МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
**«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»**
Кафедра информатики и программирования

СОЗДАНИЕ РЕКОМЕНДАТЕЛЬНОЙ СИСТЕМЫ ДЛЯ ПРИЛОЖЕНИЯ ЗНАКОМСТВ

БАКАЛАВРСКАЯ РАБОТА

студента 4 курса 441 группы
направления 02.03.03 — Математическое обеспечение и администрирование информационных систем
факультета компьютерных наук и информационных технологий
Уталиева Султана Едильбаевича

Научный руководитель
ст.преп. кафедры ИиП _____ А. А. Казачкова

Заведующий кафедрой
к. ф.-м. н., доцент _____ Огнева М. В.

Саратов 2025

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Современные мобильные приложения всё чаще стремятся к персонализации пользовательского опыта. Одной из ключевых технологий в этой области являются рекомендательные системы, позволяющие адаптировать контент под индивидуальные предпочтения пользователей. Особенно важны такие системы в приложениях для знакомств, где точность рекомендаций напрямую влияет на успешность взаимодействий между пользователями.

В данной работе рассматривается задача построения рекомендательной системы для мобильного приложения знакомств. В отличие от большинства существующих решений, рекомендации в предлагаемой системе формируются не только на основе очевидных анкетных данных, но и на основе тестов-опросов, которые пользователь может пройти по желанию. Каждая карточка с вопросом предоставляет возможность ответить «да», «нет» или пропустить — что позволяет формировать тернарные признаки $(-1, 0, 1)$, лежащие в основе построения профиля предпочтений.

Основной целью дипломной работы является разработка рекомендательной системы, обеспечивающей релевантные и разнообразные рекомендации потенциальных партнеров на основе результатов тестирования. Для достижения этой цели решаются следующие задачи:

- анализ существующих подходов к построению рекомендательных систем в контексте приложений знакомств;
- формализация задачи рекомендаций с учетом специфики представления пользовательских данных;
- проектирование и реализация микросервисной архитектуры рекомендательной системы;
- исследование и разработка методов рекомендаций на основе коллаборативной фильтрации, эвристик и кластеризации;
- разработка подхода к оценке качества рекомендаций с использованием оффлайн-метрик и пользовательской обратной связи.

С точки зрения архитектуры, система реализована в виде набора взаимодействующих компонентов: мобильное приложение на Flutter, серверная часть на Java с использованием фреймворка Spring и отдельный рекомендательный микросервис на Python. Такое разделение позволяет гибко развивать систему и экспериментировать с алгоритмами без влияния на основной функционал

приложения.

Практическая значимость работы заключается в создании масштабируемого и расширяемого рекомендательного решения, которое может быть интегрировано в реальные приложения. Использование легких и интерпретируемых моделей делает систему доступной для внедрения даже в условиях ограниченных вычислительных ресурсов.

1 Анализ предметной области и существующих решений

1.1 Особенности рекомендательных систем

Рекомендательные системы (РС) являются неотъемлемой частью современных цифровых платформ, предоставляя пользователям персонализированные предложения продуктов, услуг или контента. Они находят широкое применение в различных областях, включая электронную коммерцию, стриминговые сервисы, социальные сети и онлайн-знакомства.

Основные функции рекомендательных систем включают:

— отсутствие информации о новых пользователях и объектах (проблема холодного старта);
— высокая разреженность пользовательско-объектных матриц;
— необходимость обеспечения справедливости и отсутствия предвзятости в рекомендациях [1].

Современные исследования направлены на преодоление этих ограничений, в том числе с использованием больших языковых моделей, методов объяснимого машинного обучения и расширения пользовательского контекста [2].

1.2 Обзор рекомендательных систем в сфере онлайн-знакомств

Онлайн-сервисы знакомств предъявляют особые требования к алгоритмам рекомендаций. Поскольку конечной целью является установление реального или виртуального контакта между людьми, системы должны максимально точно учитывать совместимость по широкому спектру признаков, при этом оставаясь достаточно лёгкими в вычислении и объяснимыми.

1.2.1 OkCupid

OkCupid начала с текстовых анкет и развила модель расчёта совместимости через детальные опросы. Пользователю предлагается до 400 вопросов, ответы на которые кодируются в вектор $q_i \in \{-1, 0, 1\}$, где значения означают «против», «безразлично» и «за» [8]. Система вычисляет попарную корреляцию ответов двух пользователей:

$$\mathrm{compat}(u, v) = \frac{\sum_i w_i \, \mathbf{1}(q_{u,i} = q_{v,i})}{\sum_i w_i},$$

где w_i — вес вопроса (определяется самим пользователем). Дальнейшее ранжирование учитывает активность и исторические данные о переписках. Такой

гибридный подход сочетает content- и collaborative-элементы и позволяет получать качественные рекомендации даже при отсутствии большого числа лайков или свайпов.

1.2.2 Tinder

Tinder применяет модель *reciprocal recommendation*: предлагаются только такие профили, которые с наибольшей вероятностью ответят взаимностью [9]. Основные компоненты алгоритма:

- базовый *рейтинговый Эло-подобный механизм*, где свайпы вправо увеличивают рейтинг профиля, влево уменьшают;
- измерение *скорости реакции* и *частоты использования* для оценки «живости» аккаунта;
- фильтрация неактивных и спорных профилей, чтобы не портить пользовательский опыт.

Исследования показывают, что такие меры уменьшают количество односторонних свайпов и повышают вероятность установления диалога между пользователями [10].

1.2.3 Bumble и Hinge

Bumble унаследовал от Tinder большинство алгоритмов, но ввёл правило, что первым сообщение отправляет только женщина. Это влияет на ранжирование: при прочих равных предпочтение отдается тем, кто реже получает безответные лайки и чаще инициирует общение [11]. Hinge выделяется принципом «создания общего контекста»: анализируются общие Facebook-друзья, интересы и местоположение. Рекомендации формируются через графовую модель, где вершины — пользователи, а рёбра взвешены по степени близости в социальных связях и ответам на три вступительных вопроса в профиле [12]. Это позволяет сегментировать рынок знакомств на локальные сообщества и повышает релевантность для тех, кто ищет знакомых «по кругу общения».

1.2.4 eHarmony

eHarmony фокусируется на психологической совместимости. Новый пользователь заполняет обширный личностный опросник, после чего строится профиль в латентном пространстве через метод факторизации матрицы сходств по тестовым шкалам (MMPI и Big Five) [13]. Дальнейшие рекомендации приме-

няют content-модуль для расширения круга потенциальных партнёров за счёт схожести по интересам и ценностям.

1.2.5 Coffee Meets Bagel и другие

Coffee Meets Bagel ежедневно генерирует «порции» (bagels) — ограниченный набор рекомендаций, отбираемых на основе коллаборативной фильтрации и эвристики «совместимых интересов» (аналог OkCupid, но с жёстким ограничением на количество предложений) [14]. Такой подход направлен на снижение перегрузки выбором и повышение качества каждого матча. Другие сервисы, например Plenty of Fish и Hily, экспериментируют с машинным зрением (анализ фотографий) и NLP (анализ биографий) для дополнительной фильтрации.

1.2.6 Общие тенденции

Современные платформы склоняются к гибридным решениям, объединяющим:

- анализ анкетных данных и опросов (content);
- collaborative filtering и reciprocal recommendation;
- эвристики активности и социального контекста;
- технологические новации (GNN, NLP, CV) для более глубокой семантической обработки.

Таким образом удаётся достичь баланса между качеством рекомендаций и вычислительными ресурсами, что особенно важно для мобильных приложений с высокой нагрузкой.

1.3 Особенности сбора и представления пользовательских признаков

В большинстве современных онлайн-сервисов знакомств информация о предпочтениях пользователей строится на основе заранее подготовленных анкет и профилей. При этом традиционные текстовые опросники, как в OkCupid или eHarmony, обеспечивают глубокий анализ личности, но требуют значительного времени от пользователя и зачастую приводят к высокой доле отказов от заполнения. С другой стороны, модели, ориентированные на простой свайп-интерфейс (Tinder, Bumble), существенно упрощают взаимодействие, однако теряют информацию о степени заинтересованности: бинарный выбор «лайк» или «дизлайк» не отражает нюансы отношения и нейтральные позиции.

В предлагаемом подходе используется гибридная модель ввода данных: каждый вопрос теста представлен карточкой, которую пользователь может смахнуть вправо (положительный ответ), влево (отрицательный ответ) или вверх в случае пропуска. Такая схема объединяет скорость и простоту свайпа с более тонкой градацией отклика: положительное, отрицательное и нейтральное отношение. Внутри микросервиса ответы кодируются как тернарные признаки $q_i \in \{-1, 0, 1\}$, где 1 соответствует явному согласию, -1 отражает негативную реакцию, а 0 обозначает отсутствие чёткой позиции.

Технически процесс сбора строится следующим образом. На клиентской стороне Flutter-приложение загружает набор вопросов с уникальными идентификаторами и последовательно отображает карточки. Каждое действие пользователя сохраняется в буфере и по завершении теста передаётся на бэкенд через REST-запрос. В Python-микросервисе данные проходят валидацию и попадают в таблицу $\mathrm{user\_responses}$, где каждой паре (user_id, question_id) соответствует значение q_i. При этом фиксируются следующие метаданные: время начала и окончания прохождения теста, число пропущенных вопросов, а также количество повторных попыток. Эти параметры могут служить индикаторами уверенности пользователя в ответах и учитываться в дальнейшей калибровке весов признаков.

Для подготовки к построению рекомендаций ответы разных пользователей формируются в разреженные матрицы по каждому тесту. Элемент матрицы принимает значение $q_{u,i}$ или остаётся пустым (null), если пользователь не проходил конкретный вопрос. При вычислении метрик сходства между векторами профилей применяется нормировка по числу ненулевых элементов: в зависимости от метода это обеспечивает сравнимость результатов для пользователей, прошедших различное количество вопросов. В ряде экспериментов рассматривались методы заполнения пропусков «по умолчанию» нулём, а также замена на среднее значение по вопросу, что позволило оценить влияние стратегий импутации на качество рекомендаций.

Наконец, предусмотрено расширение тернарного вектора демографическими признаками. Пол пользователя кодируется в виде бинарной переменной, возраст — нормируется и входит в общий профиль как вещественный признак. В дальнейшем возможно дополнение текстовым описанием «О себе», представляемым в виде эмбеддингов, извлечённых трансформерными моделями. Однако

основной упор сделан на именно тернарные ответы, так как они обеспечивают наиболее компактный и одновременно выразительный способ кодирования предпочтений при минимальной нагрузке на пользователя.

1.4 Выводы и обоснование выбранного подхода

Анализ существующих решений в сегменте онлайн-знакомств выявил два крайних сценария: масштабные анкеты с высоким уровнем детализации и простые бинарные свайпы, удобные, но поверхностные. Ни один из этих подходов не совмещает одновременно удобство и глубину представления пользовательских предпочтений.

Карточно-свайповая модель с тремя вариантами ответа позволяет добиться компромисса. Во-первых, она не предъявляет к пользователю чрезмерных требований по времени и вниманию: одно движение пальцем на экране занимает доли секунды. Во-вторых, введение нейтрального варианта (пропуск) снимает двоичную жёсткость решений, фиксируя менее определённые реакции, которые в дальнейшем помогают точнее выстраивать профиль. Такой подход сходен по механике с Tinder, но обеспечивает информативность, сравнимую с результатами детальных анкет.

Представление данных в виде тернарного вектора позволяет использовать широкий набор методов рекомендаций. Коллаборативная фильтрация на основе SVD легко адаптируется к трем уровням предпочтений, а кластеризация по таким векторам выявляет группы пользователей с близкими моделями отклика. Демографическая фильтрация, применяемая на предварительном этапе, снижает объём вычислений и повышает релевантность кандидатов, исключая явно несовместимые пары по половому и возрастному признаку.

Таким образом, тернарная модель ответов обеспечивает баланс между насыщенностью представления и лёгкостью сбора данных. Именно она лежит в основе предлагаемой архитектуры микросервиса, обеспечивая достаточный объём информации для точных и объяснимых рекомендаций без лишней нагрузки на пользователя. Этот вывод служит отправной точкой для выбора методов CF, кластеризации и эвристик, подробно описываемых в следующих главах.

2 Методы построения рекомендательной системы

2.1 Коллаборативная фильтрация

Коллаборативная фильтрация является одним из наиболее популярных подходов в рекомендательных системах, особенно в условиях, когда отсутствует явное описание объектов или пользователей. Основная идея заключается в том, что предпочтения пользователей могут быть предсказаны на основе поведения других пользователей с похожими вкусами.

Существует два основных типа коллаборативной фильтрации: на основе памяти (*memory-based*) и на основе модели (*model-based*). Первый подход использует метрики сходства между пользователями или объектами (например, косинусное расстояние, корреляцию Пирсона) и агрегирует оценки соседей. Второй — строит параметризованную модель на основе данных о взаимодействиях, чаще всего через факторизацию матрицы.

Пусть имеется матрица взаимодействий $R \in \mathbb{R}^{m \times n}$, где m — количество пользователей, n — количество объектов (например, анкет потенциальных партнёров). Элемент $r_{u,i}$ может обозначать бинарную оценку (лайк/не лайк), числовой рейтинг или иной сигнал предпочтения. Коллаборативная фильтрация предполагает, что в матрице присутствует скрытая структура, отражающая закономерности во вкусах пользователей.

Одним из распространённых методов является сингулярное разложение (SVD) или его модификации (например, Funk-SVD, ALS), при которых R аппроксимируется как

$$R \approx UV^\top,$$

где $U \in \mathbb{R}^{m \times k}$ и $V \in \mathbb{R}^{n \times k}$ содержат латентные векторы пользователей и объектов соответственно, а k — число латентных признаков. Значение $\hat{r}_{u,i} = U_u \cdot V_i^\top$ интерпретируется как предсказанная степень интереса пользователя u к объекту i.

Интересным обобщением является применение коллаборативной фильтрации к данным не только о лайках, но и о признаках, таких как ответы пользователей на опросы. В этом случае каждый пользователь представлен тернарным или категориальным вектором, а матрица $Q \in \{-1, 0, 1\}^{m \times p}$ (где p — число вопросов) также может быть факторизована аналогичным способом:

$$Q \approx U'Z^\top.$$

Полученные векторы могут использоваться для оценки схожести между пользователями, для восстановления пропущенных ответов или как источник признаков для гибридных моделей.

Коллаборативная фильтрация демонстрирует высокую эффективность при наличии большого количества пользовательских взаимодействий, однако страдает от проблемы холодного старта (new user/item problem) и может усиливать популярность одних и тех же объектов, снижая разнообразие рекомендаций.

2.2 Эвристики: совпадения по ответам, популярность, фильтры

Эвристические методы в рекомендательных системах основаны на наборе простых правил и предположений, позволяющих быстро и эффективно формировать рекомендации. Несмотря на относительную простоту, такие подходы остаются актуальными, особенно в условиях ограниченности данных или требований к объяснимости.

Одним из базовых эвристических подходов является сравнение пользователей по их ответам на вопросы анкет или опросов. Если ответы представлены в тернарной шкале (например, -1 — несогласие, 0 — нейтрально, 1 — согласие), то схожесть между пользователями можно оценить по доле совпадающих ответов:

$$\mathrm{sim}(u, v) = \frac{1}{|P|} \sum_{i \in P} \mathbf{1}(q_{u,i} = q_{v,i}),$$

где P — множество вопросов, на которые оба пользователя дали ответ. Этот подход применим как в системах знакомств, так и в других областях, где важна совместимость взглядов, предпочтений и интересов.

Другим эвристическим приёмом является использование показателя популярности. Объекты (например, профили или товары), получившие наибольшее количество положительных оценок, могут предлагаться новым пользователям в качестве стартовых рекомендаций. Популярность может быть нормирована, например:

$$\mathrm{pop}(i) = \frac{\text{число лайков объекта } i}{\text{максимальное число лайков среди всех объектов}}.$$

Популярные рекомендации часто дополняются фильтрацией по демографическим и другим признакам, таким как возраст, пол, географическое местоположение, язык, наличие общих интересов и т. д.

Такие фильтры применяются до или после основного ранжирования и позволяют исключить очевидно нерелевантные варианты. Например, пользователь, заинтересованный только в кандидатах определённого возраста или пола, должен получать только соответствующие предложения.

Также может применяться эвристика совпадения по ключевым признакам. Если в профиле пользователя указаны предпочтения (например, любимые фильмы, занятия, взгляды), система может искать совпадения с другими профилями и ранжировать их по количеству совпавших интересов.

Комбинирование эвристик позволяет построить гибкую систему, способную адаптироваться к условиям отсутствия данных, начальной загрузки, а также повысить обоснованность рекомендаций. При этом эвристические методы легко интерпретируемы, что важно в чувствительных сферах, таких как онлайн-знакомства или подбор персонала.

2.3 Кластеризация (k-means, DBSCAN) и применение в рекомендациях

Кластеризация — это метод обучения без учителя, направленный на группировку объектов в кластеры таким образом, чтобы элементы одного кластера были похожи друг на друга и отличались от элементов других кластеров. В контексте рекомендательных систем кластеризация применяется для:

- сегментирования пользователей (или объектов) по интересам, поведению или признакам;
- повышения масштабируемости рекомендаций за счёт ограничения поиска релевантных кандидатов внутри кластера;
- выявления нишевых предпочтений и персонализированных паттернов.

Одним из наиболее популярных алгоритмов является *k-means*. Он принимает на вход число кластеров k и итеративно минимизирует внутрикластерную дисперсию:

$$\sum_{j=1}^{k} \sum_{x_i \in C_j} \|x_i - \mu_j\|^2,$$

где C_j — кластер, а μ_j — его центр. Алгоритм эффективен при компактных и сферических кластерах, но чувствителен к выбору k и неустойчив к выбросам.

Для более гибкой кластеризации используется алгоритм *DBSCAN* (Density-Based Spatial Clustering of Applications with Noise). Он группирует точки по

плотности: кластером считается связная по плотности область, где каждая точка имеет хотя бы $minPts$ соседей в пределах радиуса ε. DBSCAN способен находить кластеры произвольной формы и автоматически игнорирует шум (выбросы), что делает его особенно полезным в гетерогенных данных.

В рекомендательных системах кластеризация применяется по-разному:

— На пространстве пользователей: сегментация по поведенческим или опросным признакам позволяет формировать кластеры пользователей с похожими предпочтениями. Рекомендации для нового пользователя можно извлекать из наиболее близкого кластера.

— На пространстве объектов: группировка товаров, фильмов, анкет и пр. по тематике, стилю или целевой аудитории позволяет адаптировать рекомендации к интересам пользователя.

— В латентных пространствах: кластеризация векторов после факторизации (например, в SVD или autoencoder-подходах) даёт более сжатое и семантически значимое представление.

Кластеризация также применяется для визуализации и анализа структуры пользовательской базы, выявления целевых групп и построения тематических подборок. Её эффективность во многом зависит от выбора признаков и масштабов данных, поэтому нередко она используется в комбинации с другими методами.

2.4 Стратегии холодного старта

Проблема холодного старта возникает, когда система не располагает достаточной информацией о пользователях или объектах, чтобы формировать персонализированные рекомендации. Выделяют два основных сценария: появление нового пользователя и добавление нового объекта (например, анкеты).

Для новых пользователей могут применяться следующие подходы:

— заполнение вступительных тестов или анкет — позволяет собрать первичные признаки и использовать их при формировании рекомендаций;

— использование демографических данных — рекомендации подбираются на основе поведения пользователей с аналогичными характеристиками (возраст, пол, география и т.д.);

— показ популярных объектов — временная стратегия, при которой пользователю демонстрируются анкеты с высокой оборачиваемостью, что помогает быстрее сформировать профиль предпочтений.

При появлении новых объектов, которые ещё не получили откликов, возможны такие меры:

- временное повышение приоритета в выдаче — например, показ новым или активным пользователям для ускоренного накопления статистики;
- использование признаков схожести — на основе анкетных данных, внешности (в случае CV), текста описания (в случае NLP) или ответов на вопросы;
- размещение в релевантных сегментах — объект может быть временно включён в выдачу по кластерам, в которые он потенциально попадает по признаковому пространству.

Кроме того, существуют универсальные стратегии, применимые и к новым пользователям, и к новым объектам:

- инициализация признаков с помощью доступных внешних данных — анкет, биографий, метаинформации;
- использование гибридных моделей, сочетающих элементы content-based и коллаборативной фильтрации — это снижает чувствительность к отсутствию истории;
- активное обучение — выбор контента, максимально полезного для уточнения предпочтений, что позволяет за минимальное число взаимодействий улучшить качество рекомендаций.

Проблема холодного старта наиболее критична для систем, основанных на коллаборативной фильтрации, поскольку они требуют исторических данных о взаимодействии пользователей с объектами. Поэтому многие современные решения строятся с использованием дополнительных эвристик и предварительной инициализации признаков, что позволяет обеспечить устойчивость системы на ранних этапах использования.

2.5 Методы глубокого обучения в рекомендательных системах

Развитие нейросетевых архитектур оказало существенное влияние на область рекомендательных систем. Благодаря способности извлекать сложные латентные зависимости из разнородных данных, модели глубокого обучения применяются для повышения качества рекомендаций как в традиционных задачах (например, предсказание рейтингов), так и в более сложных сценариях, включая мультимодальные рекомендации, учет временной динамики и персонализацию на основе контекста.

Одним из первых направлений стало расширение классической матричной факторизации с помощью нейронных сетей. Вместо простой линейной факторизации матрицы взаимодействий $R \in \mathbb{R}^{m \times n}$, где m — количество пользователей, n — количество объектов, и R_{ij} — факт взаимодействия, используются обучаемые эмбеддинги и нелинейные функции активации. Примером такой модели является Neural Collaborative Filtering (NCF), предложенная в [**?**]. В NCF пары эмбеддингов $(\mathbf{u}_i, \mathbf{v}_j)$ передаются через многослойный перцептрон:

$$\hat{r}_{ij} = \mathrm{MLP}([\mathbf{u}_i, \mathbf{v}_j]),$$

где $[\cdot, \cdot]$ обозначает конкатенацию векторов. Модель обучается по функции потерь, например, бинарной кросс-энтропии в задаче предсказания лайка/дизлайка.

Другое направление связано с использованием рекуррентных и трансформерных архитектур для моделирования последовательности взаимодействий. Так называемые sequence-based recommenders учитывают порядок взаимодействий пользователя с объектами. Например, модель GRU4Rec [**?**] применяет Gated Recurrent Unit (GRU) для обработки истории действий пользователя. Базовая идея заключается в следующем: пусть $x_1, x_2, \ldots, x_T$ — последовательность взаимодействий, тогда на каждом шаге рассчитывается скрытое состояние h_t:

$$h_t = \mathrm{GRU}(x_t, h_{t-1}),$$

и предсказывается следующий элемент x_{t+1} с помощью softmax-слоя.

Для учёта более длинных зависимостей и параллельной обработки была предложена модель SASRec [**?**], основанная на self-attention. В отличие от рекуррентных моделей, здесь используется позиционно-кодированная последовательность эмбеддингов, проходящая через слои трансформера. Это позволяет учитывать контекст всех предыдущих действий при выборе следующей рекомендации.

Особое место занимают графовые нейронные сети, применяемые для моделирования взаимодействий в виде графов. В Graph Convolutional Matrix Completion (GC-MC) [**?**], каждый пользователь и объект представляются вершинами, соединёнными ребром при наличии взаимодействия. Представления вершин обновляются по правилу:

$$\mathbf{h}_i^{(l+1)} = \sigma\left(\sum_{j \in \mathcal{N}(i)} \frac{1}{c_{ij}} W^{(l)} \mathbf{h}_j^{(l)}\right),$$

где $\mathcal{N}(i)$ — соседи вершины i, $W^{(l)}$ — матрица весов на l-м слое, c_{ij} — коэффициент нормализации. Такой подход позволяет учитывать структуру взаимодействий и взаимосвязь объектов, что особенно актуально для задач, где важны не только пользовательские предпочтения, но и социальные или контекстные связи.

Ещё одно активно развивающееся направление — мультимодальные рекомендации. Здесь помимо взаимодействий учитываются дополнительные признаки, такие как текст описания, изображения, аудио. Для обработки таких данных применяются CNN, BERT и другие специализированные архитектуры. В модели VBPR [**?**] визуальные признаки изображений используются для расширения латентного пространства:

$$\hat{r}_{ij} = \mathbf{u}_i^\top \mathbf{v}_j + \mathbf{u}_i^\top E x_j,$$

где x_j — визуальный вектор, полученный из изображения объекта, E — обучаемая матрица проекции.

Основные преимущества методов глубокого обучения:

— способность моделировать сложные нелинейные зависимости между пользователями и объектами;
— возможность использовать разнородные источники информации;
— высокая гибкость и расширяемость архитектур.

Тем не менее, существуют и ограничения:

— высокая требовательность к вычислительным ресурсам;
— потребность в большом объёме размеченных данных;
— трудность интерпретации и объяснимости результатов.

Таким образом, методы глубокого обучения открывают широкие перспективы для построения более точных и персонализированных рекомендательных систем, особенно в условиях, когда доступны дополнительные источники информации и достаточно ресурсов для обучения. Однако выбор таких подходов должен быть обоснован задачами проекта, размером аудитории и доступной инфраструктурой.

1 Jin, D. A survey on fairness-aware recommender systems / D. Jin, L. Wang, H. Zhang, Y. Zheng, W. Ding, F. Xia, S. Pan // arXiv preprint arXiv:2306.00403. — 2023.

2 Recommender systems in the era of large language models (llms) / Z. Zhao, W. Fan, J. Li, Y. Liu, X. Mei, Y. Wang, Z. Wen et al. // arXiv preprint arXiv:2307.02046. — 2023.

3 Li, Y. A collaborative filtering recommender systems: Survey / Y. Li, K. Liu, R. Satapathy, S. Wang, E. Cambria // Neurocomputing. — 2025. — T. 617. — 128718 p.

4 Content-based collaborative generation for recommender systems / K. Bao, J. Zhang, W. Wang, Y. Zhang, Z. Yang, Y. Luo, F. Feng et al. // Proceedings of the 2023 ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. — 2023.

5 Saini, K. Hybrid recommender system for e-commerce: A comprehensive survey / K. Saini, A. Singh // Journal of Harbin Engineering University. — 2023. — T. 44, no. 8.

6 Zhou, H. A comprehensive survey of recommender systems based on deep learning / H. Zhou, F. Xiong, H. Chen // Applied Sciences. — 2023. — T. 13, no. 20. — 11378 p.

7 Zhang, K. A comprehensive review of recommender systems: Transitioning from theory to practice / K. Zhang, Q. Cao, F. Sun, Y. Wu, S. Tao, H. Shen, X. Cheng // arXiv preprint arXiv:2309.02057. — 2023.

8 Hitsch, G. J. What makes you click?—mate preferences in online dating / G. J. Hitsch, A. Horta?su, D. Ariely // Quantitative Marketing and Economics. — 2010. — T. 8, no. 4. — Pp. 393–427.

9 Courtois, C. Cracking the tinder code: An experience sampling approach to the dynamics and impact of platform governing algorithms / C. Courtois, E. Timmermans // Journal of Computer-Mediated Communication. — 2018. — T. 23, no. 1. — Pp. 1–16.

10 Pidoux, J. Online Dating Quantification Practices: A Human–Machine Learning Process: Ph.D. thesis / ?cole Polytechnique F?d?rale de Lausanne. — 2021.

11 Jennings, R. The tinder algorithm, explained / R. Jennings // Vox. — 2019. https://www.vox.com/2019/2/7/18210998/tinder-algorithm-swiping-tips-dating-app-science.

12 Bruch, E. E. Aspirational pursuit of mates in online dating markets / E. E. Bruch, M. E. J. Newman // EPJ Data Science. — 2018. — T. 7, no. 1. — Pp. 1–14.

13 Lefebvre, H. Psychometric matching and the evolution of eharmony's recommendation engine / H. Lefebvre, Y. Dallal // Journal of Personality and Social Psychology. — 2022. — T. 123, no. 5. — Pp. 912–926.

14 Lee, A. Limited choice, better matches: The coffee meets bagel approach // Proceedings of the 2021 ACM Conference on Recommender Systems. — ACM, 2021. — Pp. 345–350.