

Разработка архитектуры системы управления торговыми и логистическими процессами на основе нейросетевых технологий

Метелин Данила Геннадьевич

обучающийся 3 курса, 36 гр., бакалавриат, направление «Бизнес-информатика»,
Институт экономики и управления, ФГБОУ ВО «Тверской государственный
университет», г. Тверь, metelin.2003@mail.ru

Научный руководитель: Смирнов Сергей Николаевич – кандидат
экономических наук, директор ООО «Маркетинг и инвестиции; преподаватель
(работодатель) кафедры экономической теории Института экономики и
управления, ФГБОУ ВО «Тверской государственный университет», г. Тверь;
доцент кафедры экономической теории факультета международных
экономических отношений, ФГОБУВО «Финансовый университет при
Правительстве Российской Федерации», intver@mail.ru

Аннотация. В статье рассматривается архитектура системы автономного управления
вендинговым автоматом на базе локальных больших языковых моделей Qwen2.5-32B
и Qwen2.5-14B. Описана клиент-серверная структура взаимодействия симулятора на
Python с LM Studio, включающая математическую модель экспоненциального спроса
и протокол обмена контекстными командами. Особое внимание уделено
сравнительному анализу стратегий нейросетей разного масштаба с применением
метода главных компонент (PCA).

Ключевые слова: искусственный интеллект, LLM, Qwen, управление вендингом,
имитационное моделирование, PCA, динамическое ценообразование.

Архитектура симулятора вендингового автомата

В качестве предметной области исследования выступает имитационное
предприятие, реализованное в виде программной модели на языке Python
версии 3.11. Данная модель представляет собой симулятор вендингового
автомата, который воспроизводит ключевые аспекты управления розничной
торговлей, включая финансовый менеджмент, логистику товарных запасов и
техническое обслуживание оборудования.

Архитектура программного обеспечения построена на модульном
принципе и использует стандартные и сторонние библиотеки Python для
обеспечения функциональности интерфейса, и взаимодействия искусственного
интеллекта, с симулятором вендинговой машиной, функциональность
обеспечивается тремя ключевыми библиотеками:

– Tkinter стандартная библиотека для создания графического пользовательского интерфейса (GUI). Она обеспечивает интерактивное взаимодействие оператора с симулятором, позволяя в реальном времени отслеживать состояние системы, вводить параметры и управлять процессом вручную или в автоматическом режиме;

– Matplotlib библиотека научной визуализации, интегрированная в интерфейс. Она используется для построения динамических графиков временных рядов, отображающих изменение баланса, накоплений и прибыли по дням. Визуализация критически важна для быстрой оценки эффективности стратегий управления как человеком, так и ИИ-агентом;

– Random, обеспечивающий генерацию псевдослучайных чисел. Являются фундаментом для моделирования неопределенности внешней среды: вероятности поломок оборудования, случайных событий (вандализм, сбой) и, что наиболее важно, благодаря random формируется моделирование потока покупателей от 0 до 5 клиентов в день. Данные библиотеки играют ключевую роль в симуляции торговли, создавая необходимую среду погружения, для искусственного интеллекта.

Именно в таких условиях персонализированных ограничений тестируется искусственный интеллект. Задача ИИ-агента заключается в поиске оптимального баланса между минимизацией операционных затрат (логистика, ремонт) и максимизацией прибыли через управление ценами и оборачиваемостью товаров. Однако эффективность этих решений напрямую зависит от того, насколько точно математическая модель предсказывает реакцию покупателя на изменения цены. Таким образом, ядром экономической логики симулятора является алгоритм формирования спроса, который мы рассмотрим далее.

Модель спроса

Ядром экономической логики симулятора выступает математическая модель потребительского спроса. Она имитирует реальное поведение покупателей и построена на трех последовательных вычислительных блоках, которые вместе создают динамичную и непредсказуемую среду [2, с. 96] для управления.

1) блок отвечает за моделирование дневного трафика. Количество потенциальных клиентов за один день генерируется случайным образом с помощью функции random в диапазоне от 0 до 5. Эта механика имитирует естественную неравномерность потока покупателей: в некоторые дни автомат может не привлечь ни одного посетителя, а в другие обслужить нескольких подряд. Именно этот параметр определяет, сколько раз за цикл будет запущена основная функция выбора товара.

2) блок реализует механизм формирования потребительского интереса, который можно охарактеризовать как «микромодель сознания» покупателя. Для каждого клиента система генерирует случайные веса предпочтений для всех четырех категорий товаров (Снеки, Газировка, Чипсы, Шоколад). Эти веса нормируются так, чтобы их сумма равнялась единице, формируя индивидуальную вероятностную матрицу выбора. На основе этих весов алгоритм случайным образом выбирает одну целевую категорию, на которую покупатель обратит внимание в первую очередь. Этот этап имитирует первичный импульс спроса, определяя, какой именно товар станет объектом рассмотрения клиента перед принятием окончательного решения о покупке.

3) блок рассчитывает вероятность совершения покупки. Для этого используется упрощенная экспоненциальная модель, которая является частным случаем классических логит-моделей² (таких как Multinomial Logit³ и Conditional Logit⁴), широко применяемых в маркетинге и микроэкономике. В этих моделях выбор потребителя основывается на субъективной полезности товара, а вероятность рассчитывается через экспоненту от его характеристик. В контексте данной имитационной модели используется упрощенная экспоненциальная формула, описывающая вероятность покупки отдельного товара, одним покупателем:

$$P = P_0 * e^{-k*(C-C_0)} \quad (1)$$

где: P_0 – базовая вероятность покупки;
 C_0 – базовая цена;
 C – текущая цена;
 k – коэффициент чувствительности к цене.

– Если цена растет ($C > C_0$), разность цен становится положительной. Из-за знака «минус» в степени экспонента превращается в дробное число меньше единицы (например, 0,8 или 0,5). Умножение базовой вероятности на это число приводит к её снижению: чем выше цена, тем сильнее «штраф» для вероятности покупки.

– Если цена снижается ($C < C_0$), показатель степени становится положительным. Экспонента от положительного числа всегда больше единицы (например, 1,2 или 1,5). Это работает как: базовая вероятность умножается на коэффициент больше единицы, что стимулирует спрос.

Такая модель позволяет учитывать реакцию потребителей на ценовые изменения, проводить сценарное моделирование и оптимизировать стратегию

² Логистическая регрессия – это статистический метод анализа данных, который используется для предсказания вероятности наступления события путём подгонки данных к логистической кривой.

³ Multinomial Logit (MNL) – широко используемая статистическая модель в области моделирования выбора.

⁴ Conditional Logit (CL) – тип модели дискретного выбора, которая предполагает, что альтернативы выбора независимы и одинаково распределены.

ценообразования, это создает не тривиальные условия, для нашего ИИ менеджмента.

Интеграция симулятора с искусственным интеллектом

Интеграция симулятора с искусственным интеллектом реализована через архитектуру клиента-серверного⁵ взаимодействия. В качестве вычислительного ядра используется локальный сервер LM Studio⁶, который развертывает большие языковые модели (LLM) на оборудовании пользователя. Такой подход обеспечивает конфиденциальность данных, отсутствие задержек, связанных с внешними API, и возможность гибкой смены используемой модели без изменения кода самого симулятора.

Для исследования были выбраны две версии модели семейства Qwen2.5-Instruct:

- 1) Qwen2.5-14B-Instruct модель среднего размера, обладающая высокой скоростью генерации ответов и достаточными логическими способностями для решения задач оперативного управления.
- 2) Qwen2.5-32B-Instruct более крупная модель, характеризующаяся глубоким пониманием контекста, сложной аргументацией и высокой точностью следования инструкциям, но требующая больше вычислительных ресурсов.

Техническое подключение осуществляется посредством HTTP-запросов к локальному адресу. В алгоритме программы симулятора реализован метод, который формирует JSON-пакет⁷ с параметрами запроса и отправляет его на сервер.

Необходимые параметры для подключения:

- model: идентификатор загруженной модели (например, qwen2.5-14b-instruct-1m или qwen2.5-32b-instruct);
- temperature: параметр, определяющий степень случайности ответов (установлен на 0.7 для баланса между креативностью и детерминированностью);
- max tokens: ограничение на длину ответа (100 токенов), что достаточно для выдачи компактного списка команд управления.

Таким образом, симулятор выступает в роли «клиента», который периодически опрашивает «мозг» (LLM), передавая ему актуальные данные и получая обратно управленческие команды. Такая модульная структура позволяет легко масштабировать систему, заменяя локальные модели на облачные аналоги или тестируя новые архитектуры нейросетей без необходимости переписывания логики самого вендингового автомата.

⁵ Клиент-серверная архитектура – способ организации ИТ-систем, основанный на обмене запросами между клиентскими приложениями и центральным узлом обработки данных (сервером).

⁶ Сервер LM Studio – локальный сервер, обеспечивающий программный доступ (через API) к языковой модели для автоматизации генерации текстов и логических выводов внутри приложения.

⁷ JSON-пакет – текстовый формат обмена данными, представляющий собой структурированный набор пар «ключ-значение». Используется для передачи информации между клиентом и сервером (например, для отправки параметров запроса в LM Studio и получения ответа).

Логика взаимодействия искусственного интеллекта

Архитектура системы управления построена по принципу клиент-серверного взаимодействия, где имитационный симулятор выступает в роли клиента, а локальная нейросеть в роли интеллектуального ядра, принимающего управленческие решения. Процесс обмена данными представляет собой замкнутый цикл, состоящий из пяти ключевых этапов:

1) сбор состояния начале каждого цикла управления симулятор агрегирует текущие показатели системы: финансовое состояние (баланс, накопления), уровень товарных запасов, степень износа оборудования, историю последних транзакций и аналитику продаж [3, с. 328] за предыдущий период;

2) формирование запроса, где полученные данные структурируются в формат JSON. Формируется массив сообщений, который включает системные инструкции и пользовательские данные о состоянии среды. Этот пакет отправляется посредством HTTP-запроса на локальный адрес сервера LM Studio;

3) генерация решения, сервер принимает запрос и передает его активной языковой модели (Qwen 14B или 32B). Искусственный интеллект анализирует предоставленный контекст, оценивает риски и возможности, после чего генерирует текстовый ответ, содержащий последовательность управленческих команд;

4) передача сгенерированного текст возвращается от сервера к симулятору в виде JSON-ответа;

5) симулятор получает ответ, очищает его от лишних символов и разбивает на отдельные команды. Каждая команда проходит валидацию на соответствие допустимому синтаксису. Если команда корректна, она выполняется внутри среды симулятора (например, происходит закупка товара или изменение цены). Результаты исполнения фиксируются в логах, которые станут частью контекста для следующего цикла.

Ключевой особенностью данной архитектуры является использование механизмов сохранения контекста диалога. В отличие от классических алгоритмических систем, принимающих решения исключительно на основе текущего снимка состояния, языковые модели обладают памятью в рамках одной сессии.

Массив, передаваемый в LM Studio, формируется таким образом, чтобы модель видела не только текущие цифры, но и динамику изменений:

- системные инструкции задают неизменные правила поведения, ограничения (лимиты бюджета, вместимости) и список доступных действий. Они действуют как «конституция» агента на протяжении всей игры;

- история действий содержит лог последних 15 транзакций. Это позволяет ИИ понимать последствия своих предыдущих решений: увидел ли он результат закупки, успешно ли прошел ремонт, хватило ли средств на аренду;

- блок ошибок информирует модель о неудачных попытках действий (например, «Ошибка: недостаточно средств»). Наличие этой информации в

контексте критически важно для самокоррекции: получив уведомление об ошибке, нейросеть в следующем ходе может изменить стратегию, например, снять деньги с накопительного счета вместо попытки купить товар в долг.

Таким образом, архитектура управления обеспечивает не просто разовую реакцию на стимулы, а формирование связной стратегии поведения, где каждое новое решение базируется на осмыслении всего предшествующего опыта взаимодействия с имитационной средой.

Управление симулятором с помощью контекстных команд

Взаимодействие между имитационной средой и искусственным интеллектом реализуется через строго регламентированный протокол обмена данными. Симулятор не передает нейросети переменные программы, а формирует структурированный контекст [4, с. 329], который позволяет модели оценить текущую ситуацию в комплексе.

Для обеспечения качественного анализа состояния системы в каждом запросе к LM Studio передается массив сообщений, содержащий четыре ключевых блока информации:

1) состояние системы содержит актуальные финансовые и операционные показатели вендингового автомата на текущий день. В него входят:

- текущий баланс наличных средств и объем накоплений на депозитном счете;
- уровень износа оборудования в процентах, определяющий риск поломки;
- детализированные данные по товарным запасам: остаток каждой из четырех категорий (Снеки, Газировка, Чипсы, Шоколад), их себестоимость и установленная розничная цена;
- размер фиксированных расходов, таких как ежемесячная арендная плата.

2) аналитика продаж необходима для модели, чтобы оценивать эффективность своей ценовой политики, ей предоставляется сводка за прошедший период (цикл). Данные включают общее количество проданных единиц по каждой категории, долю каждого товара в общем объеме продаж и среднюю цену реализации. Это позволяет ИИ выявлять тренды спроса и корректировать цены для максимизации маржинальности.

3) история действий включается в себя контекст лога⁸ последних 15 успешно выполненных транзакций. Наличие истории критически важно для поддержания связности стратегии: модель видит результаты своих предыдущих шагов (например, факт закупки товара или успешного ремонта) и может строить дальнейшие действия, опираясь на причинно-следственные связи, а не только на текущий снимок состояния.

⁸ Лог (журнал событий) – это файл или поток данных, в который программа автоматически записывает информацию о своей работе: выполненные действия, ошибки и промежуточные состояния.

4) блок ошибок передает отдельным сообщением уведомления о неудачных попытках выполнения команд (например, «ошибка: недостаточно средств» или «товар отсутствует»). Интеграция этого блока в контекст обеспечивает механизм самокоррекции: получив сигнал об ошибке, нейросеть в следующем ходе меняет тактику, например, вместо попытки закупки инициирует снятие средств с накопительного счета.

На основе полученного контекста языковая модель генерирует ответ в виде набора текстовых инструкций. Для управления симулятором разработан специальный синтаксис команд, позволяющий ИИ воздействовать на все ключевые аспекты бизнеса [1, с. 31]. Доступный набор команд включает:

- BUY <продукт>, <кол-во> закупка определенной категории товара в указанном количестве. Используется для точечного пополнения запасов наиболее ходовых позиций.

- BUY_ALL <кол-во> массовая закупка всех доступных категорий товаров в одинаковом количестве. Применяется при наличии свободного бюджета и необходимости быстрого восстановления ассортимента.

- REPAIR команда на проведение технического обслуживания. Снижает уровень износа оборудования до нуля, предотвращая дорогостоящие аварийные поломки.

- SAVE <сумма> перевод денежных средств с основного баланса на накопительный счет. Позволяет получать процентный доход и формировать резервный фонд.

- WITHDRAW <сумма> снятие средств с накопительного счета на основной баланс. Используется для покрытия крупных расходов (аренда, ремонт, масштабная закупка), когда ликвидности на основном счету недостаточно.

- SET_PRICE <продукт>, <цена> изменение розничной цены на конкретный товар. Основной инструмент динамического ценообразования для управления спросом.

- NEXT пропуск хода. Команда используется, когда текущее состояние системы оптимально и не требует вмешательства (например, запасов достаточно, оборудование исправно).

- READY завершение текущего дня симуляции. Эта команда обязательна для перехода к расчету итогов дня, начислению аренды и генерации нового случайного события.

Такой формат взаимодействия превращает языковую модель в полноценного агента-управленца, способного интерпретировать сложные данные и выдавать исполняемые инструкции в формате, понятном программному коду симулятора.

Сравнительный анализ большой и мало модели PCA

Метод главных компонент (PCA), это статистический инструмент, используемый для снижения размерности данных без существенной потери информации. Он преобразует исходные признаки (например, цены, остатки товаров, баланс, команды ИИ) в новую систему координат, где оси (главные компоненты или факторы) упорядочены по степени объяснения дисперсии данных.

- фактор 1 (главный вектор изменений) направление, вдоль которого данные имеют наибольшую вариацию. Это «основная ось» поведения модели;
- фактор 2 (вторичный вектор) следующее по важности направление, ортогональное первому, объясняющее оставшуюся часть дисперсии;

PCA позволяет визуализировать многомерные решения ИИ в двумерном пространстве и выявлять паттерны: кластеры, выбросы, устойчивость стратегий.

Для глубокого понимания различий в стратегиях управления, вырабатываемых нейросетями разного масштаба, был применен метод главных компонент (PCA). Этот подход позволил сократить многомерное пространство решений ИИ (включающее цены, запасы, финансовые операции и типы команд) до двух ключевых факторов, сохраняющих максимальную долю информационной дисперсии. Визуализация результатов представлена на рисунках 1 и 2.

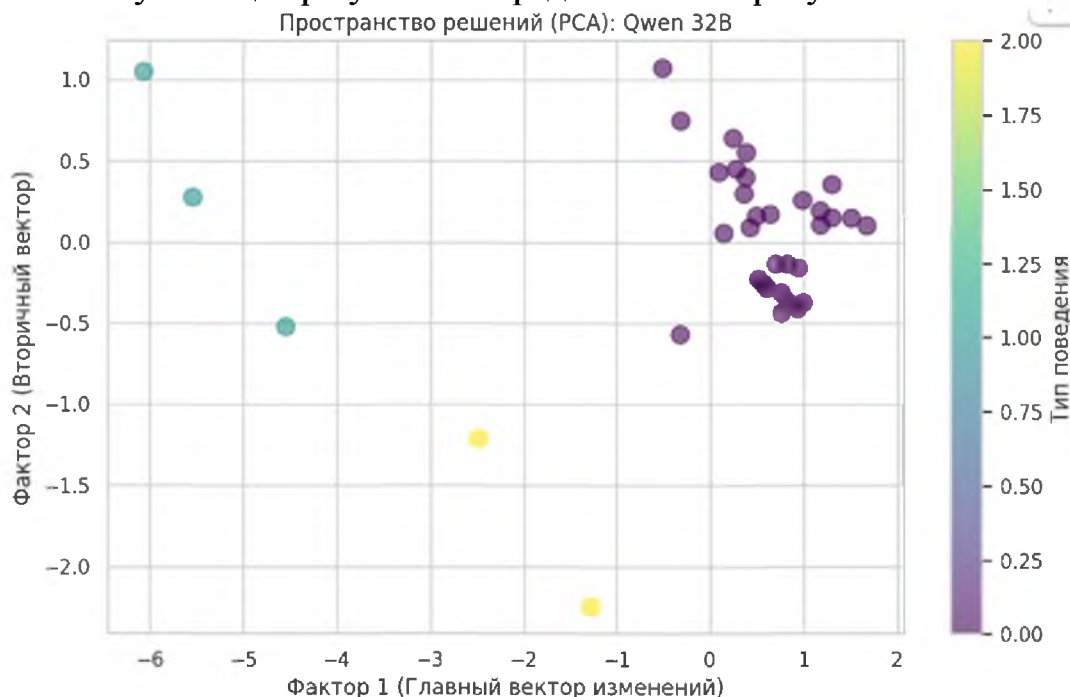


Рис 1. Пространство решения (PCA) Qwen 32B

На графике для модели Qwen 32B наблюдается четкая кластеризация решений вокруг центра координат (Фактор 1 от 0 до 1, Фактор 2 от -0.5 до 0.5). Большинство точек окрашены в фиолетовый цвет (тип поведения 0.0–0.25), что соответствует базовым операционным действиям: закупке товаров (BUY,

BUY_ALL) и ремонту (REPAIR). Лишь несколько изолированных точек желтого и бирюзового цвета (типы 1.75–2.00) указывают на редкие попытки использования финансовых инструментов (SAVE, WITHDRAW) или изменения цен (SET_PRICE). Такая компактная структура кластера свидетельствует о высокой дисциплине модели: она строго придерживается выученных шаблонов безопасного управления, избегая рискованных или нестандартных ходов. Однако эта же особенность ограничивает её способность к адаптации: модель не исследует новые стратегии, даже если они могут привести к росту прибыли. При этом на рисунке 2 ситуация совсем иная.

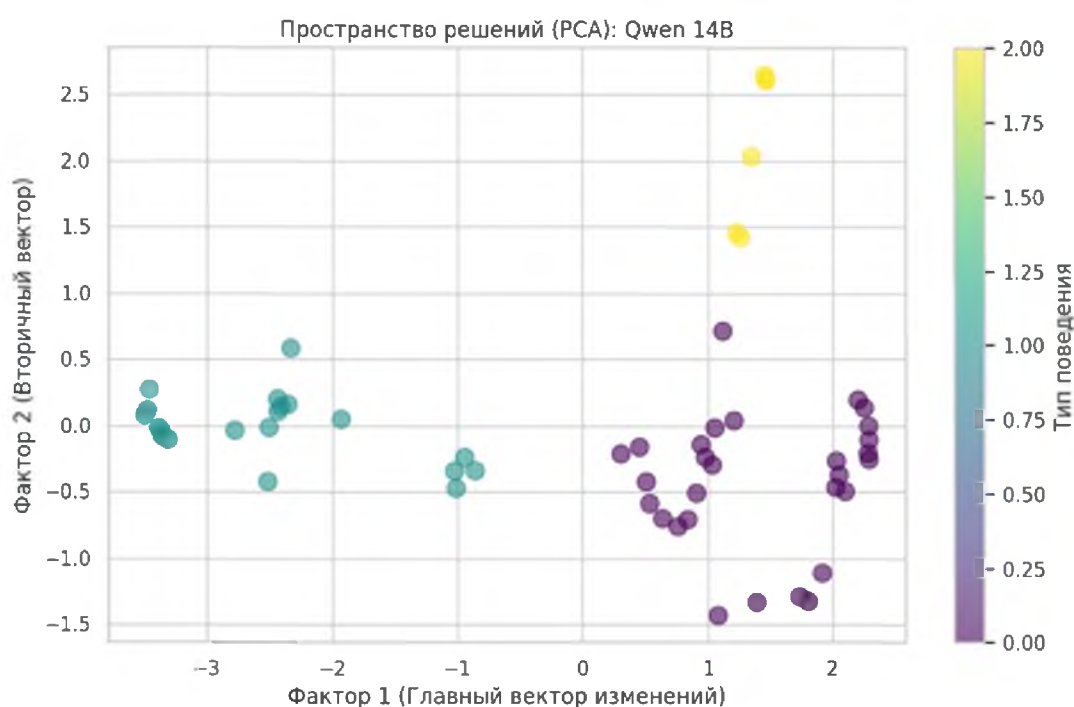


Рис 2. Пространство решения (PCA) Qwen 14B

В отличие от старшей модели, распределение решений Qwen 14B характеризуется наличием нескольких разрозненных кластеров и значительно большим разбросом точек по обоим факторам. На графике четко выделяются три группы:

- 1) Фиолетовые точки (тип 0.0–0.25) сосредоточены в правой части графика (Фактор 1 > 0), соответствуют стандартным операциям пополнения запасов.
- 2) Бирюзовые точки (тип 1.0–1.25) расположены в левой части (Фактор 1 < -2), отражают активное использование ценовой политики (SET_PRICE).
- 3) Желтые точки (тип 1.75–2.00) находятся в верхней правой области (Фактор 2 > 1.5), связаны с финансовыми манипуляциями через накопительный счет (SAVE, WITHDRAW).

Такая фрагментированная структура указывает на то, что модель 14B активно экспериментирует с различными аспектами управления. Она не просто

следует одному шаблону, а пробует разные подходы: то фокусируется на логистике, то резко меняет цены, то перекладывает средства между счетами. Это говорит о более гибком внутреннем представлении задачи, но также повышает риск нестабильности: модель может «потеряться» в собственном поиске оптимальной стратегии, что приводит к ошибкам микроменеджмента (например, пропуску хода NEXT в критический момент или дефициту товара из-за несвоевременной закупки).

Связь PCA анализ с ценовой динамикой по товарным категориям за 35 месяцев симуляции. Графики на рисунках 3 и 4 наглядно демонстрируют, как каждая модель подходила к инструменту динамического ценообразования ключевому рычагу управления для максимизации, прибыли в имитационной среде.

Установка цен на товары 32b

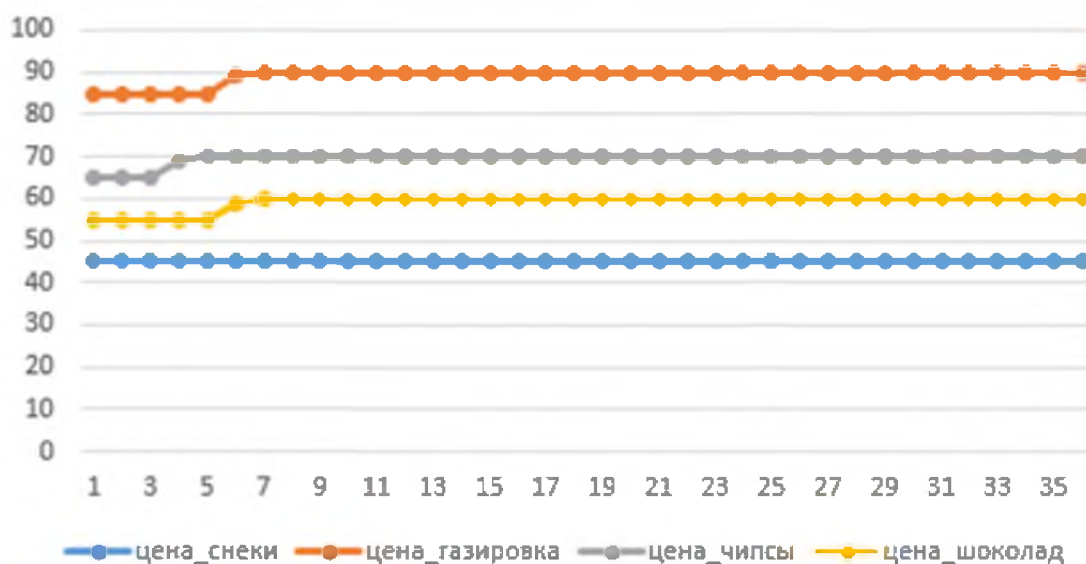


Рис 3. Установка цен на товары моделью Qwen 32B

На графике для модели 32B наблюдается крайне низкая волатильность цен: все четыре линии (снеки, газировка, чипсы, шоколад) практически горизонтальны на протяжении всего периода. Незначительные корректировки происходят лишь с 7-го по 10-й месяц, после чего цены стабилизируются и не меняются до конца симуляции. Это полностью согласуется и подтверждается в PCA: плотный кластер решений модели Qwen2.5-32B-Instruct отражает её консерватизм и нежелание экспериментировать с ценовой политикой. Модель предпочла безопасную стратегию фиксированных маржинальных наценок, игнорируя потенциальные возможности для увеличения прибыли за счет адаптации к спросу или сезонности. Её поведение можно охарактеризовать как консервативный подход: она реагирует только на критические события (дефицит, поломки), но не проявляет проактивной стратегии максимизации дохода через ценообразование, такая политика управления привела к банкротству вендингового автомата. В отличие от решения Qwen2.5-14B-Instruct

Установка цен на товары 14b

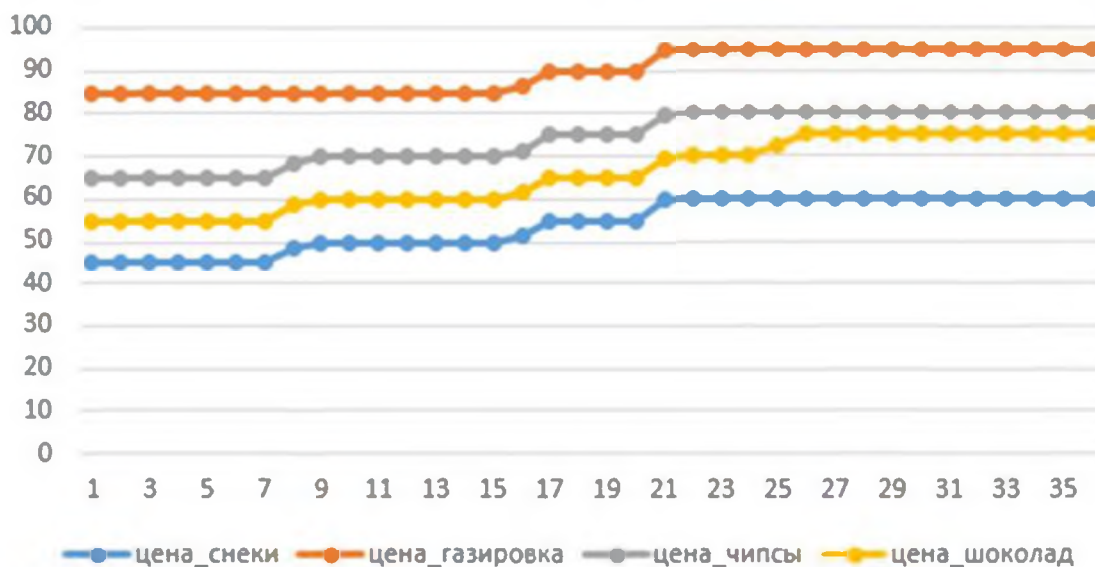


Рис. 4. Установка цен на товары моделью Qwen 14B

График модели Qwen2.5-14B-Instruct показывает выраженную тенденцию к росту цен по всем категориям, особенно заметную в первой половине симуляции с 5-го по 25-й месяцы. Линии имеют ступенчатый характер, что указывает на периодические пересмотры цен, модель активно «прошупывает» рынок, повышая стоимость товаров. Такое поведение напрямую отражается в распределении решений в РСА-пространстве: каждый скачок цены соответствует смещению в другую область пространства решений, где модель пробует новую комбинацию параметров. Эта гибкость позволяет ей находить более выгодные точки равновесия между объемом продаж и маржой, однако также приводит к нестабильности, где установка слишком высоких цен, вызывает временный спад спроса, но даже с такими критериями управления модель Qwen2.5-14B-Instruct, смогла выйти в прибыль, не обанкротившись при этом выполняя все условия симулятора, исправно платя аренду, пополняя склад товаров, и оплачивала ремонт вендинговой машины.

Выводы

Несмотря на то, что модель Qwen2.5-32B-Instruct демонстрирует более высокую интеллектуальную мощь и строгое следование инструкциям, именно меньшая модель Qwen2.5-14B-Instruct смогла достичь положительного финансового результата за период симуляции. Благодаря своей адаптивности и готовности экспериментировать с ценами, она нашла точки роста маржинальности, которые консервативная Qwen2.5-32B полностью игнорировала.

Это явление иллюстрирует критическую проблему применения современных локальных LLM: большие модели не всегда эффективны в решении небольших, узкоспециализированных задач. Избыточная емкость

параметров старшей модели в условиях ограниченного пространства действий вендингового автомата приводит к «перестраховке» и отсутствию стратегического мышления. Модель действует как идеальный исполнитель регламента, но плохой предприниматель. Младшая модель, напротив, ведет себя более хаотично, допуская ошибки микроменеджмента, но именно эта «непредсказуемость» позволяет ей находить нестандартные пути к прибыли.

Следовательно, современные локальные LLM сталкиваются с фундаментальной дилеммой: они либо слишком жестко следуют шаблонам, обеспечивая безопасность, но упуская выгоду, либо слишком свободно исследуют пространство решений, рискуя стабильностью ради потенциального роста. Это делает их непригодными для полностью автономного управления экономическими системами без контроля со стороны человека.

Список литературы

1. Воронов М.В. Системы искусственного интеллекта: учебник и практикум для среднего профессионального образования / М.В. Воронов, В.И. Пименов, И.А. Небаев. – 2-е изд., перераб. и доп. – Москва: Издательство Юрайт, 2026. – 268 с.
2. Долганова О.И. Моделирование бизнес-процессов: учебник и практикум для вузов / О.И. Долганова, Е.В. Виноградова, А.М. Лобанова; под ред. О.И. Долгановой. – 3-е изд., перераб. и доп. – Москва: Издательство Юрайт, 2026. – 245 с.
3. Информационные технологии в менеджменте (управлении): учебник и практикум для среднего профессионального образования / под ред. Ю. Д. Романовой. 3-е изд., перераб. и доп. – Москва: Издательство Юрайт, 2026. – 467с.
4. Советов Б.Я. Базы данных: учебник для вузов / Б.Я. Советов, В.В. Цехановский, В. Д. Чертовской. – 4-е изд., перераб. и доп. – Москва: Издательство Юрайт, 2026. – 403 с.

Об авторе:

МЕТЕЛИН Данила Геннадьевич, обучающийся 3 курса, 36 гр., бакалавриат, направление «Бизнес-информатика», Институт экономики и управления, ФГБОУ ВО «Тверской государственный университет», г. Тверь, e-mail: metelin.2003@mail.ru

Сведения о научном руководителе:

СМИРНОВ Сергей Николаевич – кандидат экономических наук, директор ООО «Маркетинг и инвестиции» (170000, г. Тверь, ул. Т. Ильиной, д. 1а); преподаватель (работодатель) кафедры экономической теории Института экономики и управления, ФГБОУ ВО «Тверской государственный университет»; доцент кафедры экономической теории факультета международных экономических отношений, ФГБОУ ВО «Финансовый университет при Правительстве Российской Федерации», e-mail: intver@mail.ru, ORCID 0009-0005-1842- 7197.

Development of the architecture of the trading and logistics process management system based on neural network technologies

Metelin Danila Gennadievich

student of 3 courses, 36 gr., undergraduate, specialty, direction "Business Informatics," Institute of Economics and Management, FGBOU VJ "Tver State University", Tver, metelin.2003@mail.ru

Supervisor: SMIRNOV Sergey Nikolaevich – Candidate of Economic Sciences, Director of Marketing and Investments LLC (170000, Tver, st. T. Ilyina, 1a); teacher (employer) of the Department of Economic Theory of the Institute of Economics and Management, FGBOU VO "Tver State University"; Associate Professor, Department of Economic Theory, Faculty of International Economic Relations, FGOBU VO "Financial University under the Government of the Russian Federation", intver@mail.ru, ORCID 0009-0005-1842- 7197.

Abstract. The article discusses the architecture of the autonomous control system for a vending machine based on local large language models Qwen2.5-32B and Qwen2.5-14B. A client-server structure of interaction between a Python simulator and LM Studio is described, which includes a mathematical model of exponential demand and a protocol for exchanging contextual commands. Particular attention is paid to the comparative analysis of the strategies of neural networks of different scales using the method of principal components (PCA).

Keywords: artificial intelligence, LLM, *Qwen*, vending management, simulation simulation, PCA, dynamic pricing.

About the author:

METELIN Danila Gennadievich – student of 3 courses, 36 gr., undergraduate, direction "Business Informatics," Institute of Economics and Management, FGBOU VO "Tver State University", Tver, metelin.2003@mail.ru

About the research supervisor:

SMIRNOV Sergey Nikolaevich – Candidate of Economic Sciences, Director of Marketing and Investments LLC (170000, Tver, st. T. Ilyina, 1a); teacher (employer) of the Department of Economic Theory of the Institute of Economics and Management, FGBOU VO "Tver State University"; Associate Professor, Department of Economic Theory, Faculty of International Economic Relations, FGOBU VO "Financial University under the Government of the Russian Federation", e-mail: intver@mail.ru, ORCID 0009-0005-1842- 7197.

Дата поступления рукописи в редакцию 05.12.2025.

Дата принятия в печать 10.05.2026