

**УДК 004.032.26**

**П.С. Якунин**

студент кафедры высшей математики и механики

## **СРАВНИТЕЛЬНЫЙ АНАЛИЗ ПОДХОДОВ К РАНЖИРОВАНИЮ В СИСТЕМАХ РЕКОМЕНДАЦИЙ НОВОСТЕЙ**

**Аннотация:** В работе проведен сравнительный анализ методов ранжирования в системах рекомендации новостей. Были исследованы подходы точечного, парного и списочного ранжировании и экспериментально проверены на эффективность на выборке новостных данных. Полученные данные позволяют сделать выводы о преимуществах и недостатках, рассматриваемых подходов.

**Ключевые слова:** Рекомендательные системы, рекомендации новостей, точечный, парный, списочный подходы.

Новостной домен представляет собой уникальный вызов для рекомендательных систем из-за высокой скорости устаревания контента (*cold start problem*) и динаминости интересов пользователей. В отличие от e-commerce или киноиндустрии, где товары живут годами, новости теряют актуальность за часы. Целью данного исследования является выявление оптимального подхода к ранжированию, способного балансировать между релевантностью контента и персонализированными предпочтениями аудитории.

В качестве экспериментальной базы используется набор данных, включающий логи поведения пользователей ('behaviors.tsv') и метаданные статей ('news.tsv'). Основная идея данной работы состоит в том, чтобы проверить на практике какой из подходов к ранжированию более окажется наиболее эффективным.

### **Датасет**

Для исследования используется публичный датасет для рекомендаций новостей MIND-small (Microsoft News Recommendation Dataset) [1]. Данный набор данных был собран из анонимизированных журналов поведения на веб-сайте Microsoft News и представлен широкой публике в [4]. Данные были получены случайным образом из 1 миллиона пользователей, которые как минимум 5 раз переходили по новостям в течение 6 недель с 12 октября по 22 ноября 2019 года.

В датасет входят 2 файла:

behaviors.tsv - взаимодействия пользователей с новостями, отражая их поведение в рамках системы рекомендаций. Каждый экземпляр в этом файле соответствует одному показу новостей определенному пользователю и записан в виде строки с пятью полями:

- Impression ID — уникальный идентификатор показа;
- User ID — анонимизированный идентификатор пользователя;
- Time — метка времени показа (в формате "MM/DD/YYYY HH:MM:SS AM/PM");
- History — последовательность идентификаторов новостей, ранее просмотренных/кликнутых этим пользователем до текущего показа; новости упорядочены по времени совершения клика;
- Impressions — список новостей, показанных в рамках текущей импресии, а также бинарные метки клика (1 — клик, 0 — пропуск); порядок новостей внутри списка перемешан для предотвращения позиционного шума.

news.tsv - информация о новостных статьях, фигурирующих в behaviors.tsv. Каждая строка соответствует одной новости и содержит 7 колонок:

- News ID — уникальный идентификатор статьи;
- Category — категория контента;
- SubCategory — подкатегория;
- Title — заголовок новости;
- Abstract — краткое описание;
- URL — ссылка на полную версию статьи;
- Title Entities — перечень сущностей, найденных в заголовке (например, имена, компании);
- Abstract Entities — сущности из аннотации.

Такая детализация позволяет использовать тематическую и семантическую информацию о статьях для построения разметки признаков и совершенствования моделей персонализированных рекомендаций.

## **Предобработка данных**

Перед построением моделей была проведена подготовка данных. Для повышения устойчивости моделей:

1. исключены пользователи с менее чем тремя кликами;
2. исключены статьи, получившие менее пяти кликов.
3. Проведена TF-IDF токенизация для “Title” и “Abstract”

## **Формирование обучающей выборки**

Данные о показах новостей представлены в агрегированном виде и требуют преобразования в формат «пользователь–статья–метка». Для этого применяется операция развертывания, формирующая набор индивидуальных примеров для обучения моделей типа LightGBM и LambdaMART, методах машинного обучения, построенных на базе бустинга над деревьями из [3].

Также для предотвращения утечки информации обучение и оценка проводятся на временном разбиении: train — первые 70% временной ленты, val — следующие 15%, test — последние 15%.

## **Моделирование подходов**

Классическое деление методов обучения ранжированию на поточечные, попарные и списочные подробно изложено в [3].

### **Поточечный метод (Pointwise, pointwise classification)**

В рамках данной статьи Pointwise подход сводится к задачи бинарной классификации. Модель обучается на задаче минимизации LogLoss (бинарной кросс-энтропии, 1 формула), предсказывая для каждого документа свой скор. Это можно интерпретировать как оценка вероятности клика по документу. А итоговое ранжирование определяется сортировкой по полученному скору. Несмотря на простоту реализации и хорошую калибровку вероятностей, данный метод обладает существенным недостатком: он игнорирует контекст выдачи и конкуренцию документов за внимание пользователя внутри одного списка.

$$L = - \sum (y \log p + (1-y) \log(1-p)) \quad (1)$$

### **Попарный метод (Pairwise, RankNet)**

Pairwise подход, реализованный через RankNet, фокусируется на правильном упорядочивании пар документов, а не на абсолютных значениях оценки. Оптимизируется вероятность того, что пользователь предпочтет пример  $i$  примеру  $j$ , используя функцию потерь (формула 2). Такой алгоритм менее чувствителен к дисбалансу классов, но обладает меньшей интерпретируемостью, так как скор не является вероятностью.

$$L = - \sum_{(i, j) \in D_s} \ln \sigma(\bar{x}_i - \bar{x}_j) \quad (2)$$

### **Списочный метод (Listwise, LambdaMART)**

Списочный (Listwise) подход является наиболее теоретически обоснованным для задач ранжирования, так как он оптимизирует качество всего списка выдачи целиком. Используемый алгоритм LambdaMART модифицирует градиенты бустинга таким образом (как показано на формуле 3), что градиент для каждой пары документов взвешивается на величину изменения целевой метрики ранжирования, которое произошло бы при перестановке этих документов, что показано в 4 формуле. Это означает, что модель "штрафуется" значительно сильнее за ошибки на верхних позициях списка, чем за ошибки внизу, что идеально соответствует логике работы поисковых и рекомендательных систем. Такой подход позволяет модели выучить сложные нелинейные зависимости между признаками.

$$l_{ij} = \log\left(1 + \exp\left(-\left(s_i - s_j\right)\right)\right) \quad (3)$$

$$w_{ij} = |\Delta NDCG_{ij}| \quad (4)$$

где  $w_{ij}$ - вес важности пары, а  $\Delta NDCG_{ij}$ - изменение NDCG, если поменять местами позиции объектов  $i$  и  $j$ ,  $l_{ij}$  - попарная логистическая функция потерь.

### **Анализ результатов**

Оценка эффективности производилась с помощью метрик NDCG (качество полного ранжированного списка с приоритетом на верхние позиции), MRR (оценивает успех по первому релевантному элементу) и Accuracy (попал ли клик в 1-ю позицию), как видно из рисунка 1, модель поточечного подхода показывает себя наихудшей результат при  $k = 5$  и  $k = 10$  она явно уступает другим подходам. В тоже время модель списочного подхода показывает наилучший результат при обоих параметрах  $k$ , хотя при  $k = 5$  разрыв небольшой, а при  $k = 10$  метрика становится почти идентичной с попарным подходом, при дальнейшем

росте  $k$  все модели выдают идентичные результаты, что видно на рисунке 2. Также из рисунка 1 мы видим, что по метрике MRR поточечный и попарный подходы сопоставимы и оба уступают списочной модели. Это согласуется с NDCG при  $k = 5$ , из чего можно сделать вывод о большой точности на верхних позициях рекомендации у списочного подхода. Это же подтверждает наилучшее значение Accuracy. Такое происходит из-за того, что оптимизируется метрика ранжирования по группам и ошибки в топе штрафуются сильнее. В тоже время попарная модель показала Accuracy ниже поточечной, это значит, что она чаще ранжирует лучше в целом, но не так стабильно ставит клик строго на 1-е место.

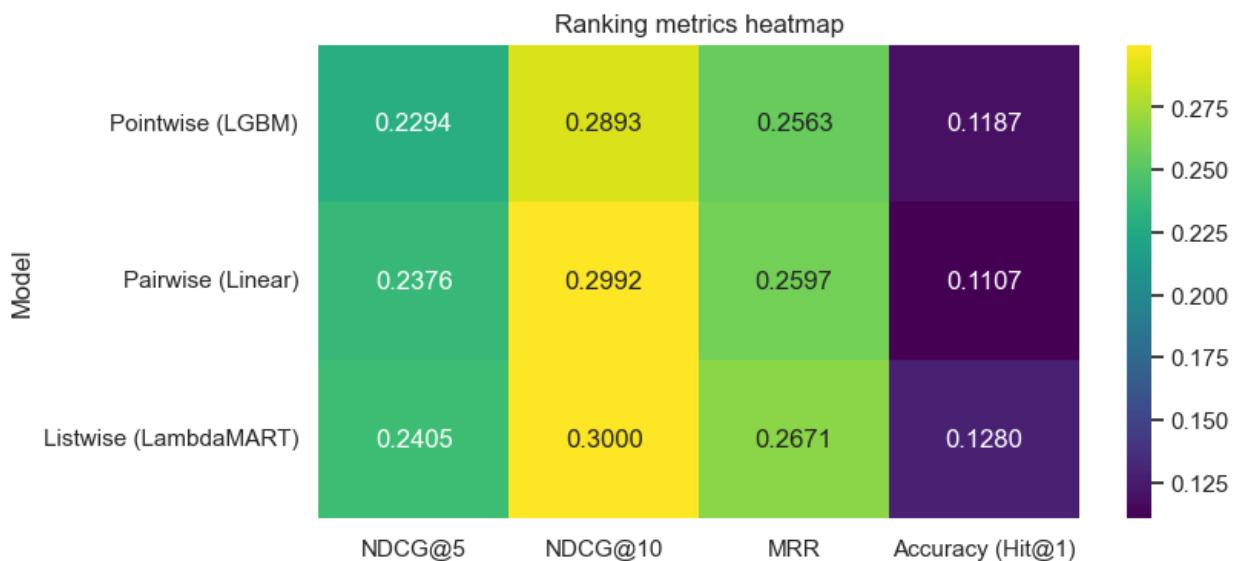


Рисунок 1 – Сравнительный анализ метрик качества моделей

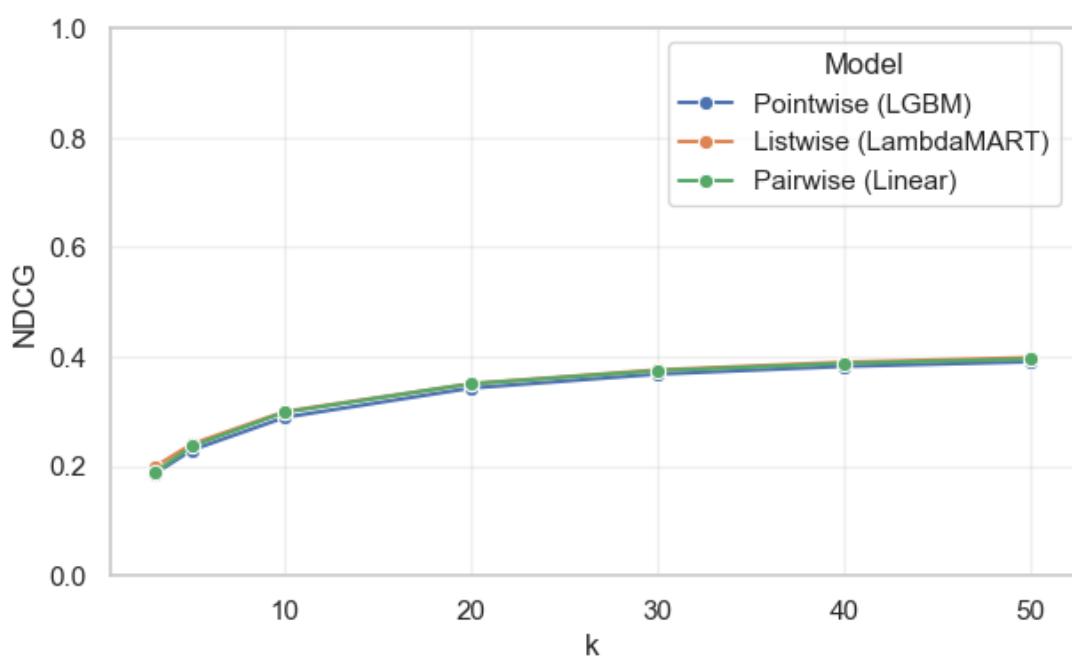


Рисунок 2 – Метрика NDCG при разных  $k$

Таблица 1 – Hit-rate на k

| <b>k</b>  | <b>3</b> | <b>5</b> | <b>10</b> | <b>20</b> | <b>30</b> | <b>40</b> | <b>50</b> |
|-----------|----------|----------|-----------|-----------|-----------|-----------|-----------|
| Listwise  | 0.289755 | 0.404030 | 0.593812  | 0.774420  | 0.866703  | 0.915902  | 0.943290  |
| Pairwise  | 0.279007 | 0.411450 | 0.610991  | 0.793218  | 0.877946  | 0.924672  | 0.950801  |
| Pointwise | 0.270507 | 0.389144 | 0.580770  | 0.774960  | 0.870076  | 0.918735  | 0.946888  |

В таблицу 1 виден hit-rate (это доля случаев, когда модель угадала нужный объект в топ-N рекомендаций). По нему можно сделать следующие важные выводы:

- Быстрый рост - модели очень быстро находят релевантный контент. Уже в топ-5 рекомендациях вероятность клика достигает ~40%. Это критически важно, так как пользователи редко доходят до дальних элементов.
- Лидерство попарного подхода: показывает лучшие результаты, но с небольшим отрывом. Хотя, стоит отметить, что при  $k = 3$ , списочный подход оказался точнее.
- Сходимость к  $k=20$  линии сближаются, так как при длинном списке даже слабая модель случайно угадывает интересную новость.

### Визуальный анализ плотности распределения предсказаний (KDE)

На рисунках 3, 4, 5 представлены оценки распределения предсказанных для новостных материалов с разметкой "Skipped" (пропущено) и "Clicked" (кликнуто), полученные с использованием различных подходов к ранжированию. Анализ данных графиков позволит лучше понять поведение моделей.

Так на рисунке 3 (графика поточечного подхода) мы видим два четких пика в диапазоне низких вероятностей (0.025 - 0.05). Clicked и Skipped почти полностью перекрываются. Пик Skipped высокий и узкий, Clicked лишь слегка смещён вправо, но отделения почти нет. Это следует интерпретировать как - низкие абсолютные значения вероятностей, модель не склонна давать очень уверенные предсказания. При таком перекрытии распределений порядок документов может легко меняться от шума. Это означает неустойчивое ранжирование.

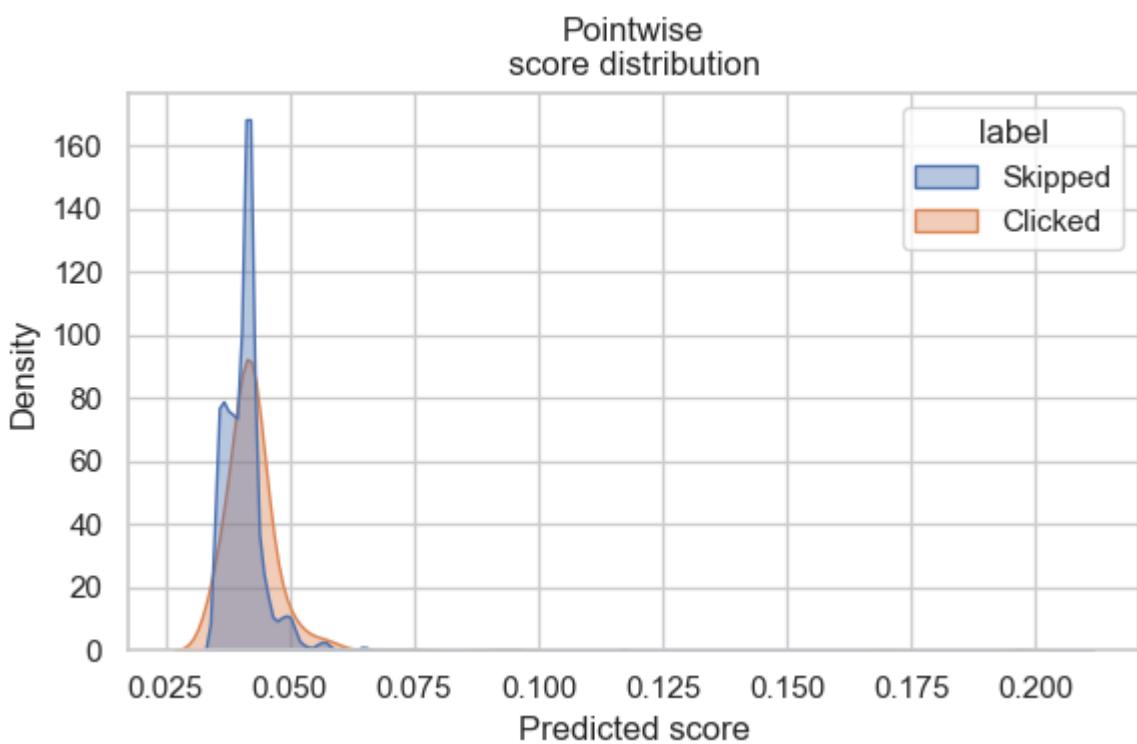


Рисунок 3 – Плотность распределения предсказаний для поточечного подхода

Анализируя график попарного подхода, рисунок 4, можно заметить, что распределения имеет заметную разделимость Clicked и и Skipped. У Clicked выше плотность в зоне больших значений, у Skipped в зоне меньших. Явная разделимость обусловлена задачей модели, которая раздвигает классы по скору.

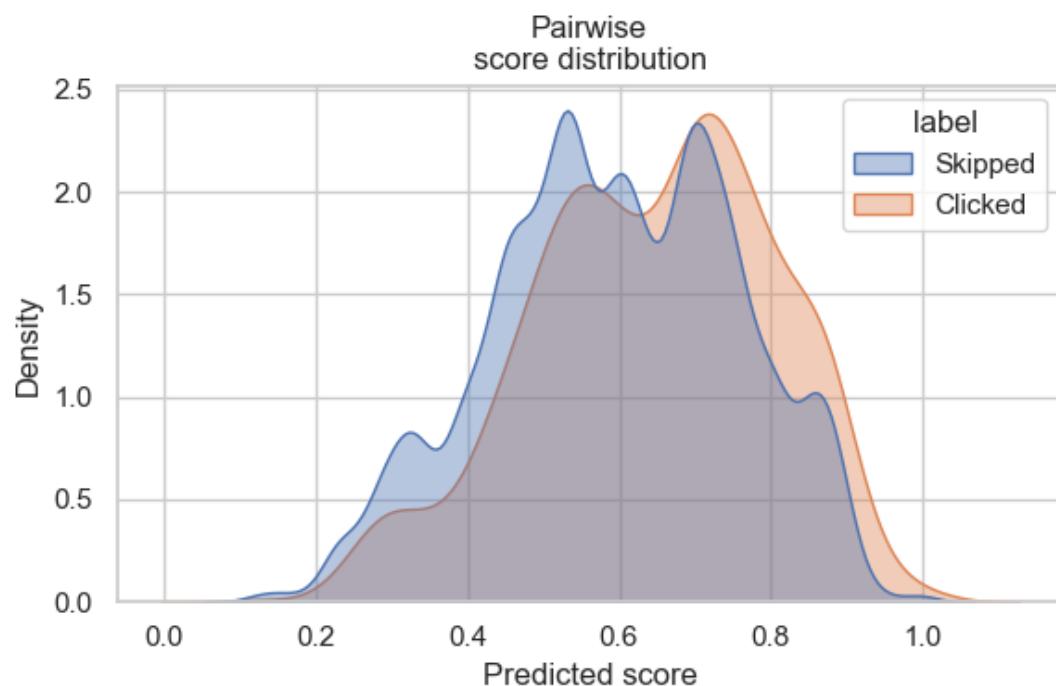


Рисунок 4 - Плотность распределения предсказаний для парного подхода

На рисунке 5 – графике распределения списочного подхода мы видим сложную форму с основным пиком около 0 и вторичным пиком в отрицательной области (около -1.5). Наличие двух пиков объясняется тем, что разные подгруппы объектов получают разные диапазоны скоров. Clicked лежит правее, а Skipped – левее. Перекрытие остаётся, но оно существенно меньше, чем у Pointwise модели.

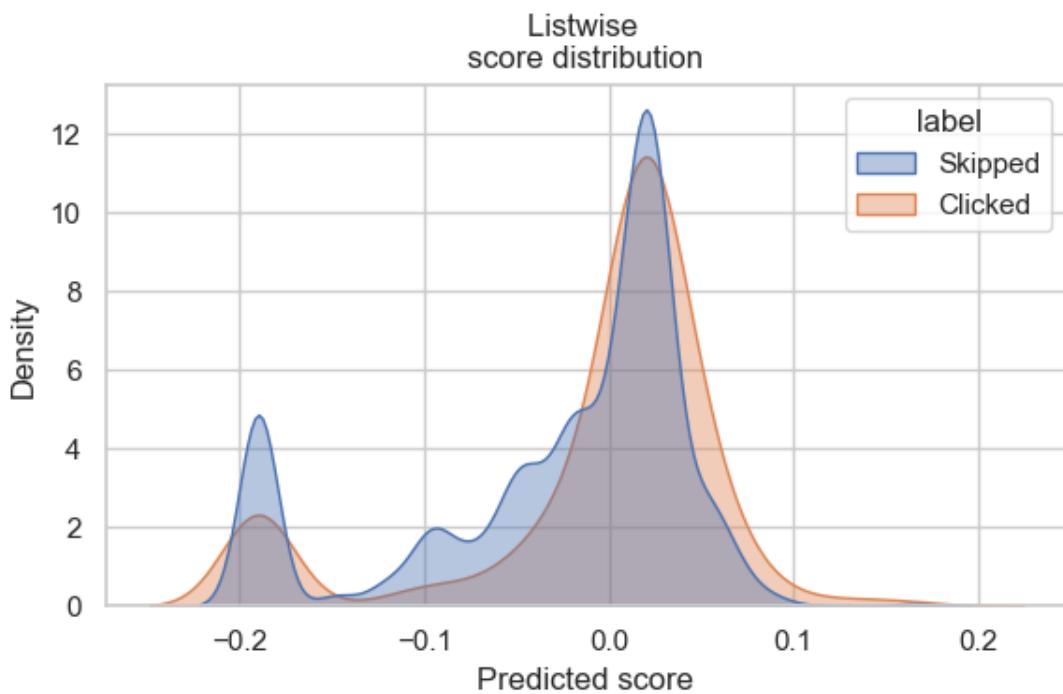


Рисунок 5 - Плотность распределения предсказаний для списочного подхода

Следует отметить, что по графикам распределения попарный подход выглядит наиболее перспективно, он даёт самое сильное разведение классов по скору. Но списочный подход получил наилучшие результаты по метрикам, это объясняется тем, что модель учится раздвигать релевантные документы выше нерелевантных внутри каждого списка, а выходной скор становится прежде всего показателем ранга документа, а не вероятности клика.

## Вычислительная сложность

Важным пунктом, который нельзя не упомянуть, является вычислительная сложность для обучения и предсказаний моделей. Как видно из данных, представленных в таблицу 2. Модель попарного подхода гораздо быстрее обучается, время на предсказание также меньше, чем у других моделей, хотя такого большого разрыва, как на обучении уже нет. Самым неэффективным оказался списковой подход, а поточечный – занял среднее место по двум критериям.

Таблица 2 – Время на обучения и предсказания моделей

| Подход                   | Обучение (с) | выполнение (с) |
|--------------------------|--------------|----------------|
| Listwise<br>(LambdaMART) | 79.000433    | 39.394317      |
| Pairwise<br>(RankNet)    | 31.220730    | 39.056416      |
| Pointwise<br>(lightgbm)  | 76.673793    | 39.941115      |

## Заключение

В рамках данной работы был проведён сравнительный анализ трёх основных подходов обучения моделей ранжирования в новостных рекомендательных системах: поточечного (pointwise), попарного (pairwise) и списочного (listwise). Исследование выполнено на публичном датасете MIND-small.

По полученным данным можно сделать вывод, что все рассмотренные подходы демонстрируют сопоставимое качество ранжирования. Однако на верхних позициях рекомендаций, наиболее критичных для пользователя, выявлены некоторые различия. Списочный подход продемонстрировал наилучшие значения метрик NDCG, MRR и Accuracy, что свидетельствует о более высокой точности размещения релевантных документов на верхних позициях выдачи. Это объясняется тем, что списочные методы напрямую оптимизируют метрики ранжирования по группам и сильнее штрафуют ошибки в верхней части списка.

Попарный подход (RankNet) показал устойчивые и конкурентоспособные результаты, особенно по метрике hit-rate, где он в ряде случаев превосходил альтернативные методы. Визуальный анализ распределений предсказанных скоров подтвердил, что именно попарная модель обеспечивает наиболее выраженное разделение кликовых и некликовых объектов. При этом более низкие значения Accuracy указывают на меньшую стабильность постановки клика строго на первую позицию, несмотря на хорошее качество ранжирования в целом.

Поточечный подход, несмотря на концептуальную простоту и интерпретируемость предсказаний, оказался наименее устойчивым с точки зрения разделимости классов и качества ранжирования на верхних позициях. Перекрытие распределений кликов и

пропусков и низкая уверенность предсказаний приводят к повышенной чувствительности к шуму и нестабильности порядка документов. Тем не менее, данный подход демонстрирует приемлемое качество и может рассматриваться как базовое или малозатратное решение.

Анализ вычислительной сложности показал, что попарный подход является наиболее эффективным с точки зрения времени обучения, в то время как списочный подход требует наибольших вычислительных затрат. Это подчёркивает практическую необходимость учитывать не только итоговое качество рекомендаций, но и ограничения при выборе модели.

Таким образом, результаты исследования подтверждают, что не существует универсально оптимального метода ранжирования для всех сценариев. Списочные методы целесообразно применять в системах, ориентированных на максимальное качество верхних позиций и устойчивость ранжирования, тогда как попарные модели представляют собой разумный компромисс между качеством и вычислительной эффективностью. Поточечные подходы, в свою очередь, могут быть оправданы при необходимости простой интерпретации результатов.

Полученные выводы подчёркивают важность выбора парадигмы ранжирования на основе особенностей данных и ограничений, а не только на основе теоретической сложности используемого метода.

## **Библиографический список**

1. Веб-сайт репозитория датасета MIND-small. URL: <https://msnews.github.io/> (дата обращения 23.11.2025).
2. Задача ранжирования [Электронный ресурс] // Учебник по машинному обучению. Образовательная платформа Яндекс. – 2025. – URL: <https://education.yandex.ru/handbook/ml/article/zadacha-ranzhirovaniya> (дата обращения: 23.11.2025)
3. Кэ Г., Мэн Ц., Финли Т., Ван Т., Чэнь В., Ма В., Е Ц., Лю Т.-Ю. LightGBM: высокоэффективный градиентный бустинг над деревьями решений // Труды конференции по нейронным информационным системам (NeurIPS). 2017. Т. 30.
4. У Ф., Цяо Ю., Чэнь Ц., У Ц., Ци Т., Лянь Ц., Лю С. MIND: крупномасштабный набор данных для рекомендательных систем новостей // Труды ежегодного собрания Ассоциации компьютерной лингвистики (ACL). 2020. С. 3597–3606.