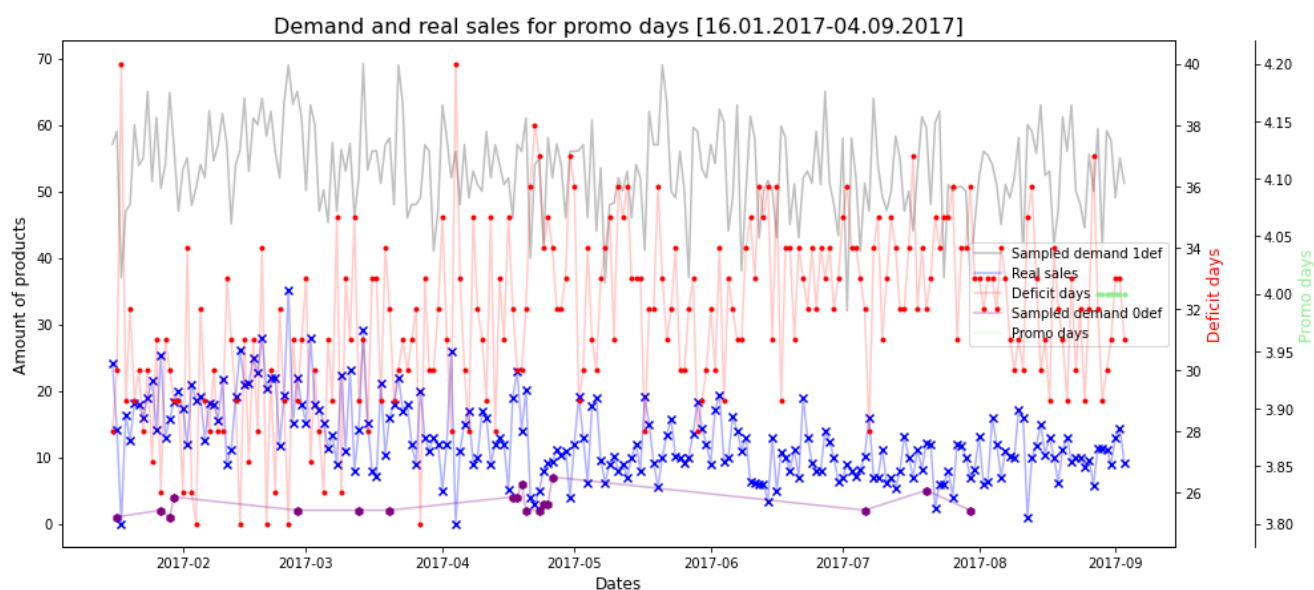


Иллюстрация 2 - пример интерактивного графика для спроса

Ещё один пример графика продаж:

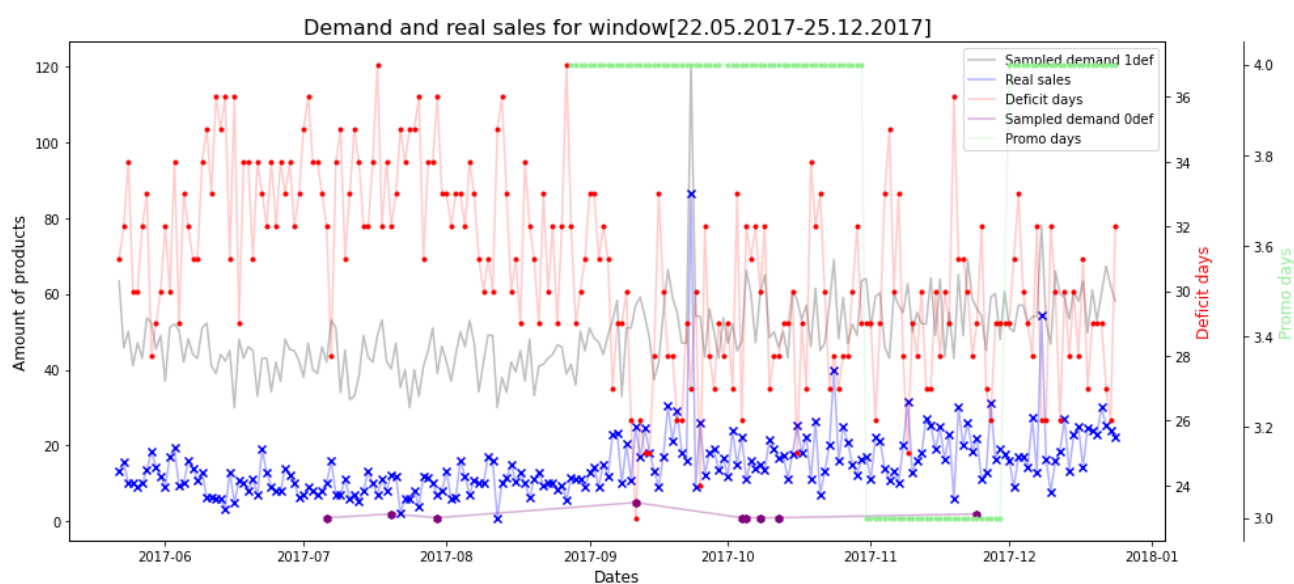
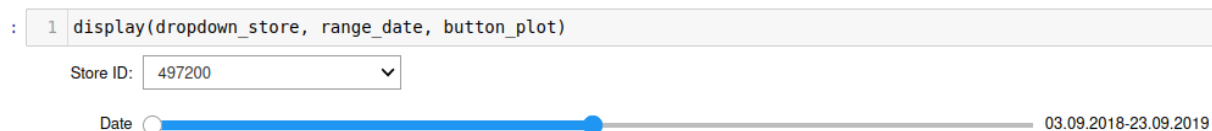


Иллюстрация 3 - пример другого графика для расчёта спроса с помощью метода окна

Так как значение величин спроса и количества дефицитных дней сильно отличается, было решено сделать отдельные шкалы значений для продаж (ось слева), для количества дефицитных дней (первая ось справа), для количества промо дней (вторая ось справа).

Интерактивность графиков заключается в том, что можно указать любой период времени и магазин товаров с помощью данных кнопок, а затем получить требуемый график.



После реализации интерактивных графиков и просмотра визуализированных данных за разные периоды времени были сделаны следующие выводы:

- 1) Не стоит реализовывать методы восстановления спроса с помощью расчёта среднего числа продаж, которые учитывают отдельные праздничные дни (около 10 во время года), так как тогда спрос может сильно подстроиться под единичные выбросы и ухудшится общее качество модели.
- 2) С учётом продаж, добавленных за дефицитные дни, общий восстановленный спрос выглядит достаточно правдоподобно. Сделано предположение, что следующие модели незначительно улучшат результаты.
- 3) На графиках были замечены большие выбросы по продажам в определенные дни за 2017 год, требовалось понять причину таких больших значений. Было проведено дополнительное исследование, более подробно изучены данные по каждому магазину и разным товаром в периоды таких дат. Выяснилось, что эта ситуация происходит почти для каждого магазина для нескольких товаров, это были повышенные продажи в определённые дни, было решено, что это является нормой, что это особенность системы подсчёта продаж, либо некая договоренность с какими-то поставщиками. Решено, что

отдельно под эти выбросы модель подстраиваться не будет, так как они разовые.

5.5 Методы с использованием скользящего среднего для восстановления спроса

Следующим действием в моём проекте была реализация методов, которые использовали информацию за некое количество соседних дней, то есть эти методы используют информацию за прошлые и за будущие дни. Этот метод имеет преимущество над прошлым, так как в данном способе лямбда рассчитывается отдельно для каждого дня, то есть интенсивность спроса настраивается более точно, более того, используется информация только за ближайшие периоды времени.

На данный момент алгоритмы реализован с помощью `pandas.rolling()`. Выбирается окно размера= N дней и минимальный период= K . Лямбда считается по дням, которые попадают в границы интервала окна и её значение ставится в день, который является серединой интервала. То есть за эти дни суммируется количество продаж и считается среднее. Так как в датасете присутствуют NaN, то такие дни будут игнорироваться, а лямбда будет считаться, если в интервал попадает количество дней с известными продажами, которое не меньше минимального периода. Иначе за текущий день ставится лямбда=NaN. Следовательно, в первые или последние дни лямбда считается по меньшему окну, потому что отсутствует примерно половина интервала окна. Оставшиеся NaN заполняются с помощью `ffill`, `bfill`.

Плюс данной модели заключается в том, что можно регулировать размер окна, то есть количество дней, за которые считается спрос, а также регулируется минимальный период, за который учитывается спрос или ставится пропуск. Отдельно была реализована ещё одна модель, в которой

было больше параметров. Во второй модели отдельно был параметр, который отвечал за учёт дней с дефицитом и без него, а также учёт дней с промо и без промо. Таким образом, помимо размера окна и минимального окна можно было отдельно считать спрос только за определённые наборы дней, что позволяет более гибко настраивать модель.

5.6 Модели машинного обучения для восстановления спроса

Одним из последних этапов в проекте была реализация моделей машинного обучения. Перед этим обсуждались идеи о реализации эконометрических моделей для временных рядов таких как ARIMA, SARIMA, а также имеется библиотека prophet, которая сочетает в себе различные методы для временных рядов, однако в данных достаточно много пропусков, причём для некоторых пар “уникальный товар” - “уникальный магазин” много пропусков в самом начале временного ряда. Всё это сильно усложняет использование классических эконометрических моделей. Чтобы использовать такую модель, требуется последовательно предсказывать пропуски, а потом их же использовать для предсказаний будущих дат, но с помощью такого способа может сильно снизиться качество предсказаний, так как для предсказаний используются сами предсказания, а не истинные значения. Таким образом, было решено не использовать эти модели и использовать сразу модели машинного обучения, так как они эффективнее, к тому же в них можно указать признаки, которые отвечают за время. Более того, именно различные имплементации градиентного бустинга используются в соревнованиях по машинному обучению, где требуется предсказывать спрос.

В качестве базовой простой модели была выбрана линейная регрессия, она быстро и эффективно строится, её минус заключается в том, что она находит только линейные зависимости. В качестве признаков

для модели использовались: идентификатор промо дня, идентификатор дефицитного дня, средний спрос за (7, 15, 30 дней) со сдвигом на (-7, 0, +7) дней. За дефицитный день мы с коллегами решили считать дни, в которые количество товара в магазине было нулевым или равнялось количеству проданного товара. Последние признаки принято называть “lagged features”, они часто используются для предсказаний значений, когда имеется временной ряд. Сдвиг означает, что средний спрос считается, например, за следующие 15 дней от текущей даты, а это значение ставится на неделю назад от текущей даты. Таким образом, модель в день предсказания видит различные характеристики спроса за прошедшее и будущее время.

В качестве улучшенной модели был реализован градиентный бустинг, использовалась библиотека XGBoost. Основной плюс данной модели заключается в том, что данная модель логическая и может находить нелинейные зависимости в данных. В качестве признаков использовались такие же признаки, как и для линейной регрессии, они себя достаточно хорошо показали.

В качестве выборки для обучения использовались только недефицитные дни, так как только в данные дни был известен истинный спрос. Так как невозможно измерить качество алгоритма на тестовой выборке требовалось перед обучением из обучающей выборки выделить отдельно валидационную выборку, которая в обучении соответственно не использовалась. Для более достоверного результата использовалась кросс валидация на 5 фолдах, то есть выборка для обучения делалась на 5 частей, 4 части из которых использовались для обучения, а качество замерялось на 5. После проверки результатов модель была обучена на всей обучающей выборке, чтобы было больше данных для обучения.

6 Сравнение результатов алгоритмов

После имплементации всех алгоритмов требовалось выбрать различные метрики и замерить качество, чтобы сравнить их эффективность. В качестве базовой метрики вначале использовалась MSE:

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

Но эта метрика плохо подходит для нашего случая, так как спрос считается в количестве товаров и, например, если истинное значение равнялось 150, а предсказанное было 148, то это неплохой результат, но если истинный спрос был 3, а предсказанный спрос 1, то это результат сильно хуже в реальной жизни, но по метрике MSE они эквивалентные. Можно было бы использовать метрику MAPE, но она не имеет конечного интервала значений, а также сильнее штрафует за перепрогноз значения. Именно поэтому было решено использовать метрику SMAPE, которая как раз часто используется в соревнованиях по прогнозированию спроса:

$$\text{SMAPE} = \frac{100\%}{n} \sum_{t=1}^n \frac{|F_t - A_t|}{(|A_t| + |F_t|)/2}$$

where A_t is the actual value and F_t is the forecast value.

Данная метрика показывает отклонение от истинного значения, что и требуется в этой задаче. Таким образом, получились следующие результаты по алгоритмам:

Название алгоритма	SMAPE
Метод обычных и дней с промо	51.285
Метод скользящего среднего	51.223
Линейная регрессия	49.812
Градиентный бустинг	47.795

Видно, что модель градиентного бустинга была лучшего всех остальных, хоть это и не очень сильное улучшение. Первые методы показали почти аналогичное качество между собой. По метрике MSE результаты были аналогичными, однако я не расписываю их здесь, так как считать ошибку в нашей задаче в помощью MSE нерелевантно.

7 Заключение

В итоге были реализованы различные алгоритмы для восстановления спроса, они могут быть использованы для среды модели. Весь код доступен и выложен в общем репозитории, где также содержится код моих коллег для модели обучения с подкреплением [6]. Главная задача была решена, а именно был выполнен переход от константного спроса к нестационарности. В конце было проведено сравнение качества алгоритмов на валидационной выборке, которое показало, что градиентный бустинг был лучшим методом из реализованных для решения этой задачи.

В качестве дальнейших шагов в данной задаче можно рассмотреть некоторые другие методы восстановления спроса, однако основная проблема заключается в том, что даже очень хорошие модели не так сильно улучшают качество, при этом качество может зависеть от самих данных. В моём проекте не были опробованы методы глубинного обучения

такие как LSTM или GRU сети, поэтому можно реализовать эти архитектуры и сравнить качество. Также можно применить методы разладки временных рядов, чтобы выделить определенные интервалы на временных рядах и настраивать модель отдельно для каждого такого участка.

Список литературы

- [1] - Christian D. Hubbs, Hector D. Perez, Owais Sarwar // OR-Gym: A Reinforcement Learning Library for Operations Research Problems // URL: <https://arxiv.org/pdf/2008.06319.pdf>
- [2] - Handle Missing Values in Time Series // URL: <https://www.kaggle.com/juejuewang/handle-missing-values-in-time-series-for-beginners>
- [3] - Victor Hutse // Reinforcement learning for inventory optimization in multi-echelon supply chains
- [4] - M5 Forecasting Accuracy 1st place solution // URL: <https://www.kaggle.com/c/m5-forecasting-accuracy/discussion/163684>
- [5] - XGBoost optimized distributed gradient boosting library // URL: <https://xgboost.readthedocs.io/en/latest/>
- [6] - Reinforcement learning for Inventory Optimization demand restoration repository // URL: <https://github.com/aromanenko/RLIO/tree/master/Demand%20Restoration>