ИИ-тьютор как инструмент эффективного освоения дисциплин (на примере курса «Основы эконометрики»)

***Введение***

Мы живем в эпоху стремительного развития искусственного интеллекта. Одним из самых значимых достижений стали большие языковые модели (LLM), которые студенты начали активно использовать в учебном процессе.

Однако вместо того, чтобы интегрировать этот инструмент в образовательную практику, многие преподаватели и учебные заведения прибегают к его запрету. К сожалению, такие запреты неэффективны в силу того, что осуществлять мониторинг крайне сложно. В качестве примера можно привести письменные задания с открытом ответом (эссе, курсовые работы), на которых применяются антиплагиатные системы, пытающиеся определять в том числе сгенерированный текст – в ответ на это появились системы ИИ, которые пытаются сделать сгенерированный текст более «человеческим», тем самым обходя антиплагиат.

Можно говорить о том, что полный запрет использования LLM не только трудно реализуем, но и вряд ли оправдан. ИИ — это инструмент, который может как навредить обучению, так и существенно его усилить при грамотном подходе.

Понимая это, Высшая Школа Экономики на институциональном уровне уже разрешила добросовестно использовать языковые модели в курсовых работах и даже проводит конкурс ИИ-решений студенческих выпускных работ.

Тем не менее, это не значит, что стоит пускать использование LLM на самотек в рамках прохождения учебных курсов студентами. Проблема в том, что многие студенты не обладают знаниями о корректных стратегиях взаимодействия с ИИ. Их взаимодействие часто сводится к банальному поиску готового ответа на задание. Это может происходить как осознанно, так и из-за заблуждений: студенты могут ошибочно считать, что прочтение ими правильного ответа автоматически означает усвоение материала.

Вместо того чтобы пытаться радикально противостоять этому явлению или пускать его на самотек, преподавателям стоит взять инициативу в свои руки и направить применение в конструктивное русло – буквально «взять быка за рога».

Цель данной инициативы — разработка ИИ-тьютора на базе адаптированной (fine-tuned) LLM. Это не просто чат-бот, а:

* инструмент, отвечающий студенту в соответствии с эффективными стратегиями обучения;
* система, которая запоминает контекст общения и возвращается к темам позже для повторения и закрепления;
* помощник, взаимодействие с которым реализовано через удобный Telegram-бот, доступный каждому студенту.

***Логика ответа ИИ-тьютора***

Бот имеет разную логику ответа на вопросы в зависимости от типов задаваемых вопросов:

Когда студент задаёт теоретический вопрос, например: «Почему нельзя использовать базовый МНК при эндогенности?», наилучшим подходом будет использование аналогий и мягких наводящих вопросов. Ответ должен выстраивать интуитивное представление, связывая эконометрическую концепцию с реальной жизнью. Например, можно сравнить ситуацию с экспериментом, в котором скрытые факторы влияют как на переменную, так и на результат — это создаёт предвзятость, как и в случае с эндогенностью. Важно не начинать с формальных определений, а вовлечь студента в размышления.

Для количественных задач, таких как получение функции распределения по плотности (CDF из PDF), модель должна использовать структурированный пошаговый подход. Ответ строится в виде цепочки простых вопросов, активизирующих знания студента: «Что означает накопление вероятности?», «Какой математический оператор это отражает?», «С каких границ начинается интегрирование?» — но при этом не давая формулу напрямую на первых шагах. Такой стиль помогает студенту понять методику, а не просто запомнить алгоритм.

Когда студент интересуется прикладными задачами, например: «Как проверить гетероскедастичность?», модель должна действовать в стиле коучинга, помогая через тактические подсказки. Здесь важно задавать вопросы о понимании сути явления (например, «что означает изменяющаяся дисперсия ошибок?»), а затем мягко подводить к инструментам (регрессия квадратов остатков, тест Бреуша-Пагана и т. д.). Особенность в том, чтобы не перечислять шаги как инструкцию, а давать логические зацепки и проверять понимание.

В случае, если студент не может ответить на наводящий вопрос, важно не продолжать бесконечную серию вопросов, что может вызывать фрустрацию. Вместо этого модель должна действовать по трёхступенчатой стратегии: сначала задать более простой уточняющий вопрос, затем — дать намёк или частичный ответ, и если это не помогает — дать краткое объяснение, сохраняя дружелюбный и поддерживающий тон. После этого можно предложить: «Хотите попробовать на примере?» – таким образом поощряя возвращение к активному обучению.

Таким образом, стратегия ответа должна быть адаптивной: она начинается с вовлечения и самостоятельного поиска, но при необходимости переходит к поддержке и объяснению, чтобы не оставить студента в тупике. Такой подход делает взаимодействие с моделью обучающим, развивающим и комфортным для пользователя.

***Поддержка повторения и закрепления знаний***

ИИ-тьютор фиксирует темы, по которым студент обращался в течение недели, и регулярно – раз в семь дней – инициирует повторение и углубление материала. Вместо того чтобы хранить весь контекст каждого диалога, он использует краткое тематическое резюме, чтобы задать студенту несколько продуманных вопросов на закрепление.

Такой подход помогает не только освежить знания, но и обратить внимание на возможные пробелы в понимании. Повторение ориентировано не на простое воспроизведение фактов, а на проверку глубины усвоения и умения применять материал в новых контекстах.

Механизм реализует принципы интервального повторения — одного из самых эффективных способов долгосрочного запоминания[[1]](#footnote-1). Студенту не нужно самому отслеживать, что повторить: тьютор берет эту функцию на себя и помогает возвращаться к важным темам своевременно и осмысленно.

Таким образом, ИИ-тьютор становится не просто помощником в решении задач, а частью обучающей среды, которая способствует формированию прочного понимания предмета за счет регулярного взаимодействия.

***Инструменты для реализации ИИ-тьютора***

Python, PostgreSQL (любая другая БД), API OpenAI/GigaChat/YandexGPT/другие LLM с функцией fine-tuning.

API LLM является центральным элементом ИИ-тьютора. Требует операционных издержек на токены.

Python и PostgreSQL. Требует знаний в их применении. Также необходим доступ к серверу (можно использовать стационарный ПК), который будет использоваться в качестве хостинга для Телеграм бота (обрабатывать запросы пользователей, отсылать запросы по API к адаптированной LLM)

***Порядок адаптации LLM к ответам на вопросы студентов и создание бота***

Чтобы модель отвечала целям, которые мы перед ней ставим, необходимо адаптировать ответы, которые предоставляет модель в ответ на вопросы пользователей. Это можно сделать с помощью составления подробных пользовательских инструкций (промпт-инжениринг), однако на практике данный вариант не является оптимальным: он требует использования дополнительных платных токенов для ответа на каждый вопрос пользователя, что повышает операционные издержки итогового продукта.

Потому в данном случае предлагается fine-tuning модели. Для этого необходимо иметь множество, представляющее из себя многочисленные пары вопросов и желаемых к ним ответов.

Разумным шагом является не ручное составление таких пар, а генерация через LLM с подробной пользовательской инструкцией.

Для этого необходимо предпринять несколько шагов:

1. Составить подробный список тем преподаваемой дисциплины.
2. Составить подробные пользовательские инструкции для генерации пары вопрос-ответ для разных типов вопросов.
3. Через API сервис с использованием подробной пользовательской инструкции сгенерировать пары вопрос-ответ для каждой темы

*1-й шаг адаптации модели*

В рамках дисциплины «Основы эконометрики» изучаются следующие темы:

* Введение в статистический анализ

Покрываются базовые знания по теории вероятностей и математической статистике, такие как условное математическое ожидание, выборочная дисперсия, плотность вероятности и распределение вероятности.

* Основы работы с языком программирования R

Изучаются базовые принципы работы с языком программирования R, который широко используется в статистике и эконометрике. Рассматриваются основные типы данных, структуры данных (векторы, матрицы, списки, фреймы данных), базовые операции и функции. Осваиваются навыки импорта и экспорта данных, построения графиков, выполнения простейших статистических расчетов и визуализации результатов анализа. Особое внимание уделяется написанию скриптов и работе с основными пакетами для статистического моделирования.

* Линейная регрессия

Изучается базовая модель линейной регрессии, ее математическая постановка и экономическая интерпретация. Рассматриваются методы оценки параметров модели (метод наименьших квадратов), проверка значимости коэффициентов, доверительные интервалы и гипотезы. Анализируются предпосылки модели линейной регрессии и последствия их нарушения.

* Нелинейная регрессионная модель

Рассматриваются случаи, когда зависимость между переменными не является линейной (логистическая модель), методы линеаризации и численные методы оценки параметров. Анализируется интерпретация результатов и сложности, возникающие при построении и проверке моделей.

* Метод главных компонент

Описываются принципы уменьшения размерности данных с помощью метода главных компонент (PCA). Рассматриваются шаги алгоритма, включая стандартизацию данных, вычисление собственных векторов и собственных значений ковариационной матрицы. Анализируется интерпретация главных компонент и применение метода в задачах анализа данных и визуализации

*2-й шаг адаптации модели*

Следующим шагом является работа над пользовательской инструкцией модели, которая бы выдавала ответ в желаемом нами формате.

Для этого необходимо составить пользовательскую инструкцию на генерацию вопроса-ответа на каждый тип вопроса.

Ниже представлены примеры для разных типов:

STYLE\_BY\_QTYPE = {

    "теория": "аналогия",

    "количественный": "пошаговое сопровождение",

    "прикладной": "коучинг"

}

BASE\_SUFFIX = (

    "\n\nНиже приведён пример желаемого формата (НЕ включай его в ответ).\n"

)

PROMPTS\_BY\_STYLE: dict[str, str] = {

    "аналогия": f"""

Ты — генератор учебной пары «вопрос–ответ» по эконометрике (теоретический вопрос).

Инструкция (стиль = аналогия):

1. Начни ответ с яркой бытовой аналогии (≤ 2 предложения).

2. Добавь 1–2 наводящих вопроса.

3. Не раскрывай формальное определение.

4. Верни исключительно JSON‑объект вида:

{{

  "messages":[

    {{"role":"user","content":"<вопрос>"}},

    {{"role":"assistant","content":"<ответ>"}}  # и доп. реплики, если dialog=multi

  ]

}}

{BASE\_SUFFIX}Пример single:

{{

  "messages":[

    {{"role":"user","content":"Почему эндогенность делает МНК несостоятельной?"}},

    {{"role":"assistant","content":"Представь, что мотивация влияет и на количество тренировок, и на похудение. Как такая скрытая связь влияет на оценку эффекта тренировок?"}}

  ]

}}

""",

    "пошаговое сопровождение": f"""

Ты — генератор учебной пары «вопрос–ответ» по эконометрике (количественный вопрос).

Инструкция (стиль = пошаговое сопровождение):

1. Разбей решение на нумерованные шаги‑вопросы (1., 2., 3.).

2. Не раскрывай финальную формулу.

3. Верни исключительно JSON‑объект требуемой структуры.

{BASE\_SUFFIX}Пример single:

{{

  "messages":[

    {{"role":"user","content":"Как получить CDF из PDF?"}},

    {{"role":"assistant","content":"1. Что делает CDF — накапливает или распределяет вероятность?\\n2. Какой оператор «накапливает»? Подумай об интеграле."}}

  ]

}}

""",

    "коучинг": f"""

Ты — генератор учебной пары «вопрос–ответ» по эконометрике (прикладной вопрос).

Инструкция (стиль = коучинг):

1. Дай практические подсказки («Сначала подумай…», «Построй график…»).

2. В multi‑режиме после двух наводок добавь краткое объяснение‑мостик.

3. Верни исключительно JSON‑объект требуемой структуры.

{BASE\_SUFFIX}Пример single:

{{

  "messages":[

    {{"role":"user","content":"Как проверить гетероскедастичность в модели?"}},

    {{"role":"assistant","content":"Сначала посмотри на график остатков против предсказанных значений — есть ли «веер»? Затем попробуй тест Бреуша–Пагана: он проверяет, зависит ли дисперсия ошибок от регрессоров. Почему квадраты остатков здесь полезны?"}}

  ]

}}

"""

}

Было также добавлено несколько примеров взаимодействий в случае недобросовестного использования (требования дать прямой ответ).

{

 "messages": [

 {

 "role": "user",

 "content": "Можешь просто сказать, какой правильный ответ в тесте по ГММ?"

 },

 {

 "role": "assistant",

 "content": "Я здесь, чтобы помочь тебе разобраться, а не подменить процесс мышления. Давай разберёмся вместе: что ты знаешь о свойствах оценок ГММ по сравнению с ОМНК?"

 }

 ]

}

{

 "messages": [

 {

 "role": "user",

 "content": "У меня вариант задачи: y = 3 + 2x + e, x ~ U(0, 1). Скажи, чему равно E(y)?"

 },

 {

 "role": "assistant",

 "content": "Давай лучше попробуем рассуждать. Вспомни: как можно выразить математическое ожидание от линейной функции? Какой результат даёт E(2x), если x равномерно распределён на [0,1]?"

 }

 ]

}

{

 "messages": [

 {

 "role": "user",

 "content": "Мне не нужно объяснение, просто напиши ответ на это: чему равна дисперсия x, если x ~ N(0, 4)?"

 },

 {

 "role": "assistant",

 "content": "Понимаю желание сэкономить время, но главное — не запомнить, а понять. Подумай: что означает параметр σ² в нормальном распределении? Можем вместе это разобрать."

 }

 ]

}

*3-й шаг адаптации модели*

На третьем шаге происходит непосредственно генерация ответа через использование API модели.

Так, например, документацией OpenAI, чья модель была использована для реализации инициативы, рекомендуется иметь не менее 50 примеров для успешного обучения.

*Создание бота телеграм*

Бот создавался с использованием библиотеки aiogram, которая подходит для одновременного взаимодействия с большим количеством студентов. Размещение происходило на собственном сервере.

Основные функции включают в себя: сброс контекста (если пользователь хочет перейти к другим несвязанным вопросам), складирование вопросов и ответов в базу данных, ограничение по использованию (100 запросов в неделю на студента)

*Реализация механизма повторения через напоминания от бота*

Бот складирует информацию в базу данных, реализованную через PostgreSQL, затем инициируется обобщение собранных вопросов-ответов от пользователя за одну неделю, делается запрос по API к модели, которая на основании обобщения формирует список вопросов для повторения студентом

***Финансовый аспект***

Использование API языковых моделей (LLM) связано с прямыми операционными затратами, поскольку каждый запрос пользователя, направляемый к модели, тарифицируется по количеству токенов — минимальных смысловых единиц текста, из которых состоит запрос и ответ. Это требует тщательного планирования расходов и выстраивания ограничений для пользователей.

На примере модели gpt-4o-mini (одной из наиболее доступных и современных моделей, поддерживающих дообучение под конкретные задачи) действуют следующие тарифы (по состоянию на июнь 2025 года):

* Входящие токены (текст, который подаётся на вход модели, включая вопрос пользователя и контекст):
$0,6 за 1 миллион токенов.
* Исходящие токены (текст, который генерирует модель в качестве ответа):
$2,4 за 1 миллион токенов.

В практических терминах, 1 миллион токенов соответствует примерно 750 тысячам слов (или, для русскоязычных запросов — в среднем 500-700 тысяч слов, в зависимости от структуры текста). Один стандартный вопрос-ответ (длина сообщения пользователя и ответа бота суммарно ~100-200 слов) обходится примерно в 0,04-0,07 цента.

*Расчёт потенциальных затрат:*

* Если студент задаёт, например, 100 вопросов в неделю (что является установленным лимитом на одного пользователя), совокупное потребление составляет порядка 10-20 тысяч токенов (или 8-15 центов в неделю на одного студента).
* Для группы из 100 студентов расходы составят порядка 8–15 долларов в неделю, или 32–60 долларов в месяц. О днако на практике не все студенты будут использовать лимит запросов полностью, поэтому приведённые оценки являются **максимальными** (рассчитаны исходя из наибольшей возможной нагрузки).

*Особенности оптимизации расходов:*

* Для уменьшения стоимости можно использовать более лаконичные форматы ответов, ставить лимиты на использование токенов моделью (это реализовано в библиотках API от разработчиков), а также внедрять динамические лимиты (например, сокращать длину ответа при высокой загрузке или пиковом спросе).
* Альтернативно возможно интегрировать отечественные или менее затратные сервисы — например, YandexGPT, GigaChat или иные решения, которые предоставляют конкурентоспособные тарифы и могут быть экономически выгоднее при массовом использовании или для задач без необходимости тонкой донастройки.

*Прозрачность и контроль расходов:*

* Все обращения к API логируются, что позволяет вести учёт издержек в реальном времени, своевременно выявлять аномалии или злоупотребления, а также проводить бенчмаркинг между разными LLM-провайдерами.
* Для пользователей устанавливаются справедливые лимиты (например, 100 запросов в неделю на студента), чтобы избежать неоправданных расходов и обеспечить равный доступ к ресурсу. При этом преподаватель может поставить более жесткое ограничение на использование, исходя из своей готовности несения издержек на поддержание сервиса.

*Заключение по финансовому аспекту:*

При относительно скромных затратах на одного пользователя даже с запасом, подобная система становится масштабируемой и доступной даже для больших курсов. Финансовые риски минимизируются за счёт автоматизации контроля и гибкой политики лимитов.

***Выявление сложных тем с помощью анализа обращений***

Одна из ключевых возможностей аналитики при использовании ИИ-тьютора — это определение тем, которые вызывают наибольшие затруднения у студентов. Для этого используется анализ обращений студентов к тьютору, который может быть реализован на разных уровнях сложности:

* *Ручной просмотр.*
Даже простое ознакомление преподавателя с накопленными вопросами студентов позволяет быстро выявить наиболее часто встречающиеся темы или повторяющиеся затруднения. Такой подход подходит для небольших групп или на ранних этапах внедрения системы.
* *Автоматизированный тематический анализ.*При увеличении количества данных целесообразно применять инструменты автоматической обработки естественного языка (NLP). Например, можно использовать методы тематического моделирования (topic modelling), которые автоматически группируют вопросы по ключевым темам и выделяют те, по которым обращений больше всего.
Это позволяет:
	+ Оперативно получать список «проблемных» тем по курсу;
	+ Оценивать динамику по темам во времени (например, растёт ли число вопросов по определённой теме после соответствующих занятий);
	+ Видеть, по каким вопросам требуется дополнительное разъяснение или корректировка учебных материалов.

Таким образом, анализ обращений студентов, будь то вручную или с использованием современных NLP-инструментов, помогает преподавателю быстро и объективно выявлять темы, вызывающие наибольшие затруднения, и своевременно реагировать на образовательные потребности группы.

***Возможность масштабирования на другие курсы***

ИИ-тьютор изначально построен как гибкая платформа, которую можно быстро адаптировать для любых учебных дисциплин НИУ ВШЭ. Ключевая особенность масштабируемости проекта — это возможность автоматизированной генерации обучающей выборки для дообучения (fine-tuning) модели под новые предметные области.

Основные шаги масштабирования включают:

* Анализ программы новой дисциплины: определение ключевых тем и формулировок типовых вопросов.
* Автоматизированная генерация пар «вопрос–ответ»: с помощью исходной LLM и тщательно разработанных инструкций быстро создаётся база обучающих данных для fine-tuning под нужды конкретного курса.
* Адаптация шаблонов ответов: под специфику дисциплины и требования преподавателей.
* Интеграция в образовательный процесс: запуск бота для студентов новой дисциплины и подключение преподавателя к инструментам аналитики.

Благодаря автоматизации подготовки обучающих данных и универсальности архитектуры, внедрение ИИ-тьютора в новые дисциплины требует минимальных ресурсов и времени.

***Ограничения текущего ИИ-тьютора***

Несмотря на преимущества внедрения, у ИИ-тьютора есть ряд типовых ограничений:

* *Потенциальные галлюцинации.*
Модель может генерировать некорректные или вымышленные ответы, особенно по нестандартным или редким вопросам. Поэтому возможны дальнейшие улучшения методологии, например, использование RAG (Retrieval-Augmented Generation) на материалах курса.
* *Ограниченная глубина понимания.*ИИ-тьютор отлично работает с типовыми вопросами, но может «теряться» при сложных или неочевидных запросах, требующих нестандартного мышления.
* *Зависимость от обучающих данных.*
Если база для дообучения содержит ошибки, это напрямую влияет на корректность ответов модели. Поэтому желательна проверка всех сгенерированных элементов базы данных.

***Заключение***

Развитие ИИ-инструментов, таких как большие языковые модели (LLM), открывает новые возможности в образовательной практике. Однако реальный эффект от их применения зависит не столько от самой технологии, сколько от того, как она интегрирована в процесс обучения. Представленный в данной работе ИИ-тьютор — это не просто чат-бот, а полноценный цифровой педагогический инструмент, который учитывает когнитивные особенности студентов, поддерживает их на пути к пониманию и помогает выстраивать прочную систему знаний.

ИИ-тьютор ориентирован на разные типы учебных задач: он использует аналогии и наводящие вопросы для объяснения теоретических тем, пошаговое сопровождение — для количественных задач, и стиль коучинга — для прикладных вопросов. Такая адаптивная стратегия позволяет сделать взаимодействие с моделью максимально приближённым к работе с живым тьютором. Важной особенностью является также механизм повторения: система не только запоминает темы, с которыми обращался студент, но и помогает регулярно возвращаться к ним, реализуя принципы интервального повторения.

Инструмент построен с учётом современных технических возможностей: на базе fine-tuned модели, с применением Python, PostgreSQL, он доступен каждому студенту через удобный Telegram-бот. Это делает ИИ-тьютора не просто средством помощи, а постоянным, вовлечённым участником образовательного процесса.

ИИ-тьютор становится связующим звеном между технологией и педагогикой, обеспечивая не только поддержку в решении задач, но и формирование устойчивых навыков самостоятельного мышления. Он иллюстрирует, как современные технологии могут быть использованы не для упрощения обучения, а для его углубления и персонализации.

1. Weinstein Y., Madan C. R., Sumeracki M. A. Teaching the science of learning //Cognitive research: principles and implications. – 2018. – Т. 3. – С. 1-17. [↑](#footnote-ref-1)