

УДК 004.415

А.А. Табылдиева, tabyldievaaijamal29@gmail.com

А.К. Кемелова, altyshakemelova24@gmail.com

С.В. Корякин, srgkoryakin1@gmail.com

Э.А. Джалбиев, emirbekdzhalbiev@gmail.com

Кыргызско-Германский институт прикладной информатики

ТЕХНОЛОГИИ И МЕТОДЫ ПРОЕКТИРОВАНИЯ РЕКОМЕНДАТЕЛЬНЫХ СИСТЕМ ГОРОДСКИХ СЕРВИСОВ

Аннотация: в данной статье проводится обзор и сравнительный анализ методов и технологий проектирования рекомендательных систем для подбора городских услуг. Исследуются контентные, коллаборативные и гибридные подходы, описаны этапы сбора, очистки и предобработки данных, а также формирование пользовательских профилей и применение семантических эмбедингов. На основе экспериментальной оценки нескольких моделей обсуждаются их преимущества и ограничения в условиях городских сервисов.

Ключевые слова: рекомендательные системы, контентная фильтрация, коллаборативная фильтрация, гибридный метод, классические подходы, семантические эмбединги, LightGBM, холодный старт, предобработка данных

Введение

В настоящее время считается, что анализ существующих методов рекомендательных систем для городских сервисов и их проверка на реальных данных позволяют выявить ряд существенных ограничений применяемых подходов. Следует отметить, что быстрое развитие цифровых платформ и расширение ассортимента городских услуг — от медицинских и образовательных до бытовых, юридических и муниципальных — приводят к росту объема и усложнению структуры данных об объектах и взаимодействиях пользователей. Эти данные включают описания сервисов, поведенческие профили, географические координаты и временные закономерности активности. В таких условиях многие существующие модели, как считается, не справляются с корректным учётом пространственно-временных факторов при формировании и ранжировании рекомендаций, демонстрируют низкую эффективность при решении задачи «холодного старта» и не обеспечивают адекватных процедур оценки, отражающих как точность ранжирования, так и пространственную релевантность. Поэтому возникает необходимость разработки более совершенных методов, способных комплексно учитывать поведенческие, контентные и пространственно-временные особенности пользователей в условиях высокой динамики городской среды и ограниченности данных.

Рекомендательные системы представляют собой ключевой инструмент для прогнозирования пользовательских предпочтений и формирования персонализированных рекомендаций на основе анализа больших объёмов данных [1]. В контексте городских сервисов актуальность таких систем дополнительно определяется необходимостью учитывать пространственное расположение объектов и временные характеристики взаимодействий, поскольку именно эти факторы определяют практическую полезность и удобство выдачи для конечного пользователя.

При рассмотрении существующей литературы выявляется значительный разрыв между теоретическими разработками и прикладными требованиями городской предметной области: многие исследования фокусируются либо на коллаборативных, либо на контентных подходах, при этом пространственно-временные компоненты остаются недостаточно проработанными. Это ограничение снижает применимость традиционных моделей в задачах

локальных сервисов, где географическая привязка и временные паттерны поведения существенно влияют на релевантность рекомендаций.

Анализ конкретных трудов подтверждает указанные пробелы: часть работ стремится к объединению контентных и коллаборативных методов с целью повышения точности рекомендаций, но при этом опускает учет географической привязки и временных особенностей взаимодействий, что ухудшает их эффективность в городских условиях (Козлова и Германчук, 2022) [2]. Другие исследования сосредоточены на проблемах интеграции и структурирования данных в гибридных моделях, однако уделяют мало внимания адаптации алгоритмов к индивидуальным особенностям пользователей и к пространственно-временным зависимостям (Ерёмин и Моркулёв, 2023) [3]. В результате преобладают работы, описывающие общие архитектурные принципы и методы персонализации, но не предлагающие целостных решений, одновременно учитывающих поведенческие, контентные и пространственно-временные признаки.

Во избежание этих ограничений настоящая работа направлена на заполнение выявленного пробела путём аналитического рассмотрения, разработки и экспериментальной валидации гибридного подхода к построению рекомендательных систем, адаптированного к специфике городской среды. Основная цель исследования — создание и эмпирическая оценка интегрированной методологии, способствующей повышению точности и практической значимости рекомендаций для локальных сервисов за счёт одновременного учета семантических, поведенческих и пространственно-временных характеристик.

В качестве иллюстрации практических реализаций рекомендательных механизмов можно привести примеры крупных платформ, использующих сложные двухэтапные архитектуры (генерация кандидатов и ранжирование) и широкий набор сигналов взаимодействия: так, YouTube формирует рекомендации с опорой на глубокие нейросетевые модели и огромное количество пользовательских сигналов [4]; LinkedIn применяет распределённые вычислительные технологии для решения задач коллаборативной персонализации и масштабного ранжирования [5]; на C2C-маркетплейсах, таких как Lalafo, персонализация ранжирует объявления с учётом поведения пользователей, контекста запроса и характеристик объектов, включая локацию и отзывы [6].

Цель данного исследования — разработать и эмпирически оценить методы проектирования рекомендательных систем для городских сервисов с учётом пространственно-временных факторов. *Задачи:* проанализировать существующие подходы, методы и модели рекомендательных систем для подбора услуг в городской среде — выявить их сильные и слабые стороны; спроектировать гибридную модель, объединяющую контентные, поведенческие и пространственно-временные признаки; провести экспериментальную проверку предложенных методов на реальных данных и сравнить результаты с базовыми моделями.

Анализ современных подходов к построению рекомендательных систем

Выделяют *три ключевых подхода* к формированию рекомендаций: коллаборативная фильтрация, фильтрация на основе содержимого (контентная) и гибридные методы, сочетающие элементы первых двух подходов [7, 8].

Коллаборативная фильтрация

Одним из наиболее результативных подходов к формированию персонализированных рекомендаций является *метод коллаборативной фильтрации*, который опирается на анализ поведенческих данных и предпочтений пользователей. Основная идея данного метода

заключается в выявлении сходства между пользовательскими профилями либо в установлении близости между самими объектами контента. В процессе работы система обрабатывает историю взаимодействия пользователя с объектами, например, выставленные оценки фильмов, совершённые покупки или просмотренные веб-страницы. На основе этих данных выявляются группы пользователей с аналогичными интересами либо объекты с высокой степенью схожести, после чего формируются рекомендации, отражающие вероятные предпочтения конкретного пользователя [1].

Как отмечено в диссертации Кокачева В. А. [9], коллаборативная фильтрация опирается на два типа входных данных (рис. 1): множество пользователей и множество объектов интереса. Взаимосвязи между пользователями и объектами, как правило, фиксируются в виде оценок, выставляемых пользователями; эти оценки служат основой для последующего прогнозирования рейтингов для ранее оцененных объектов. Когда пользователь Pa взаимодействует с коллаборативной системой рекомендаций, система в первую очередь выявляет ближайших соседей, пользователей с похожим поведением, и затем на основе их оценок прогнозирует рейтинг, который мог бы поставить пользователь Pa .

Два основных подхода к коллаборативной фильтрации: *User-based* фильтрация и *item-based* фильтрация. Оба варианта предсказывают, в какой степени пользователь будет интересоваться объектами, которые до сих пор не были им оценены. *User-based* фильтрация идентифицирует k ближайших соседей активного пользователя и на основе этих ближайших соседей вычисляет прогноз пользователя для определенного объекта интереса. В отличие от *item-based* фильтрации, при коллаборативной фильтрации на основе элементов для текущего объекта ищутся «соседи», которые получили аналогичные рейтинги [10].



Рисунок 1 — Коллаборативная фильтрация

User-based фильтрация

Определим меру схожести между двумя пользователями $s(u,v)$, которая увеличивается с ростом сходства между пользователями u и v . Для каждого пользователя u рассмотрим множество пользователей, похожих на него:

$N(u) = \{v \in U \setminus \{u\} \mid s(u,v) > \alpha\}$, где α — настраиваемый гиперпараметр.

Допустим, мы хотим теперь оценить рейтинг R_{ui} , который пользователь u поставил бы объекту i . Для этого можно использовать оценки пользователей, похожих на u . Например, взвешенное среднее можно записать таким образом:

$$\hat{r}_{ui} = \frac{\sum_{v \in N(u)} s(u, v) r_{vi}}{\sum_{v \in N(u)} |s(u, v)|} \quad (1)$$

Абсолютное значение в знаменателе используется для корректной обработки случаев, когда пользователи мало похожи друг на друга, то есть для пар с отрицательной схожестью, которая может возникнуть, если при построении $N(u)$ выбрать достаточно маленькое α .

Можно улучшить метод предсказания рейтингов, учитывая, что у разных пользователей могут быть разные диапазоны оценок. Например, один пользователь чаще ставит оценки в диапазоне 1-3, а другой предпочитает 4-5. Иными словами, то, что для одного «нормально», для другого может быть «низко» или «высоко».

Чтобы устранить эту проблему, вместо использования исходного рейтинга r_{vi} учитывают отклонение оценки от среднего пользователя \bar{r}_v :

$$r_{vi} - \bar{r}_v. \quad (2)$$

Таким образом, учитывается только разброс оценок вокруг среднего, и итоговое предсказание становится более корректным с учётом индивидуальных особенностей пользователей.

$$\hat{r}_{ui} = \bar{r}_u + \frac{\sum_{v \in N(u)} s(u, v) (r_{vi} - \bar{r}_v)}{\sum_{v \in N(u)} |s(u, v)|} \quad (3)$$

Метрика для оценки схожести пользователей (**Корреляция Пирсона**):

$$s(u, v) = \frac{\sum_{i \in I_u \cap I_v} (r_{ui} - \bar{r}_u)(r_{vi} - \bar{r}_v)}{\sqrt{\sum_{i \in I_u \cap I_v} (r_{ui} - \bar{r}_u)^2} \sqrt{\sum_{i \in I_u \cap I_v} (r_{vi} - \bar{r}_v)^2}} \quad (4)$$

Корреляция Пирсона показывает, насколько согласованно два пользователя оценивают объекты относительно своих средних оценок. Значение, близкое к +1, говорит о схожих предпочтениях, к -1 говорит о противоположных, а около 0 — об их отсутствии [11].

Item-based фильтрация

Рассмотрим транспонированную постановку задачи, в которой вводится мера сходства объектов $s(i, j)$. В случае, когда требуется предсказать рейтинг, который пользователь u мог бы поставить ещё не оценённому им объекту i , используется множество $N(i)$ — набор объектов, наиболее схожих с i . В этом случае прогнозируемое значение рейтинга r_{vi} вычисляется по аналогии с пользовательским подходом (user-based), но с использованием схожести объектов:

$$\hat{r}_{ui} = \frac{\sum_{j \in N(i)} s(i, j) r_{uj}}{\sum_{j \in N(i)} |s(i, j)|} \quad (5)$$

Меру сходства между объектами целесообразно определить с помощью модифицированного косинусного сходства (adjusted cosine), выражающего степень корреляции оценок пользователей:

$$s(i, j) = \frac{\sum_{u \in U_i \cap U_j} (r_{ui} - \bar{r}_u)(r_{uj} - \bar{r}_u)}{\sqrt{\sum_{u \in U_i \cap U_j} (r_{ui} - \bar{r}_u)^2} \sqrt{\sum_{u \in U_i \cap U_j} (r_{uj} - \bar{r}_u)^2}} \quad (6)$$

где U_i является множеством пользователей, оценивших товар i . Важно также понимать, что \bar{r}_u — это средняя оценка пользователя, а не объекта, то есть это не корреляция Пирсона, на практике данный подход обычно работает лучше [11].

Недостатки и преимущества коллаборативной фильтрации указаны в таблице 1.

Таблица 1 — Недостатки и преимущества коллаборативной фильтрации

Преимущества	Недостатки
--------------	------------

Высокая точность рекомендаций	Проблема холодного старта
Не требует подробной информации о самих объектах	Чувствителен к разреженности данных

Контентная фильтрация

Контентная фильтрация — это метод рекомендательных систем, при котором рекомендации формируются на основе содержательных признаков самих объектов (текстовых описаний, метаданных, визуальных характеристик и др.) и их соответствия профилю пользователя, а не на основе поведения других пользователей [1]. В рамках данного подхода (рис. 2) [9] в качестве входных данных используются множества пользователей и категорий (или ключевых слов), характеризующих объекты интереса.



Рисунок 2 — Фильтрация на основе контента

Практически для каждого объекта строят числовое представление (эмбединг), например $e_i \in R^n$, которое может быть получено с помощью моделей глубокого обучения (BERT для текстов, CNN для изображений) или иными методами извлечения признаков [11]. Мы можем посчитать скалярное произведение (или косинусное расстояние) до оцененных пользователем объектов и оценить рейтинги, как:

$$\hat{r}_{ui} = \max_{j \in I_u, r_{uj} > \alpha} \rho(e_i, e_j) r_{uj}, \quad (7)$$

где ρ — скалярное произведение или косинусное расстояние между двумя векторами, I_u — множество оцененных пользователем объектов, а α — гиперпараметр. Таким образом, высокие рейтинги получают объекты, похожие на те, что понравились пользователю — мы получили простую ранжирующую модель.

Контентный подход удобен для работы с новыми объектами, поскольку он не зависит от истории взаимодействий других пользователей. В то же время такой метод нередко ограничивает разнообразие рекомендаций и может приводить к чрезмерно узкой персонализации. Поэтому на практике контентную фильтрацию часто используют как часть гибридных систем, где она дополняется коллаборативными и ранжирующими моделями для повышения качества рекомендаций. Плюсы и минусы контентной фильтрации указаны в таблице 2.

Таблица 2 – Недостатки и преимущества контентной фильтрации

Преимущества	Недостатки
Работа с новыми пользователями и объектами	Высокий порог внедрения
Простота и универсальность	Склонен «зацикливаться» на узком круге интересов пользователя
Персонализация на основе характеристик	Ограничен узкой специализацией

товара

Гибридные рекомендательные системы

Наиболее сложным и эффективным типом рекомендательных систем являются гибридные. На рисунке 3 представлено сравнение двух классических подходов к построению рекомендательных систем – коллаборативной и контентной фильтрации.

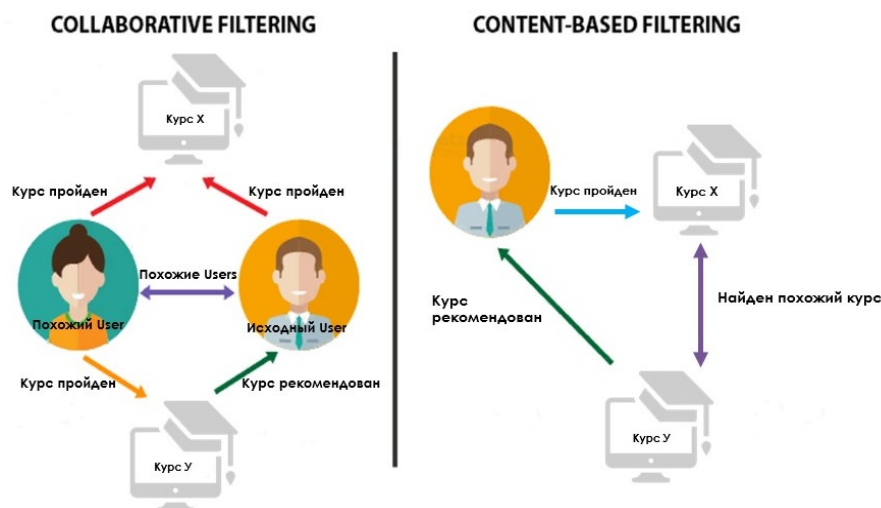


Рисунок 3 – Процесс реализации двух подходов

Слева показан принцип работы коллаборативной фильтрации (Collaborative Filtering): система анализирует поведение пользователей и выявляет группы с похожими интересами. Если «похожий пользователь» уже прошёл определённый курс (объект X), то этот курс может быть рекомендован «исходному пользователю», имеющему схожие предпочтения. Таким образом, рекомендации формируются на основе сходства между пользователями и их действиями.

Справа иллюстрируется подход контентной фильтрации (Content-Based Filtering): здесь система не сравнивает пользователей, а оценивает схожесть объектов. Если пользователь прошёл курс X, то ему будет предложен другой курс, похожий по содержанию или характеристикам.

Гибридные системы объединяют оба подхода (рис. 4), используя как поведенческие данные пользователей, так и описание самих объектов. Это позволяет повысить точность рекомендаций, компенсировать недостатки каждого отдельного метода (например, проблему «холодного старта») и обеспечить более разнообразные и релевантные предложения. В таких системах используются разнообразные подходы, например, нечеткие методы и искусственный интеллект [12], широко применяются подходы кластерного анализа данных [13]. Использование гибридных алгоритмов позволяет достичь более высокой точности.

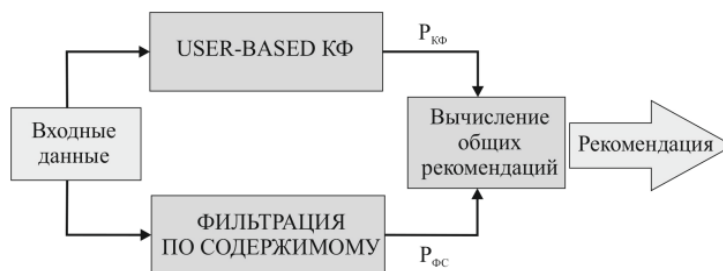


Рисунок 4 – Архитектура гибридных рекомендательных систем

Сравнение характеристик основных рекомендательных подходов

Три основных подхода к построению рекомендательных систем – коллаборативная фильтрация, фильтрация по содержимому и гибридные методы – опираются на различную исходную информацию и отличаются как своими достоинствами, так и ограничениями. В таблице 3 сравниваются основные характеристики трех основных подходов.

Таблица 3 – Сравнение характеристик основных рекомендательных подходов

Подход	Коллаборативная фильтрация	Фильтрация на основе контента	Гибридная
Быстрое развертывание	да	да	да
Адаптивность	да	да	да
Интуитивность	да	нет	да
Прозрачность	нет	нет	частично
Холодный старт	нет	да	да
Высокая степень участия	нет	нет	нет

Следовательно, на основании данных, представленных в таблице 3, можно заключить, что наиболее эффективным и рациональным вариантом при разработке рекомендательной системы является использование гибридного подхода.

Модели и подходы машинного обучения, выбранные для исследования

В исследовании были использованы три класса моделей: полносвязная нейронная сеть (Fully Connected Neural Network, FCNN), градиентный бустинг на решающих деревьях LightGBM и Transformer FNN. Ниже приведено краткое описание ключевых принципов работы каждой модели, типичных областей применения, а также обоснование их выбора для решения задач, рассматриваемых в исследовании.

Полносвязная нейронная сеть (FCNN)

FCNN представляет собой многослойный персептрон, в котором каждый нейрон слоя связан со всеми нейронами предыдущего слоя; обучение осуществляется методом обратного распространения ошибки с использованием стохастических вариантов градиентного спуска и методов регуляризации (L2, Dropout, BatchNorm) для борьбы с переобучением [14]. За счёт большой аппроксимирующей способности такие сети эффективно моделируют сложные нелинейные зависимости между признаками и целевыми переменными и выполняют роль универсального базового подхода и «головы» в гибридных архитектурах, поэтому были использованы как основной нелинейный регрессор/классификатор в задачах, где требуется извлечь высокоуровневые представления из исходных признаков [14]

На каждом слое сети выполняется линейное преобразование входного вектора с добавлением смещения:

$$z^{(l)} = W^{(l)}a^{(l-1)} + b^{(l)} \quad (8)$$

где $z^{(l)}$ – вектор активаций до применения функции активации, $W^{(l)}$ – матрица весов l -го слоя, $a^{(l-1)}$ – вектор выходов предыдущего слоя, $b^{(l)}$ – вектор смещений.

Далее к вектору $z^{(l)}$ применяется функция активации:

$$a^{(l)} = f(z^{(l)}) \quad (9)$$

где $f(\cdot)$ – нелинейная функция (сигмоида, ReLU, tanh и др.).

Обучение сети производится путём минимизации функции ошибки, например, среднеквадратичной:

$$L = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2 \quad (10)$$

где y_i – истинное значение, \hat{y}_i – предсказание модели, N – количество обучающих примеров.

LightGBM

LightGBM – это высокоэффективная реализация градиентного бустинга на решающих деревьях, ориентированная на работу с большими объёмами данных и высокой размерностью признакового пространства; ключевые оптимизации включают Gradient-based One-Side Sampling и Exclusive Feature Bundling, что позволяет сократить вычислительные затраты без заметной потери качества предсказаний [15]. Деревянные ансамбли традиционно демонстрируют хорошее соотношение скорости обучения, устойчивости к масштабированию признаков и интерпретируемости (feature importance), поэтому LightGBM выбран для быстрого прототипирования и в качестве сильного табличного бенчмарка в задачах скоринга и предсказания на структурированных данных [15].

Основные уравнения LightGBM:

1) Функция потерь

В бустинге минимизируется обобщённая функция ошибки:

$$L = \sum_{i=1}^n l(y_i, \hat{y}_i) + \sum_{k=1}^K \Omega(f_k) \quad (11)$$

где $l(y_i, \hat{y}_i)$ – функция потерь (например, логистическая или квадратичная),

f_k – базовый алгоритм (дерево решений),

$\Omega(f_k)$ – регуляризационный член, штрафующий сложность модели.

2) Регуляризация дерева решений

$$\Omega(f) = \gamma T + \frac{1}{2} \lambda \sum_{j=1}^T w_j^2. \quad (12)$$

где T – количество листьев в дереве,

w_j – вес предсказания в листе j ,

γ, λ – гиперпараметры регуляризации.

3) Второй порядок приближения для оптимизации

На t -й итерации новая модель строится с учётом разложения функции потерь по второму порядку:

$$L^{(t)} \approx \sum_{i=1}^n \left[g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i) \right] + \Omega(f_t) \quad (13)$$

где $g_i = \frac{\partial l(y_i, \hat{y}_i^{(t-1)})}{\partial \hat{y}_i^{(t-1)}}$ – первый градиент,

$h_i = \frac{\partial^2 l(y_i, \hat{y}_i^{(t-1)})}{\partial (\hat{y}_i^{(t-1)})^2}$ – второй градиент.

LightGBM не является классическим алгоритмом коллаборативной фильтрации, однако эффективно использует коллаборативные признаки при обучении.

Transformer Feedforward Neural Network (FNN)

Позиционно-независимая feedforward-сеть в архитектуре Transformer представляет собой двухслойную полносвязную нейронную сеть с промежуточной нелинейностью (как правило, ReLU или GELU), которая применяется к каждому элементу последовательности независимо от его позиции. Принцип её работы заключается в последовательном линейном расширении эмбединга до скрытого пространства повышенной размерности, последующей активации и обратного линейного сжатия, что обеспечивает дополнительную модельную ёмкость и позволяет извлекать более информативные представления после механизма внимания [16]. Типичные области применения FFN связаны с обработкой последовательных данных в задачах машинного перевода, обработки естественного языка, анализа временных рядов и других сценариев, где требуется учитывать сложные контекстуальные зависимости. В рамках настоящего исследования использование Transformer FFN обосновано необходимостью усилить выразительность модели при работе с последовательными признаками и обеспечить формирование высокоуровневых представлений, которые недоступны при использовании только блока внимания [16].

Основные уравнения Transformer FNN:

Пусть $x \in R^{d_{model}}$ – входной вектор для одной позиции последовательности. Тогда операция в слое FNN описывается следующим образом:

$$FFN(x) = \max(0, xW_1 + b_1)W_2 + b_2 \quad (14)$$

где $W_1 \in R^{d_{model} * d_{ff}}$, $W_2 \in R^{d_{ff} * d_{model}}$ – матрицы весов, b_1, b_2 – векторы смещений, $\max(0, \cdot)$ – функция активации ReLU.

Результат FNN слоя далее подается на слой нормализации и остаточное соединение (residual connection):

$$y = \text{LayerNorm}(x + FFN(x)) \quad (15)$$

Это способствует стабилизации обучения и ускоряет сходимость сети за счет уменьшения затухания и взрывов градиентов.

Для комплексного анализа применяемых в исследовании моделей была проведена их сравнительная характеристика в таблице 4 по ключевым параметрам: типу входных данных, архитектуре, вычислительной сложности, интерпретируемости и типичным областям применения. Сравнение позволяет выявить различия в принципах работы и преимущества каждой из моделей при решении задач прогнозирования, классификации и рекомендаций.

Таблица 4 – Сравнительная характеристика моделей FCNN, LightGBM, FNN

Критерий	FCNN	LightGBM	FNN
Тип Модели	Глубокая нейронная сеть	Градиентный бустинг на решающих деревьях	Компонент архитектуры Transformer
Основной принцип работы	Линейное преобразование входов с нелинейной активацией и обучением по градиенту	Итеративное добавление деревьев, минимизирующих ошибку предыдущих моделей	Применение двухслойной ReLU-сети к каждому элементу последовательности
Тип входных данных	Векторные признаки фиксированной размерности	Табличные данные, числовые и категориальные признаки	Эмбединги из слоя внимания (вектор представления токена)
Функция потерь	MSE, Cross-Entropy	LogLoss, MSE, Custom	Cross-Entropy
Метод оптимизации	Стохастический градиентный спуск	Newton/Gradient boosting	Adam / AdamW

Нелинейность	Сигмоида, ReLU, tanh	Piecewise constant (в узлах деревьев)	ReLU
Преимущества	Простота реализации, универсальность, гибкость	Высокая скорость и точность, интерпретируемость	Повышает выразительность и стабильность трансформеров
Недостатки	Склонность к переобучению, много параметров	Зависимость от гиперпараметров, необходимость тюнинга	Высокие требования к ресурсам и объёму данных
Типичные области применения	Классификация, регрессия, анализ сигналов	Предсказание, скоринг, задачи табличных данных	Обработка естественного языка, изображений и аудио

Ввиду результатов, представленных в таблице 4, целесообразно подробно изложить методологию сбора и предобработки данных, поскольку именно качество исходного корпуса и принятые решения на этапах очистки и формирования признаков определяют воспроизводимость эксперимента и интерпретируемость полученных различий в поведении моделей.

Сбор и предварительная обработка данных

Для формирования базы данных о компаниях и услугах в г. Бишкек использовались комбинированные методы получения информации. В ходе разработки проекта (март 2025 г.) использовался бесплатный доступ к API сервиса 2GIS для получения данных о компаниях [18], а также веб-скрапинг.

С помощью API были извлечены базовые сведения: название компании, рубрика, категория, адрес и уникальный идентификатор. В результате было собрано около 8000 записей, сохраненных в отдельном файле.

Поскольку интерфейс API не предоставляет контактных данных, рейтингов, отзывов и координат, для их извлечения был реализован дополнительный модуль парсинга с использованием Selenium WebDriver. В процессе обработки каждая страница компании загружалась динамически, после чего с применением WebDriverWait извлекались недостающие атрибуты: телефон, рейтинг, количество отзывов, широта и долгота. Результирующие данные автоматически объединялись с исходными по идентификатору компании и сохранялись в другом файл.

На этапе предобработки данные были очищены от текстовых ошибок и пропусков, заменённых на *NaN*, а числовые признаки («Рейтинг», «Широта», «Долгота») приведены к единому формату с помощью `pd.to_numeric()`. Обновлённый набор сохранён в csv-файле.

Для дальнейшего анализа выполнялась векторизация текстовых признаков (названия компаний, рубрики и категории) с использованием многоязычной модели SentenceTransformer “distiluse-base-multilingual-cased-v1”, а также One-Hot Encoding категориальных данных с помощью библиотеки *scikit-learn*. Итоговый векторизованный набор признаков также сохранён в отдельном файле. Так как реальные пользовательские данные отсутствовали, была сформирована *синтетическая выборка* из 20 000 пользователей с полями: `user_id`, `nickname`, `gender`, `age`, `latitude`, `longitude`, `interests`. Для векторного представления интересов применялась та же модель SentenceTransformer, усредняющая эмбединги интересов до 512-мерного вектора.

Для моделирования пользовательской активности были сгенерированы взаимодействия пользователей с компаниями, в которых оценка релевантности (*score*) рассчитывалась по формуле (16):

$$score = 0.5 \cdot similarity + 0.3 \cdot norm_rating + 0.2 \cdot engagement - geo_penalty \quad (16)$$

где учитывались семантическое сходство интересов и рубрики, географическая удалённость, рейтинг и активность пользователей.

Эмбединги пользователей и компаний, сохранённые в строковом формате, преобразовывались в числовые массивы с помощью `ast.literal_eval()` и разворачивались в отдельные признаки.

После объединения таблиц `users`, `companies` и `interactions` по ключам `user_id` и `company_id` были выделены только релевантные признаки и целевая переменная (`label`). Для унификации масштабов признаков применялся *StandardScaler* [19], обучение которого выполнялось пакетно для оптимизации памяти. Полученный корпус данных и процедуры предобработки обеспечивают основу для обучения и сравнения моделей, о которых идет речь в следующем разделе: далее описано построение, настройка FCNN, LightGBM и Transformer-FNN на подготовленном датасете.

Обучение моделей на готовых данных

Для сравнительного анализа все модели обучались на одной и той же подготовленной выборке, которая была описана выше, и оценивались по общему набору метрик (`precision`, `recall`, `F1`, а также ранговые критерии при необходимости). В ходе экспериментов применялись как временные схемы валидации, так и стратифицированные разбиения в зависимости от наличия временных меток; для получения устойчивых оценок каждое обучение повторялось несколько раз с фиксированными `seed`-ами, а итоговые показатели приводятся как средние значения с оценкой вариативности.

Полносвязная нейронная сеть (FCNN) реализовывалась на базе TensorFlow/Keras как многослойный персептрон с двумя скрытыми слоями (64 и 32 нейрона), слоями `BatchNormalization` и `Dropout` для регуляризации и выходным сигмоидным нейроном в задаче бинарной классификации. Обучение проводилось стохастическим градиентным спуском с адаптивным оптимизатором (Adam), использовались батчи и ранняя остановка; в задачах с несбалансированными метками применялось взвешивание классов в функции потерь. Архитектура FCNN использовалась как «универсальный» нелинейный аппроксиматор при оценке способности извлекать сложные зависимости из объединённого признакового пространства.

В качестве табличного бенчмарка использовался LightGBM – алгоритм градиентного бустинга на решающих деревьях. Для LightGBM выполнен автоматизированный подбор гиперпараметров: первичный `randomized search` по набору параметров (`num_leaves`, `max_depth`, `learning_rate`, `n_estimators`, `feature_fraction`, `bagging_fraction`, `min_data_in_leaf`, `lambda_l1`, `lambda_l2`), с последующей локальной оптимизацией и ранней остановкой по валидационной метрике (`early_stopping`). Обучение LightGBM в эксперименте проводилось с кросс-валидацией (3-fold) и логированием лучших конфигураций; полученная модель показала наилучший баланс `precision` и `recall` (`precision` \approx 0.91, `recall` \approx 0.87, `F1` \approx 0.89) и была сохранена для дальнейшей интеграции.

Transformer-FNN реализован как позиционно-независимая `feedforward`-сеть, применяемая к эмбедингам компаний и пользователей; в обучении использовались взвешенные кросс-энтропийные функции потерь (для компенсации дисбаланса классов), оптимизатор Adam с `learning rate` = 0.001 и обучение в течение фиксированного числа эпох (в работе – 15 эпох). Тренировочный цикл реализован стандартно: прогон батчей, вычисление `loss`, обратное распространение и шаг оптимизатора; результаты показали удовлетворительную точность при относительно низкой полноте по сравнению с LightGBM.

После завершения этапа обучения моделей и их тонкой настройки на подготовленных наборах данных следующей необходимой стадией является их количественная оценка и сопоставление по заранее выбранным критериям. В качестве процедур валидации применялась многократная кросс-валидация, а в качестве основных метрик – `precision`, `recall` и `F1`, позволяющие сбалансированно оценить точность и полноту рекомендаций (таблица 6).

Таблица 6 — Сравнение моделей по метрикам качества

Метрика	FCNN	LightGBM	Transformer FNN
precision	0.76	0.91	0.87
recall	0.99	0.87	0.76
f1-score	0.86	0.89	0.81

Выбор наилучшей модели

Проведенный анализ показателей качества моделей на задаче рекомендательной системы (таблица 6), в рамках которого были рассмотрены и протестированы несколько подходов на основе эмбедингов интересов, характеристик компаний и оценок взаимодействий, показал, что градиентный бустинг LightGBM обеспечивает оптимальный баланс между точностью (precision) и полнотой (recall), демонстрируя стабильные и высокие результаты по F1-мере. Так, модель достигает precision = 0.91 и recall = 0.87, что соответствует $F1 \approx 0.89$ по результатам кросс-валидации. Градиентный бустинг, как подход, считается одним из наиболее эффективных методов для работы с табличными данными [15].

LightGBM обладает рядом преимуществ, которые определили её выбор для дальнейшего применения в рекомендательной системе. Прежде всего модель характеризуется высокой скоростью обучения и хорошей масштабируемостью. Это достигается благодаря использованию гистограммного алгоритма построения деревьев, который дискретизирует непрерывные признаки на интервалы (бины), что существенно снижает вычислительные затраты и потребление памяти. Подобная архитектура позволяет эффективно обучать модель на больших объемах данных высокой размерности.

Кроме того, LightGBM демонстрирует сбалансированные показатели точности и полноты благодаря использованию регуляризационных параметров (max_depth, min_data_in_leaf), регулирующих сложность деревьев и предотвращающих переобучение. Существенным преимуществом также является интерпретируемость – встроенные механизмы анализа важности признаков (feature importance) позволяют определить вклад каждого параметра в итоговое предсказание, что делает процесс оптимизации модели более прозрачным и осмысленным.

Важной особенностью является способность модели к инкрементальному обучению: при поступлении новых данных модель можно дообучать без необходимости полного переобучения. Это свойство обеспечивает адаптивность и устойчивость модели в условиях динамически обновляемых данных.

Таким образом, выбранная модель сочетает в себе вычислительную эффективность, устойчивость к переобучению, высокий уровень интерпретируемости и стабильные показатели качества. Эти характеристики делают данную модель оптимальным выбором для построения рекомендательной системы, обеспечивая надежное качество предсказаний и возможность масштабирования при росте объемов данных.

Интеграция модели в веб-приложение

В качестве завершающего этапа исследования реализована интеграция выбранной рекомендательной модели в веб-сервис, обеспечившая практическую применимость полученных результатов. На серверной стороне развернут API-слой, принимающий пользовательские запросы, извлекающий профиль пользователя и формирующий входной вектор признаков для модели. В производственной цепочке воспроизводятся ключевые этапы предобработки: преобразование текстовых описаний в векторные представления

(эмбединги), объединение семантических и табличных признаков, вычисление пространственно-временных характеристик и нормализация данных перед подачей в модель. Предсказания релевантности выполняются обученной моделью; результаты ранжируются и возвращаются клиенту в структурированном формате (рис. 5).

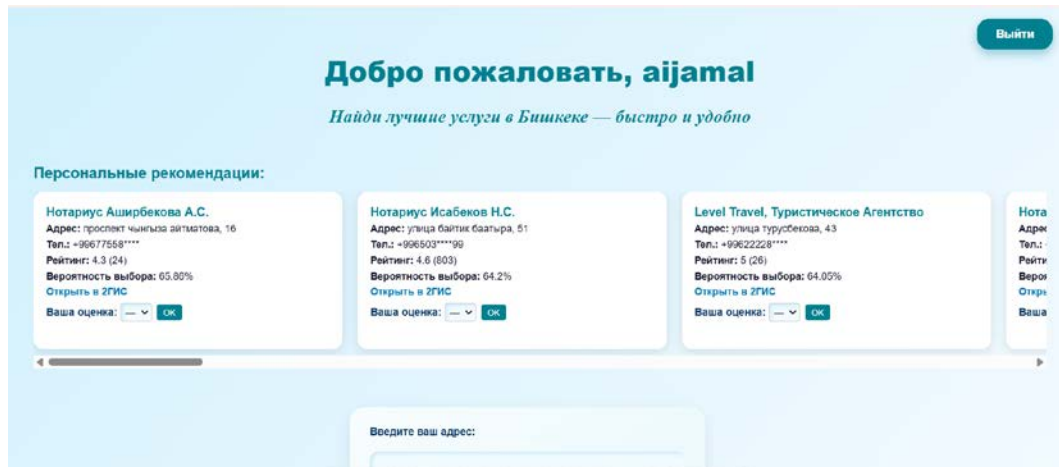


Рисунок 5 – Представление предсказаний релевантности рекомендаций пользователям

Клиентская часть обеспечивает интерактивное представление рекомендаций и сбор обратной связи. Взаимодействия пользователей (оценки, скрытие объектов и др.) фиксируются и используются для адаптации профилей и последующего улучшения выдачи, что позволяет реализовать механизм онлайн-адаптации без необходимости полного переобучения модели.

При тестировании особое внимание уделялось корректности логики формирования выдачи, устойчивости работы при частично отсутствующих данных и адекватности результатов в сценариях «новый пользователь» и «повторный пользователь». Валидация подтвердило работоспособность прототипа в условиях, приближенных к прикладным задачам, и показала готовность архитектуры к дальнейшему масштабированию при добавлении стандартных инженерных компонентов (кэширование, фоновые очереди задач, мониторинг качества).

Примененная архитектура носит характер feature-level гибридной рекомендательной системы: на этапе формирования признаков объединены контентные (текстовые эмбединги), пространственно-временные и коллаборативные сигналы. LightGBM функционирует в роли обучаемого ранжировщика, предсказывающего вероятность релевантности и осуществляющего окончательное фильтрование кандидатов; такой подход позволяет интегрировать преимущества различных методов при использовании единого скорера.

Внедрение модели в виде веб-сервиса завершает исследовательский цикл, представленный в настоящей работе – от обзора и формализации методов и моделей до их прикладной реализации. Полученный прототип демонстрирует корректность и устойчивость выдачи рекомендаций в интерактивном пользовательском окружении и одновременно выявляет прикладные требования к масштабированию и поддержанию качества, которые следует учесть в дальнейших инженерных доработках. Схематические иллюстрации интерфейса и примеры работы системы приведены в Приложениях.

Заключение

Проведённое исследование позволило комплексно рассмотреть методы и технологии проектирования рекомендательных систем для городской предметной области с учётом пространственно-временных факторов и контекстных характеристик пользователей [20]. В результате выполненного анализа установлено, что традиционные коллаборативные и контентные подходы, несмотря на высокую популярность, оказываются недостаточными для

задач локальных сервисов, где ключевую роль играют географическая привязка, время активности и поведенческие особенности пользователей.

Учитывая выявленные ограничения классических парадигм, в работе разработана гибридная архитектура, которая объединяет преимущества контентной, коллаборативной и контекстной фильтрации на уровне признакового описания (feature-level hybrid). В качестве модельного ранжировщика применён LightGBM, обеспечивающий высокую скорость обучения, устойчивость к разреженности данных и интерпретируемость результатов; включение семантических, пространственных и временных признаков в единое обучаемое пространство позволило повысить качество персонализированных рекомендаций по сравнению с базовыми моделями, что подтверждено экспериментальными испытаниями.

Для подтверждения практической применимости предложенного подхода проведена реализация веб-прототипа и соответствующая интеграция модели в веб-среду. Прототип обеспечивает генерацию и ранжирование рекомендаций, сбор обратной связи и онлайн-адаптацию пользовательских профилей, что демонстрирует возможность использования разработанных методов в реальных пользовательских сценариях городской среды.

На основе полученных результатов сделан вывод о том, что успешная реализация рекомендательной системы требует комплексного подхода, включающего не только выбор алгоритмической модели, но и тщательную организацию данных, инженерную инфраструктуру и механизмы непрерывного обновления. Разработанный прототип служит базой для дальнейшего расширения функциональности — в частности для внедрения более сложных нейросетевых компонентов, масштабирования системы на новые регионы и типы сервисов, а также для интеграции с мобильными платформами.

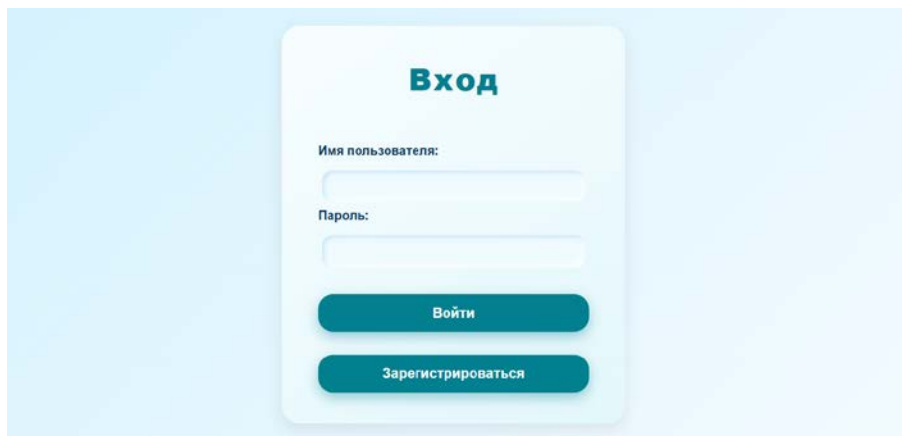
В перспективе целесообразно сосредоточить внимание на развитии методов адаптивного обучения, устойчивых к постепенным изменениям характеристик данных и динамике пользовательского поведения, а также на включении механизмов оценки справедливости и прозрачности рекомендаций. Продолжение исследований в этом направлении будет способствовать повышению качества, надёжности и релевантности рекомендательных систем в цифровой городской инфраструктуре.

Литература

1. Джейд Картер, *Нейросети начало*, Глава 7, 2023. (Дата обращения: 17.09.2025)
2. Научная статья Козловой и Германчук (2022) «Разработка гибридной системы рекомендаций» <https://cyberleninka.ru/article/n/razrabotka-gibridnoy-sistemy-rekomendatsiy> (Дата обращения: 17.09.2025)
3. Научная статья Ерёмина и Моркулёва (2023) «Методы реализации гибридных рекомендательных систем» <https://cyberleninka.ru/article/n/metody-realizatsii-gibridnyh-rekomendatelnyh-sistem> (Дата обращения: 17.09.2025)
4. Youtube <https://www.youtube.com> (Дата обращения: 17.09.2025)
5. LinkedIn <https://www.linkedin.com> (Дата обращения: 17.09.2025)
6. Lalafo <https://lalafo.kg> (Дата обращения: 17.09.2025)
7. Berry M.W. Large scale singular value computations / International Journal of Supercomputer Applications, 1992. С. 13–49. (Дата обращения: 17.09.2025)
8. Billsus D., Pazzani M.J. Learning Collaborative Information Filters / Proceeding 15th International Conference on Machine Learning, 1998. С. 46-54. (Дата обращения: 19.09.2025)
9. Магистерская диссертация Кокачева В. А. <https://api.dspace.spbu.ru/server/api/core/bitstreams/f8aec00e-3f20-4265-87dd-a734a6066263/content> (Дата обращения: 19.09.2025)
10. Ekstrand M. D., Riedl J. T., Konstan J. A. Collaborative Filtering Recommender Systems / Foundations and Trends in Human–Computer Interaction, 2011. No. 2. С. 81-173. (Дата обращения: 21.09.2025)

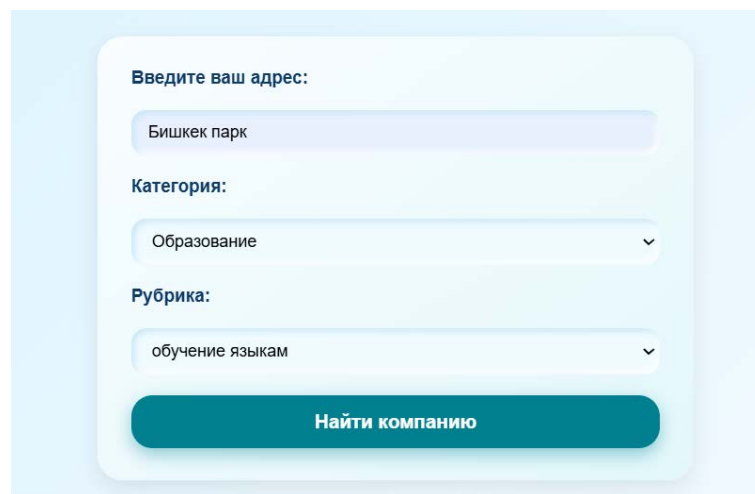
11. Введение в рекомендательные системы, Яндекс образование, <https://education.yandex.ru/handbook/ml/article/intro-recsys> (Дата обращения: 21.09.2025)
12. Kermany, N. R., & Alizadeh, S. H. (2017). A hybrid multi-criteria recommender system using ontology and neuro-fuzzy techniques. *Electronic Commerce Research and Applications*, 21, 50-64 (Дата обращения: 27.09.2025).
13. Mohammadpour, T., Bidgoli, A. M., Enayatifar, R., & Javadi, H. H. S. (2019). Efficient clustering in collaborative filtering recommender system: Hybrid method based on genetic algorithm and gravitational emulation local search algorithm. *Genomics*, 111(6), 1902-1912 (Дата обращения: 27.09.2025).
14. Goodfellow I., Bengio Y., Courville A. *Deep Learning* / MIT Press, 2016. С. 1–800. (Дата обращения: 2.10.2025).
15. Ke G., Meng Q., Finley T., Wang T., Chen W., Ma W., Ye Q., Liu T.-Y. *LightGBM: A Highly Efficient Gradient Boosting Decision Tree* / *Proceedings of the 31st Conference on Neural Information Processing Systems USA*, 2017. С. 3149–3157 (Дата обращения: 2.10.2025)
16. Vaswani A., Shazeer N., Parmar N., Uszkoreit J., Jones L., Gomez A. N., Kaiser Ł., Polosukhin I. *Attention Is All You Need* / *Proceedings of the 31st Conference on Neural Information Processing Systems USA*, 2017. С. 5998–6008 (Дата обращения: 5.10.2025).
17. Kula M. *Metadata Embeddings for User and Item Cold-start Recommendations* / *Proceedings of the 2nd Workshop on New Trends on Content-Based Recommender Systems (co-located with RecSys 2015) Vol. 1448, Vienna, Austria, 2015*. С. 14–21. (Дата обращения: 5.10.2025).
18. 2GIS API. – Официальная документация
19. <https://dev.2gis.ru/order> (дата обращения: 12.03.2025)
20. DataFinder. StandardScaler в машинном обучении <https://datafinder.ru/products/standardscaler-v-mashinnom-obuchenii> (дата обращения: 10.10.2025).
21. Верзунов, С. Н. Анализ и ARIMA-модели динамики изменения концентрации PM_{2.5} в атмосферном воздухе Г.Бишкек / С. Н. Верзунов, Н. М. Лыченко // *Проблемы автоматизации и управления*. – 2019. – № 1(36). – С. 147-155. – DOI 10.5281/zenodo.3253027. – EDN IKKKUC.

ПРИЛОЖЕНИЕ 1



Приложение 1, рисунок 1 – Вход пользователя в систему

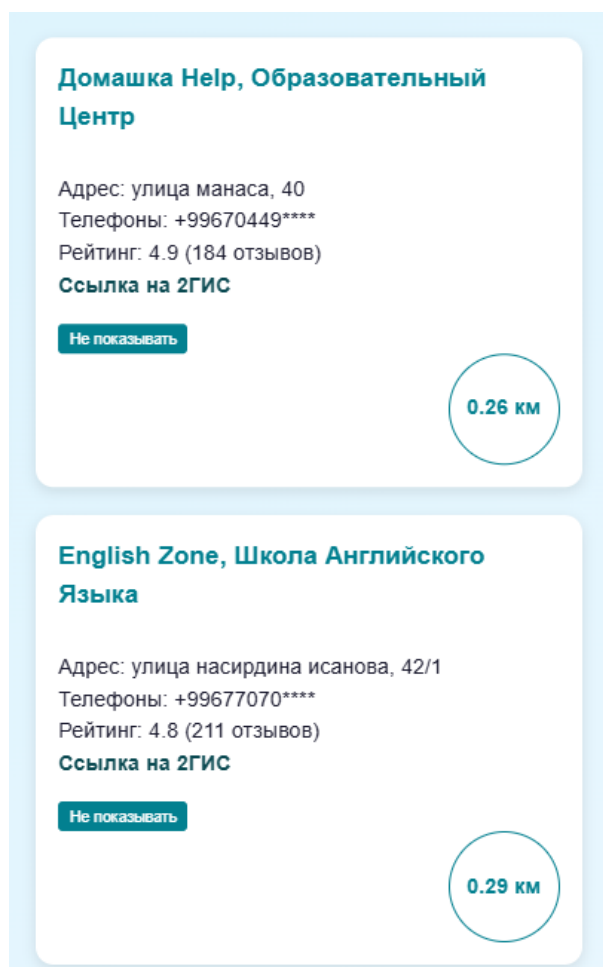
На рисунке 1 приложения 1 представлена страница входа в систему, обеспечивающая возможность регистрации и аутентификации пользователей.



The screenshot shows a search form with a light blue background. It contains three input fields: 'Введите ваш адрес:' with the text 'Бишкек парк', 'Категория:' with a dropdown menu showing 'Образование', and 'Рубрика:' with a dropdown menu showing 'обучение языкам'. Below these fields is a large teal button labeled 'Найти компанию'.

Приложение 1, рисунок 2 – Форма для поиска

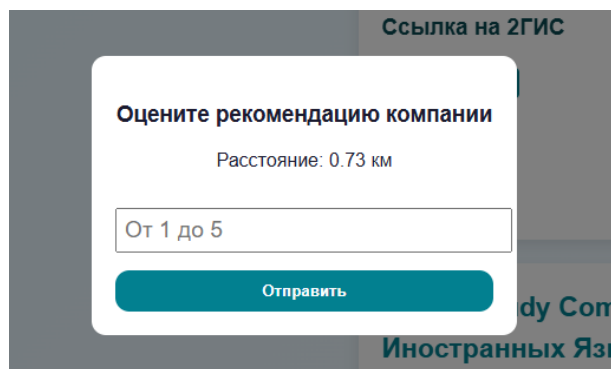
После выполнения процедуры входа пользователь получает доступ к интерфейсу поиска и может заполнить соответствующую форму для отбора организаций по категориям, рубрикам и адресу (приложение 1, рисунок 2), что позволяет сократить пространство поиска и повысить релевантность выдачи.



The screenshot displays two search results. The first result is for 'Домашка Help, Образовательный Центр' with address 'улица манаса, 40', phone '+99670449****', rating '4.9 (184 отзывов)', and a 'Ссылка на 2ГИС'. It includes a 'Не показывать' button and a distance indicator of '0.26 км'. The second result is for 'English Zone, Школа Английского Языка' with address 'улица насирдина исанова, 42/1', phone '+99677070****', rating '4.8 (211 отзывов)', and a 'Ссылка на 2ГИС'. It also includes a 'Не показывать' button and a distance indicator of '0.29 км'.

Приложение 1, рисунок 3 – Результаты поиска

Результаты поиска отображаются в таком виде: название, адрес, номер телефона, рейтинг, ссылка на местоположение компании на карте и расстояние от текущего адреса пользователя до самой компании (приложение 1, рисунок 3). Точное местоположение компании можно посмотреть на карте, перейдя по ссылке.



Приложение 1, рисунок 4 – Окно для оценки рекомендации

После ознакомления с карточкой компании (приложение 1, рисунок 4) пользователю предлагается выставить оценку данной организации. Полученная оценка фиксируется в базе данных и рассматривается как сигнал предпочтения пользователя, используемый при дальнейшей персонализации. В последующем рекомендательная система использует эти сохранённые взаимодействия и высокие оценки при формировании рекомендаций, отдавая приоритет компаниям, с которыми пользователь ранее взаимодействовал и которым он присвоил высокие оценки.