Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Enhancing Mobile App Ranking through Trust-Based Sentiment Analysis using Large Language Models

Muhammad Hassam Aslam khan

khan19@iu.edu

Indiana University Indianapolis Indiana

Rajeev R. Raje

Indiana University Indianapolis Indiana

Research Article

Keywords: Mistral, LLaMA, Gemini, BDU, Google Play

Posted Date: March 5th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-8943112/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Enhancing Mobile App Ranking through Trust-Based Sentiment Analysis using Large Language Models

Muhammad Hassam Aslam Khan
Department of Computer Science
Indiana University Indianapolis
Indiana, USA
khan19@iu.edu

Rajeev R. Raje
Department of Computer Science
Indiana University Indianapolis
Indiana, USA
rraje@iu.edu

***Abstract*—Selecting the "most suitable" app from a category remains a challenge due to the presence of multiple highly rated alternatives offering similar functionality. Users often rely on trial and error, including installing and testing several apps, before finding one that meets their needs. In this study, we extend previous research conducted by our group and propose an enhanced trust-based app ranking approach that takes advantage of user feedback available in the "Relevant" section of Google Play reviews. We collect apps from a specific category and analyze user reviews using three different large language models (LLMs) to extract sentiment data. From these sentiment scores, we derive a BDU tuple, representing Belief, Disbelief, and Uncertainty about the app. We use the BDU tuple to compute a trust score that reflects users' perceived trust about each app. Apps are then ranked based on these trust scores. We compare our ranking with existing app ratings, available in Google Play, using the Kendall Tau distance. Our approach offers a user-centric alternative to traditional rating systems by emphasizing perceived trustworthiness derived from real user experiences.**

***Index Terms*—Mistral, LLaMA, Gemini, BDU, Google Play**

I. Introduction

Choosing the "most suitable" app from any category on platforms like Google Play can be confusing and time consuming. Since many apps have high ratings and large download numbers, users often end up installing and testing multiple apps before finding one that meets their needs. This problem is further exacerbated when many reviews and ratings do not accurately reflect actual user experiences [1].

Understanding how users feel about an app, through their textual reviews, provides a valuable way to measure app quality and user satisfaction. Reviews often offer rich, detailed feedback on actual performance and usability. For example, a popular app, Chess – Play & Learn *Chess – Play & Learn* app [2] currently boasts a 4.8-star rating with over 2.48 million reviews on the Google Play Store. Similarly, AI Factory's *Chess* app [3] maintains a 4.8-star average, supported by approximately 14,000 user reviews.

These large volumes of user-generated feedback highlight the potential of review analysis for evaluating app credibility and enhancing recommendation accuracy. Instead of focusing only on the average rating and/or the number of downloads, analyzing user reviews can provide deeper insights into how reliable, trustworthy or user friendly an app really is. Previous research has shown that user feedback and sentiment analysis can play a significant role in app recommendation systems [4] [5].

This study focuses on using large language models (LLMs) to understand what users are commenting about apps in their reviews. We propose a ranking scheme for similar apps by calculating trust scores based on the opinions expressed by users. By collecting "Relevant" reviews from Google Play and analyzing them with three different large language models, Llama, Gemini, and Mistral, we extract sentiments and evidence and then compute trust scores based on a Belief, Disbelief, and Uncertainty (BDU) tuple for each app. We then compare the app rankings generated by each LLM and their combination with default ranking in Google Play using Kendall Tau distance, which shows how much the rankings agree or differ with each other. Unlike the E-SERS approach [6], which integrates both internal artifacts (e.g., security vulnerabilities via the static taint analysis) and external user opinions to assess app trustworthiness, our work focuses exclusively on extracting and interpreting user feedback using multiple LLMs. While E-SERS [1] includes programmatic evidence and reviewer reputation weighting, we emphasize on the language-model-driven interpretation of trust embedded in review narratives. In this sense, our work complements E-SERS by proposing a lightweight, review-only alternative that can scale quickly without needing APK analysis or security instrumentation.

This work is designed to support end users in making more informed decisions about how apps are perceived by other users. While we do not explicitly analyze code- level vulnerabilities or permission risks as in E-SERS, our contribution

lies in showing how LLMs can offer a fine-grained, sentiment-based trust analysis in a crowded app marketplace, especially where internal artifacts may be inaccessible. This trust-based ranking approach, using LLMs, offers a novel, user-centric dimension for evaluating mobile apps, potentially increasing both satisfaction and usability.

The rest of the paper is organized as follows: Section II discusses related work. Section III presents our methodology and implementation. Section IV describes the experiments conducted using Google Play. Section V outlines the limitations of our approach, and Section VI concludes the paper with future work and lessons learned.

II. Related Work

Mobile app platforms such as Google Play and Apple's App Store employ complex and not published ranking algorithms that account for numerous factors, including downloads, ratings, and reviews, to present relevant apps to user queries. In Google Play, specifically, average star ratings and user reviews significantly impact an app's visibility and ranking [7]. However, selecting an app out of choices solely based on the average star rating can be problematic. Prior research has shown that star ratings frequently fail to accurately reflect true user sentiment or the actual quality of the app. For example, Noei et al. identified multiple cases where extremely positive textual reviews were paired with low star ratings and vice versa, underscoring significant inconsistencies in the user feedback [8]. These inconsistencies can mislead traditional ranking systems that treat ratings as definitive. In addition, such a discrepancy can led to an incorrect app selection by a naive user if they base the decision only on the available average star ratings.

User reviews from app stores have been extensively studied for sentiment and opinion analyses and to gauge user satisfaction precisely. Early work by Guzman et al., introduced a fine-grained sentiment analysis to app reviews, revealing that analyzing textual content could provide deeper insights into user satisfaction with specific features independent of star ratings [9]. Their findings emphasized the nuanced information can be often overlooked by focusing on the star ratings alone.

Opinion mining for positive and negative sentiment continues to be foundational in quantifying user attitudes. Traditional approaches range from lexicon-based sentiment detection to machine learning classifiers trained on labeled datasets. For example, Panichella et al. used sentiment lexicons to identify negative sentiments reflecting user frustrations [10]. The advent of deep learning enhanced sentiment analysis significantly. Studies employing LSTM-based models, such as those analyzing Google Play feedback, successfully predicted sentiment trends, identifying apps experiencing declining user satisfaction [11].

More recently, transformer-based models like BERT and LLMs have emerged, offering improved accuracy in sentiment classification tasks. Niimi et al. demonstrated that instruction-tuned LLMs notably outperform classical models in accurately analyzing user sentiments from reviews [12]. In addition to sentiment analyses, our approach employs research on trust and reliability assessments derived from user-generated content. Central to this domain is Subjective Logic (SL), introduced by Jøsang, which uses belief, disbelief, and uncertainty to represent opinions probabilistically [13]. Unlike simple binary or single-probability approaches, Subjective logic explicitly models uncertainty, critical when aggregating multiple noisy user reviews. Esposito et al. expanded on this by proposing a multi-criteria trust assessment in social reviewing systems, effectively combining diverse signals, such as text sentiment, user credibility, and review timelines to calculate robust trust scores [14].

Similarly, the Dempster–Shafer theory of evidence has been employed to aggregate review opinions and handle uncertainty effectively. Kumar et al. utilized this theory to infer user trustworthiness through reputation aggregation based on review analyses [15]. While simplistic methods might label outlier reviews as malicious, these trust models consider temporal dynamics and source credibility to account for legitimate shifts in app quality or minority yet valid opinions

In the specific context of mobile apps, Chowdhury et al. explored trust-based app ranking through BDU trust tuples, derived from static code metrics such as security and privacy permissions [6]. They subsequently enhanced this method by incorporating user review sentiment [1], demonstrating that an integrated approach considering both internal and external evidence provides a holistic approach and improves overall app selection accuracy.

Recent work has shown that transformer-based approaches like LLaMA and BERT outperform traditional methods in sentiment analysis, particularly on user-generated content [16]. While prior studies often use a single LLM for classification, our approach compares multiple LLMs (LLaMA, Mistral, Gemini) to extract sentiments and capture agreement/dis agreement across different LLMs. We then translate these outputs into BDU tuples using subjective logic. Unlike E-SERS [1], which combines code-level security analyses with review sentiments, we focus solely on LLM-derived analyses of reviews and hence, offer a scalable review-centric ranking system for apps offering similar functions.

III. Methodology and Implementation

To rank apps offering a similar function, using the reviews available in Google Play, we designed a multi-step methodology. Our approach included analyzing Google Play reviews by using three popular LLMs. The goal is to extract evidence from reviews for a set of similar apps, compute trust scores

based on extracted evidence using subjective logic, and then compare these apps. This section outlines the steps taken for data collection, processing, scoring, and comparison. Figure 1 shows our approach.

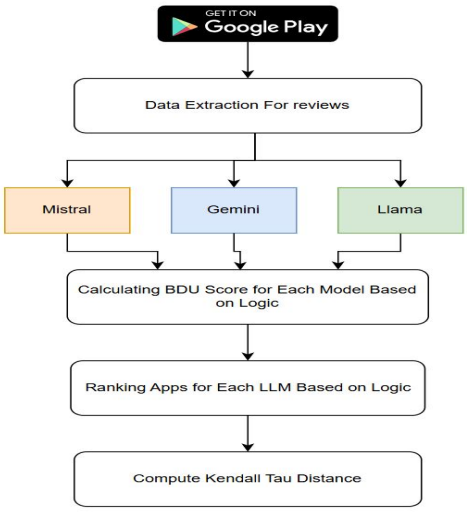

Fig. 1. Flow Diagram

A. Data Collection

We used Google Play reviews as our primary dataset for a set of popular apps offering a similar functionality. To extract customer reviews, we implemented a Python script using Selenium. This script automated the retrieval of 1000 user reviews for each of these apps and stored them in an Excel file, including all relevant metadata as described in Table I.

TABLE I
EXPLANATION OF DATASET HEADERS

Column Name	Description
`Name`	The username or display name of the reviewer on the platform.
`Rating`	Numerical rating provided by the user, typically on a scale from 1 (lowest) to 5 (highest).
`Review`	The textual review written by the user, expressing their experience, feedback, or complaints.
`Date`	The date when the review was posted. This helps analyze trends over time.
`Helpful`	Number of other users who marked the review as helpful. This can indicate the impact or trust level of the review.

B. LLM Selection

The rapid advancement of LLMs in recent years has provided powerful tools for analyzing and extracting insights from user-generated content. Popular LLMs such as LLaMA (and LLaMA 2 by Meta), Mistral 7B, and Google's Gemini represent the state of the art in language understanding. These models are trained on massive text corpora and demonstrate strong capabilities in tasks such as sentiment analysis, summarization, and reasoning.

Recent research supports using LLMs for textual analyses. For example, Niimi et al. studied the application of local LLMs for sentiment classification of online reviews [12], showing that a fine-tuned LLaMA-based model (70B parameters) achieved a high accuracy in classifying restaurant reviews, outperforming both smaller models and traditional deep learning networks.

As indicated earlier, we have used three popular LLMs in this study. Those are briefly described below.

a) Gemini: Gemini, announced by Google DeepMind in late 2023, is a multimodal LLM capable of processing both text and other media. While noted for its strong reasoning abilities, it also excels in pure text-analyses tasks [17]. We included Gemini in our study to examine how effectively it interprets user sentiments and uncertainties from the "Relevant" review section on Google Play.

b) Mistral-7B: Mistral 7B is a lightweight and high-performance language model introduced in late 2023. With 7.3 billion parameters, it has shown superior performance compared to larger models such as LLaMA-2 13B on a range of standard NLP benchmarks, and often rivals models with more than 30 billion parameters, including LLaMA-1 34B, particularly in reasoning, mathematics, and code-related tasks [18]. Its architecture incorporates grouped-query attention and sliding-window attention, contributing to its computational efficiency. These qualities make Mistral 7B especially suitable for real-time review analysis and on-device sentiment classification, which motivated its inclusion in our study.

1) LLaMA-7B: LLaMA-7B, from Meta's open LLaMA model family, is known for robustness and competitive performance in its parameter class. Although smaller than its 13B and 70B counterparts, LLaMA-7B offers strong capabilities for fine-tuned applications. We included it in our study to evaluate its effectiveness in sentiment classification and review summarization tasks

C. Sentiment and AI Detection Scripts

For each LLM, we developed scripts to process the review files for several popular apps. Each script reads individual reviews and uses an LLM to analyze the content. At the prompt, we ask each LLM to provide following three outcomes on the given reviews.

- Sentiment Classification: Each LLM classifies the sentiment of a review as either Positive, Negative, or Neutral. The output is returned as text labels only.
- AI Score: A value between 0 and 1 indicating the likelihood of the review being AI-generated.
- Review Type: Functional parameters (e.g., feedback about bugs, crashes, features, or usability) and non-functional features (e.g., feedback about design, ads, performance, etc.).

1) Prompt Design: We used a comprehensive prompt across (shown below) all three LLMs to ensure consistency in our sentiment analysis study. The prompt was manually crafted and refined through pilot testing on a subset of reviews to check its clarity and model output quality [19]. Additionally, the prompt included explicit examples highlighting stylistic and structural cues—such as coherence, emotional nuance, repetition, and contextual relevance, that align with recognized heuristics for distinguishing AI-generated text [20]. These examples guided LLMs in assessing review authenticity, classifying sentiment (positive, negative, or neutral), and identifying whether a review addressed functional or non-functional aspects of the app.

```
We are analyzing a review for a snooker
app to determine if it is AI-generated.
Consider the following guidelines to
assist your analysis:
- Signs of AI-Generated Text:
- Lack of contextual depth or incorrect
context.
- Repetitive or formulaic phrases.
- Overly formal language or
inconsistent flow.
- Inconsistencies in style or content
over long texts.
- Failing to capture subtle nuances of
emotion or cultural references.
- Characteristics of Human-Generated
Text:
- Diverse vocabulary and complex
sentence structures.
- Use of idiomatic language and
cultural depth.
- Consistent and coherent personal
style.
- Logical argument progression and
narrative depth.
- Emotional resonance and distinct
personal voice.
- Review Type Classification:
- 'Functional': Feedback about bugs,
crashes, features, or usability.
- 'Non-functional': Feedback about
design, ads, performance, etc.
Please provide the response only in
JSON format for the following review:
'{review_text}':
{
'AI Score': '0{1 scale',
'Sentiment': 'positive/negative/neutral',
'reviewtype': 'Functional/Non-functional'
}
```

Each review was then processed through the three LLMs using this standardized prompt. The resulting analysis—including sentiment classification (positive, negative, or neutral), review type (functional or non-functional – i.e., indicating keywords about the functionality of the app and/or indicating non-functional – e.g., preferences and performance-related aspects of the app, respectively), and the AI-likelihood score—were stored alongside the original review for subsequent evaluation.

After obtaining outputs from three LLMs using the standardized prompt, the next step involved computing the BDU values and deriving a trust score for each app. This score, as indicated earlier provided a quantifiable measure of user sentiment grounded in the outputs from multiple models.

D. BDU Values and Trust Score Calculation

We use the BDU (Belief-Degree-Uncertainty) tuple, from Subjective Logic [21], to evaluate sentiment classification outputs obtained from different three LLMs for each review analyzed. The B, D, U values for each app, based on analyses of each review, are computed as follows:

- $b = \frac{\text{positive sentiment count}}{\text{total}}$
- $d = \frac{\text{negative sentiment count}}{\text{total}}$
- $u = \frac{\text{neutral + mixed sentiment count}}{\text{total}}$

The B, D,U tuple allows us to compute the **trust score** of each app. This Trust Score, for each app, is then calculated using:

$$\text{Trust Score} = b + \left(\frac{1}{n \cdot u}\right) \cdot 5$$

where $n = 3$ represents the number of sentiment categories (positive, negative, neutral). This trust score helps us to compare apps offering similar functionality.

E. Logic Criteria and Evaluation

To assess the quality of outputs generated by the three LLMs, we applied the filtering approaches shown in Table II to the review dataset. These approaches represent various scenarios that we experimented with. The approach 1 uses the entire dataset as the baseline while the approach 2 considers only those reviews where a majority consensus exists between the LLMs.

TABLE II
FILTERING APPROACHES

Approach Number	Filtering Applied	Purpose
1	No filter (entire dataset)	Baseline BDU and sentiment disagreement analysis across all data
2	At least 2 of 3 LLMs agree on sentiment	Analyze trust when partial LLM consensus exists
3	All three AI scores < 0.7	Filter out lower AI confidence and compute sentiment consistency
4	At least two AI scores < 0.7	Relaxed AI confidence threshold to assess partial trust
5	Average AI score < 0.7	Use average AI scores to examine sentiment distributions
6	Functional Review type	Analyze trust and sentiment only in Functional reviews
7	Non-Functional Review type	Analyze trust and sentiment only in Non-functional reviews

Recent studies have found that AI-generated reviews are increasingly prevalent—rising from approximately 5% in 2019 to nearly 19% by the end of 2024 on Google platforms [22]. Similar trends are observed elsewhere, with 26% of B2B software reviews likely AI-generated and 14% of consumer sector reviews identified as AI or bot-produced [23], [24]. Given this trend, applying AI-likelihood filtering helps ensure our dataset more accurately reflects genuine user sentiment.

To address this AI-generated aspect, approaches 3, 4, and 5 consider the AI-likelihood score for each review (as generated by LLMs) and filter the dataset using different thresholds. A threshold of 0.7 was chosen based on preliminary pilot testing and existing literature, where a 0.7 cutoff is commonly used as a conservative filter to eliminate likely AI-generated content [22] . Finally, approaches 6 and 7 focus on the functional and non-functional elements indicated in the reviews while filtering the dataset.

Once trust-based rankings were generated using each filtering approach, we compared these rankings to evaluate how much they differed. To do this, we applied Kendall Tau distance—a widely used metric that captures the extent of ranking disagreement between models and actual app store ratings.

F. Kendall Distance

The **Kendall distance** (also known as Kendall tau distance) is a metric used to measure the disagreement between two rankings. Specifically, it counts the number of pairwise disagreements (discordant pairs) between two ordered lists.

Formally, given two rankings X and Y, each containing n items, the Kendall distance is calculated as:

$$K(X,Y) = \sum_{1 \le i < j \le n} \bar{K}_{i,j}(X,Y)$$

where $\bar{K}_{i,j}(X,Y)$ is defined as:

$$\bar{K}_{i,j}(X,Y) = \begin{cases} 0, & \text{if the pair } (i,j) \text{ is ordered the same way in both } X \\ 1, & \text{if the pair } (i,j) \text{ is ordered differently in } X \text{ and } Y. \end{cases}$$

The **normalized Kendall distance** is given by dividing $K(X,Y)$ by the maximum possible number of discordant pairs, which is $\frac{n(n-1)}{2}$:

$$K_{\text{norm}}(X,Y) = \frac{2K(X,Y)}{n(n-1)}$$

A smaller Kendall distance indicates a similarity, while a larger value suggests a divergence between the rankings. Kendall distance is frequently employed in evaluating the consistency and agreement between ranking methods, as it captures the extent of discordance at a granular, pairwise level.

IV. EXPERIMENTS AND ANALYSIS

We conducted two significatnt experiments that focused on mobile games from the Google Play Store, specifically within the chess and snooker categories. Each experiment involved evaluating app reviews using a series of filtering approaches designed to assess the trustworthiness and sentiment quality of user feedback.

In Experiment 1, we analyzed five chess-related apps using the first five filtering strategies outlined in Table II. This experiment provided an initial assessment of how AI-likelihood scores and model agreement affect trust-based app rankings.

In Experiment 2, we expanded the scope by applying all seven filtering approaches, including those that differentiate between functional and non-functional review types. This extension allowed for a more nuanced evaluation of how review content influences app rankings.

We selected five popular apps from the **chess** and **snooker/pool** categories on the Google Play Store for our analysis. Each app met the following criteria: (i)

1) a minimum of 10000 publicly available user reviews,
2) an average rating above 4.0, and relevance to its respective genre (i.e., chess or snooker/pool).

These criteria ensured that the selected apps had substantial user engagement, were actively used, and offered similar

functionality within their categories, making them suitable for meaningful comparison using our sentiment and trust-based ranking approaches.

A. Experiment 1

We selected, as indicated above, five popular chess apps and retrieved their reviews from Google Play. Each review was processed through all three LLMs using a common prompt. The output was then filtered using the approaches described in Table II to evaluate LLM agreements and trustworthiness based on BDU and Trust Score metrics. We compare the results of rankings, obtained by applying various techniques, including the default rating in Google Play, using Kendall Tau distance.

The ratings of these five apps in Google Play are indicated below – we have anonymized the names of these apps. We refer this default rating as "Actual" from here onwards

- **ChessApp1**: 4.7
- **ChessApp2**: 4.6
- **ChessApp3**: 4.8
- **ChessApp4**: 4.5
- **ChessApp5**: 4.7

1) No filter (entire dataset): This approach serves as the baseline, with no filtering applied to the dataset. Sentiment labels were independently assigned by each of the three LLMs—Mistral, LLaMA, and Gemini. Based on these labels, trust scores were computed separately for each model to rank the apps. The resulting rankings are shown in Figure 2, where the blue line represents the original Google Play ranking based on average user star ratings.

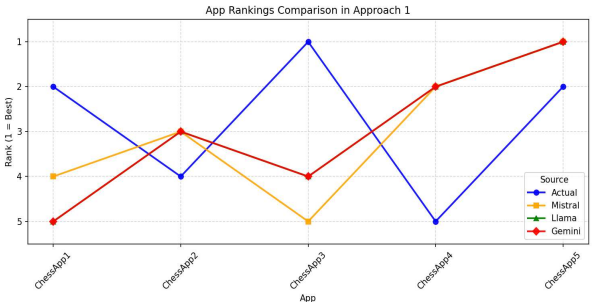

Fig. 2. No filter (entire dataset)

Table III reports the Kendall Tau distances between model-generated rankings and the actual Play Store rankings, as well as between the models themselves.

TABLE III
KENDALL DISTANCE BETWEEN MODELS AND ACTUAL (APPROACH 1)

Model Comparison	Distance
Mistral vs LLaMA	1
Mistral vs Gemini	1
LLaMA vs Gemini	0
Mistral vs Actual	7
Gemini vs Actual	6
LLaMA vs Actual	6

Discussion of Results: The Kendall Tau distance values indicate a high level of agreement among the LLMs themselves (distances of 0 or 1), suggesting that when a sentiment consensus exists across models, the resulting app trust rankings are largely consistent. However, the distances between the rankings produced by each model and the actual Play Store rankings are significantly larger (6 or 7), highlighting a clear misalignment between user-reported trust—captured through sentiment analysis—and the aggregate rating shown on the Google Play platform.

A closer examination reveals that this divergence is likely driven by how reviews are presented on the Google Play Store. When users open the review section, they are shown the "Relevant" reviews—typically the most detailed and upvoted. Our extraction and analysis of these top reviews revealed that, for several high-rated apps, a majority of these prominent reviews expressed dissatisfaction or frustration. When these reviews were passed through the LLM-based sentiment pipeline, the resulting trust scores reflected a predominantly negative sentiment, which significantly affected the app's position in our ranking.

This explains the large Kendall Tau distances observed between LLM-generated rankings and the Play Store's ranking order. While the average rating may remain high, user trust, as expressed in detailed, high-engagement reviews, tells a more critical story. This reinforces the need for alternate ranking approaches that give an additional weight to qualitative user feedback rather than relying solely on the aggregated star ratings.

2) At least 2 of 3 LLMs agree on the sentiment: In the second filtering approach, we included reviews where at least two out of the three LLMs agreed on the sentiment classification. This relaxed consensus criterion allowed for a broader inclusion of reviews compared to the full-consensus filtering in the Approach 1. Table IV summarizes the total reviews and the percentage of reviews dropped between the Approach 1 and Approach 2:

TABLE IV
PERCENTAGE OF REVIEWS DROPPED (APPROACH 1 TO APPROACH 2)

App Name	Dropped (%)
chessApp5	5.0%
chessApp4	8.2%
ChessApp3	6.3%
ChessApp2	6.4%
ChessApp1	10.5%

As seen in the table, this approach excluded only a small fraction of the reviews (between 5% and 10.5%), preserving most of the dataset while maintaining a reasonable agreement. The retained reviews were analyzed using the same trust-based methodology as Approach 1, and app rankings were generated for each LLM. Figure 3 shows the resulting rankings, and Table V presents the Kendall Tau distances between each model and the official Google Play ranking.

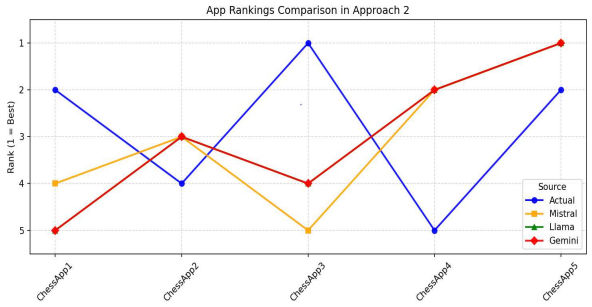

Fig. 3. At least 2 of 3 LLMs agree on sentiment

TABLE V
KENDALL DISTANCE BETWEEN MODELS AND ACTUAL (APPROACH 2)

Model Comparison	Distance
Mistral vs LLaMA	1
Mistral vs Gemini	1
LLaMA vs Gemini	0
Mistral vs Actual	7
Gemini vs Actual	6
LLaMA vs Actual	6

Discussion of Results: This approach produced results like the previous one (Table III), with very low Kendall Tau distances among the LLMs, indicating strong agreement when at least two models concur on the sentiment classification. However, the distance between each LLM's ranking and the actual Play Store ranking remained high, consistent with earlier observations. This reinforces our earlier observation (as seen in Approach 1) that there is a persistent disconnect between user-perceived trust—derived from sentiment analysis of detailed reviews and the aggregated star ratings presented on the Google Play platform.

3) All three LLM's AI scores < 0.7 *:* The third filtering approach focused on ensuring human-authored authenticity by including only those reviews where all three LLMs assigned an AI-likelihood score of less than 0.7. Applying this criterion resulted in a significant reduction in the dataset. The percentage of reviews dropped for each app, relative to Approach 1, is as shown in Table VI.

TABLE VI
PERCENTAGE OF REVIEWS DROPPED (AI SCORE < 0.7 FILTER – APPROACH 3)

App Name	Dropped (%)
chessApp5	47.2%
chessApp4	64.2%
ChessApp3	62.5%
ChessApp2	62.6%
ChessApp1	64.3%

Despite this substantial reduction, the remaining reviews were processed using the same trust score computation and ranking methodology. The results are shown in Figure 4, with the corresponding Kendall Tau distances provided in Table VII.

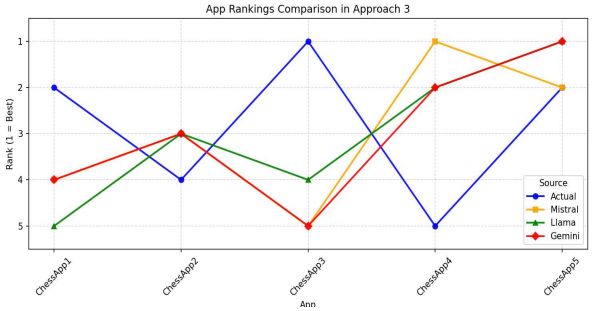

Fig. 4. All three LLM's AI scores < 0.7

TABLE VII
KENDALL DISTANCE BETWEEN MODELS AND ACTUAL (APPROACH 3)

Model Comparison	Distance
Mistral vs LLaMA	2
Mistral vs Gemini	1
LLaMA vs Gemini	1
Mistral vs Actual	8
Gemini vs Actual	7
LLaMA vs Actual	6

Discussion of Results: In this approach, we kept only the reviews that all three LLMs marked with AI-likelihood scores below 0.7. This reduced the dataset size significantly—over 60% of reviews were dropped for most apps. The goal was to focus on content that is likely written by humans. However, it is possible that some genuine human-written reviews were also removed if they showed patterns that looked AI-generated to the LLMs.

Even with fewer reviews, the LLMs showed strongly agreement in their rankings (Kendall distances of 1 or 2).

However, the disagreements between LLM rankings and Play Store ratings grew larger (distances of 6 to 8). This suggests that once the likely AI-written content is removed, the remaining user feedback often reveals concerns that are not reflected in the app's star rating, highlighting the need for trust-based ranking systems.

4) At least two LLM's AI scores < 0.7 *:* In the fourth filtering approach, we considered reviews for which **at least two out of the three LLMs** assigned an AI-likelihood score below 0.7. This approach aimed to strike a balance between data authenticity and coverage by relaxing the strict condition from the approach 3.

Table VIII shows the percentage of reviews dropped relative to the original dataset in the Approach 1:

TABLE VIII
PERCENTAGE OF REVIEWS DROPPED (APPROACH 1 TO APROACH 4: AI SCORE < 0.7 FOR AT LEAST 2 LLMS)

App Name	Dropped (%)
chessApp5	3.3%
chessApp4	13.3%
ChessApp3	10.3%
ChessApp2	6.2%
ChessApp1	10.1%

The results indicate that this condition is significantly less restrictive than Approach 3, with most apps retaining more than 85% of their reviews. These filtered reviews were then used to compute trust scores, generate app rankings per LLM, and compare those rankings against the Google Play order. The results are presented in Figure 5, and the Kendall Tau distances are reported in Table IX.

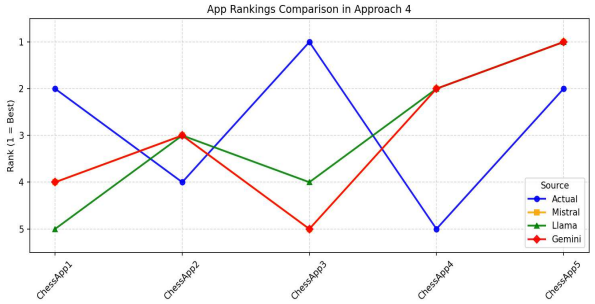

Fig. 5. At least two LLM's AI scores < 0.7

Discussion of Results: Relaxing the AI-score filter to include reviews where at least two LLMs assigned a score below 0.7 led to only a small drop in data (3%–13% across apps). This retained more user feedback while still aiming to exclude likely AI-generated reviews. Notably, many reviews dropped were flagged primarily by LLaMA, which appeared more conservative in detecting AI-like patterns.

TABLE IX
KENDALL DISTANCE BETWEEN MODELS AND ACTUAL (APPROACH 4)

Model Comparison	Distance
Mistral vs LLaMA	1
Mistral vs Gemini	0
LLaMA vs Gemini	1
Mistral vs Actual	7
Gemini vs Actual	7
LLaMA vs Actual	6

LLM agreement remained high (Kendall Tau distances of 0 or 1), showing stable trust rankings even with a more inclusive dataset. However, the disagreement between LLM-based rankings and Play Store ratings stayed large (distances of 6–7), reinforcing earlier findings: user sentiment in reviews often differs from star ratings. This again highlights the value of trust-based metrics in capturing deeper insights from review content.

5) Average AI Score < 0.7*:* In the fifth filtering approach, we calculated a **weighted average AI score** for each review, where the weights reflected the relative reputations of the LLMs. A review was selected if its weighted average AI score was less than 0.7.

We assigned **reputation weights** to each LLM based on its observed performance in distinguishing AI-generated from human-written text, using the benchmark dataset described in [25]. Each LLM was evaluated on a random sample of 5000 records from this dataset. The resulting reputation-based weights were:

- Gemini: 42%
- LLaMA: 43%
- Mistral: 48%

Although all LLMs performed below 50% accuracy on this specific benchmark, the scores served as a reference to compare their relative competence. We used these weights to compute a weighted average AI score for each review. This filtering approach not only improved the robustness of our trust estimation but also captured cross-model sentiment agreement more effectively.

Table X shows the percentage of reviews dropped due to the weighted AI score threshold (< 0.7):

TABLE X
PERCENTAGE OF REVIEWS DROPPED (WEIGHTED AVG. AI SCORE < 0.7 – APPROACH 5)

App Name	Dropped (%)
chessApp5	0.1%
chessApp4	4.7%
ChessApp3	4.6%
ChessApp2	2.9%
ChessApp1	3.4%

As seen, this filtering approach retained nearly all reviews across all apps. Rankings based on the filtered data are shown in Figure 6, with the corresponding Kendall Tau distances listed in Table XI.

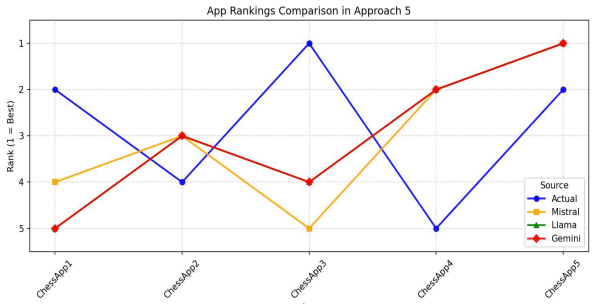

Fig. 6. Average AI score < 0.7

TABLE XI
KENDALL DISTANCE BETWEEN MODELS AND ACTUAL (APPROACH 5)

Model Comparison	Distance
Mistral vs LLaMA	1
Mistral vs Gemini	1
LLaMA vs Gemini	0
Mistral vs Actual	7
Gemini vs Actual	6
LLaMA vs Actual	6

Discussion of Results: By using reputation-weighted AI scores, this approach retained nearly all reviews—dropping less than 5%—making it the most inclusive filtering scenario. The LLMs continued to show strong agreement in their rankings (low Kendall Tau distances), but their results still differed significantly from the Play Store rankings. This suggests that even with refined filtering, user sentiment often tells a different story than the average star rating, reinforcing the value of trust-based ranking systems.

B. Insights from Experiment 1

Applying the five filtering approaches described earlier produced varying types of app rankings. In the case of chess apps, we observed that the LLM-generated rankings were often consistent across models, with low Kendall Tau distances indicating minimal disagreement. These minor differences largely stemmed from sentiment classification inconsistencies—some models labeled a review as neutral, while others classified it as positive. For instance, one review stating, "Good game but crashes occasionally," was interpreted as neutral by LLaMA but as positive by Mistral and Gemini. Such discrepancies affected app positions slightly, typically in the 10–20

However, when comparing LLM-based rankings to the official Google Play Store ratings—particularly using Approach 1 (no filtering beyond consensus on sentiment)—clear divergences emerged. For example, **ChessApp3** had the highest Play Store rating (4.8), but was ranked the lowest by all three LLMs. Upon inspecting a sample of its reviews, we found that over 50 were classified as negative. This strongly influenced the LLM-derived trust score and lowered the app's position in our rankings. This result aligns with prior findings in [1], which noted that app ratings and review sentiments are frequently inconsistent.

One plausible reason for **ChessApp3**'s high public rating—despite its negative sentiment distribution—may be its developer engagement. Our review inspection revealed that the developers responded to nearly every negative comment, acknowledging issues or offering solutions. This likely enhanced the app's perceived credibility and reliability among users, which could explain the high Play Store score.

In contrast, apps such as **ChessApp1** and **ChessApp3**, which had a higher proportion of positive sentiment in reviews but lower official ratings, showed little or no developer interaction. This contrast suggests that app ratings are shaped not only by user sentiment but also by how developers respond to feedback.

These observations highlight a critical insite – i.e., app ratings do not solely reflect the user sentiment. Other important factors such as responsiveness of developers, frequency and quality of updates, and transparency about support shape user sentiments and perceptions. Hence, a comprehensive rating system will need to capture all these dimensions, else any such system will mislead the users.

C. Experiment 2

This experiment focuses on a different category of mobile apps, snooker-related games, to evaluate whether our approach of developing a trust-based ranking framework generalizes beyond the chess domain. We apply the same five filtering approaches used in Experiment 1, along with two additional steps that analyze review types (functional and non-functional). These extensions allow for a more detailed investigation of how review content influences app trust rankings across different genres. We, again, have selected the following five snooker games (anonymized) for the second experiment – their "actual" ratings in Google Play are indicated as well.

- **SnookerApp1**: 4.1
- **SnookerApp2**: 4.5
- **SnookerApp3**: 4.3
- **SnookerApp4**: 4.7

1) No filter (entire dataset): In the first approach, we assigned sentiment labels to reviews of snooker apps using all three LLMs. Without applying any filtering, we computed the BDU scores for each review. This provided an initial baseline for trust score computation and app ranking. Figure 9 shows the results for this base case

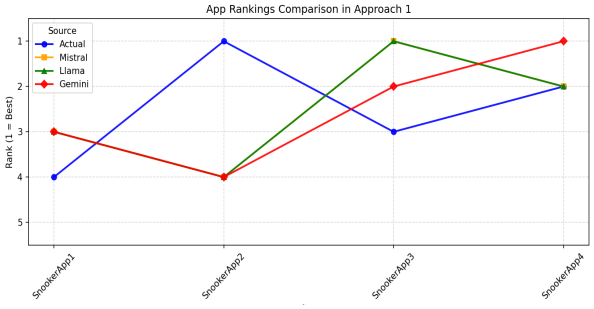

Fig. 7. No filter (entire dataset)

Table XII reports the Kendall Tau distances between LLM generated rankings and the actual Play Store rankings, as well as between the LLMs themselves.

TABLE XII
KENDALL DISTANCE BETWEEN MODELS AND ACTUAL (APPROACH 1)

Model Comparison	Distance
Mistral vs LLaMA	0
Mistral vs Gemini	1
LLaMA vs Gemini	1
Mistral vs Actual	4
Gemini vs Actual	3
LLaMA vs Actual	4

Discussion of Results: Consistent with findings from the chess app experiment, the LLMs exhibited strong agreement with each other, as reflected by low Kendall Tau distances. This underscores the stability of sentiment classification across different LLM models. However, a persistent mismatch remains between LLM-based rankings as shown in Table XII

Upon reviewing samples labeled as negative by all LLMs, we observed that many of these reviews expressed specific critiques in an otherwise constructive tone—for example: `"Decent game. Balls roll way too far. Too many ads with no option to just pay a bit of money to avoid watching several ads between games. Room for improvement if the developers decide to update the game."` Despite such a balanced view, all LLMs classified such feedback as negative due to the emphasis on shortcomings.

We also noted instances where reviews offering mixed feedback or user suggestions were labeled as neutral. For example: `"You leave, come back an hour later or more, and the other player is still there. Definitely a bot. Rules also have problems as they differ from real snooker. After trying many of the same games..."` In such cases, LLMs tended to split sentiment, effectively giving a partial weight to the concerns raised.

These observations suggest that LLMs interpret critique-heavy feedback as negative, even when it is framed constructively, and treat mixed feedback with diluted sentiment. This behavior influences trust scoring and contributes to the observed misalignment with platform ratings.

2) At least 2 of 3 LLMs agree on sentiment: In the second approach, again we retained reviews for which at least two out of the three LLMs agreed on the sentiment classification (positive, negative, or neutral). This relaxed consensus criterion decreased the dataset size compared to the full agreement case, while still maintaining a reasonable level of confidence in sentiment consistency.

Table XIII shows the percentage of reviews dropped for each snooker app relative to the total number of reviews in approach 1.

TABLE XIII
PERCENTAGE OF REVIEWS DROPPED (APPROACH 1 TO APPROACH 2: AT LEAST 2 OF 3 LLMS AGREE ON SENTIMENT)

App Name	Dropped (%)
SnookerApp1	13.8%
SnookerApp4	8.5%
SnookerApp3	8.0%
SnookerApp2	7.1%

These reviews were then processed using the same BDU-based trust score methodology as in approach 1. Figure 9 displays the rankings generated from this filtered set.

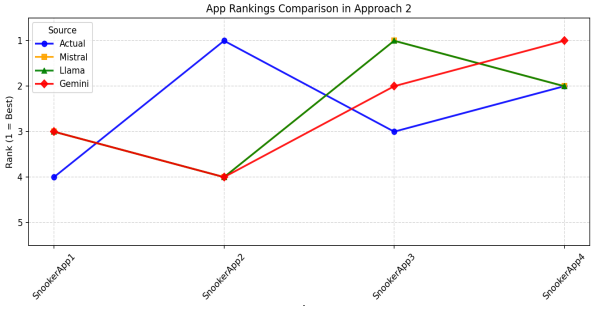

Fig. 8. At least 2 of 3 LLMs agree on sentiment

The Kendall Tau distances in Table XIV again show strong alignment between the LLMs, with low inter-LLM distances. However, moderate distances between each LLM and the actual Google Play ranking indicate a continued divergence between user sentiment and official ratings.

Discussion of Results: By allowing partial agreement (2 out of 3 LLMs), we retained a larger portion of the dataset. The small distances between model rankings confirm that even under relaxed conditions, LLMs remain consistent in their

TABLE XIV
KENDALL DISTANCE BETWEEN MODELS AND ACTUAL (APPROACH 2)

Model Comparison	Distance
Mistral vs LLaMA	0
Mistral vs Gemini	1
LLaMA vs Gemini	1
Mistral vs Actual	4
Gemini vs Actual	3
LLaMA vs Actual	4

view about the reciews. However, noticeable disagreement still exist between LLM-based rankings and app store ratings, reinforcing the value of sentiment-driven trust scoring as an alternative to star ratings.

3) All three LLM's AI scores < 0.7 *:* In the third approach again, we filtered out potentially AI-generated reviews by retaining only those for which all three LLMs assigned an AI-likelihood score below 0.7.

Table XV presents the percentage of reviews dropped under this approach relative to the full dataset in Approach 1. Notably, SnookerApp1 saw a significant reduction in the number od reviews, while the other three apps experienced minimal loss reviews.

TABLE XV
PERCENTAGE OF REVIEWS DROPPED (TOTAL RECORDS TO APPROACH 3: ALL THREE LLMS AI SCORES < 0.7)

App Name	Dropped (%)
SnookerApp1	35.5%
SnookerApp4	8.8%
SnookerApp3	8.0%
SnookerApp2	10.5%

Figure 9 shows the app ranking results after applying this filter.

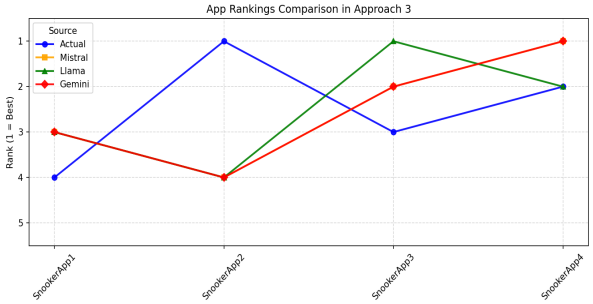

Fig. 9. All three LLM's AI scores < 0.7

The corresponding Kendall Tau distances in Table XVI again demonstrate high agreement among LLMs (distances of 0 or 1). Although slightly improved alignment is observed between the actual Google Play rankings and the three LLMs, meaningful gaps still persist.

TABLE XVI
KENDALL DISTANCE BETWEEN MODELS AND ACTUAL (APPROACH 3)

Model Comparison	Distance
Mistral vs LLaMA	1
Mistral vs Gemini	0
LLaMA vs Gemini	1
Mistral vs Actual	3
Gemini vs Actual	3
LLaMA vs Actual	4

Discussion of Results: The LLMs again showed strong agreement with each other while ranking the app, LLMs rankings moved slightly closer to the app store ratings—but differences were still noticeable. For the SnookerApp1 app, we found that Mistral flagged around 250 reviews as likely AI-generated. When we manually read these dropped reviews, many showed the features that we had included in our prompt, such as repeated phrases, unclear context, or generic writing. This suggests that removing likely AI-generated reviews helps improve the quality of the results. Still, the differences between trust-based rankings and Play Store ratings show that user star ratings may not always reflect what's actually being said in the reviews.

4) At least two LLM's AI scores < 0.7 *:* In this fourth approach, we again adopted a slightly flexible criterion for the review authenticity by including reviews for which at least two of the three LLMs assigned an AI-likelihood score below 0.7.

As shown in Table XVII, the percentage of dropped reviews for this case was minimal—under 9 percent for all apps—making this one of the most inclusive filters while still maintaining a focus on review authenticity.

TABLE XVII
PERCENTAGE OF REVIEWS DROPPED (TOTAL RECORDS TO APPROACH 4: AT LEAST TWO LLMS AI SCORES < 0.7)

App Name	Dropped (%)
SnookerApp1	8.3%
SnookerApp4	1.6%
SnookerApp3	0.8%
SnookerApp2	1.3%

Figure 10 displays the app rankings produced under this criterion.

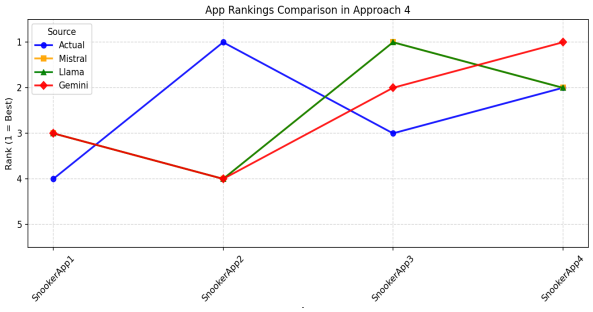

Fig. 10. At least two LLM's AI scores < 0.7

The Kendall Tau distances in Table XVIII again reflect high inter-LLM agreement (0 or 1), while the misalignment between LLM-generated trust ranking and actual app store ranking persisted.

TABLE XVIII
KENDALL DISTANCE BETWEEN MODELS AND ACTUAL (APPROACH 4)

Model Comparison	Distance
Mistral vs LLaMA	0
Mistral vs Gemini	1
LLaMA vs Gemini	1
Mistral vs Actual	4
Gemini vs Actual	3
LLaMA vs Actual	4

Discussion of Results: This approach offered a practical middle ground—preserving most of the review dataset while still screening for likely human-authored feedback. The LLMs continued to show consistent sentiment evaluations, and while some distance remained from actual app ranking, the results point to the robustness of the trust-based framework even with a relaxed filtering approach.

5) Average AI score < 0.7 : In the fifth approach again, we calculated the weighted average AI-likelihood score across all three LLMs for each review. The weights used were the same as those applied in Experiment 1 (Mistral: 48%, LLaMA: 43%, Gemini: 42%), based on their relative performance in detecting AI-generated content. If the resulting average score was less than 0.7, the review was retained for further analysis. This strategy enabled a balanced and inclusive filtering process by incorporating both model confidence and consensus-based assessment of review authenticity

As shown in Table XIX, this method resulted in minimal data loss, with fewer than 1% of reviews dropped for any app. It is the most inclusive of all AI-score-based filtering steps.

TABLE XIX
PERCENTAGE OF REVIEWS DROPPED (TOTAL RECORDS TO APPROACH 5: AVERAGE AI SCORE < 0.7)

App Name	Dropped (%)
SnookerApp1	0.5%
SnookerApp4	0.6%
SnookerApp3	0.0%
SnookerApp2	0.1%

Figure 11 shows the resulting rankings under this criterion.

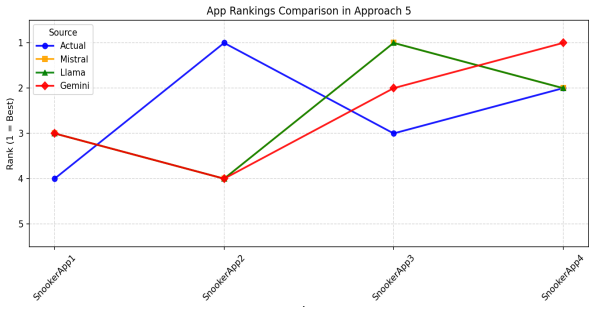

Fig. 11. Average AI score < 0.7

TABLE XX
KENDALL DISTANCE BETWEEN MODELS AND ACTUAL (APPROACH 5)

Model Comparison	Distance
Mistral vs LLaMA	0
Mistral vs Gemini	1
LLaMA vs Gemini	1
Mistral vs Actual	4
Gemini vs Actual	3
LLaMA vs Actual	4

Discussion of Results: This approach kept almost all reviews, removing less than 1% across the apps. The LLMs still agreed closely with each other, showing consistent sentiment judgments. However, their rankings,as seen in Table XX continued to differ from the Play Store-based rankings. This suggests that even with minimal filtering, trust-based analysis can uncover user feedback patterns that star ratings might miss.

6) Review type = Functional: In this experiment, we introduced two additional filters to evaluate app rankings from a user-centric perspective. Specifically, this approach focused exclusively on functional reviews—feedback pertaining to core app features such as responsiveness, stability, interface usability, and in-app performance. These reviews are typically more objective and technically grounded. For instance, users might report issues like "the app crashes on startup" or "the login feature doesn't work after the update," which reflect observable problems. In contrast, subjective preferences such as "I don't like the color scheme" or "the game feels boring" offer limited insight into the app's actual functionality. As such, functional

reviews serve as a valuable subset for assessing app quality from the standpoint of real user experience.

The filtered functional reviews were scored using our approach trust framework based on outputs from Mistral, Gemini, and LLaMA. Figure 12 shows the resulting app rankings, and Table XXI presents the corresponding Kendall Tau distances from actual Google Play ratings.

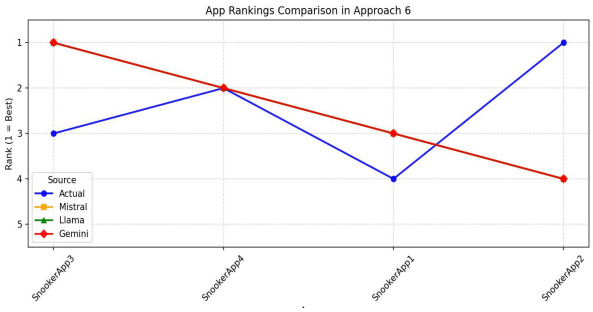

Fig. 12. Functional reviews-based Rankings

The corresponding Kendall Tau distances between each model's rankings and actual Google Play rankings are shown in Table XXI.

TABLE XXI
KENDALL DISTANCE BETWEEN MODELS AND ACTUAL (APPROACH 6)

Model Comparison	Distance
Mistral vs LLaMA	0
Mistral vs Gemini	0
LLaMA vs Gemini	0
Mistral vs Actual	4
Gemini vs Actual	4
LLaMA vs Actual	4

Discussion of Results: All three LLMs produced identical sentiment classifications for functional reviews, indicating strong inter-LLM consistency. However, our trust-based rankings remained misaligned with the actual app store ratings. This suggests that while functional feedback provides stable and objective insights into app performance, such sentiment is not fully reflected in aggregate star ratings. These findings, again underscore the importance of incorporating trust-based analysis to capture deeper dimensions of user experience beyond surface-level metrics.

7) Review type = Non-functionals: In the final approach, we filtered and analyzed only the **non-functional reviews**—those that reflect user opinions unrelated to core app performance. Examples of such reviews include statements like "I love the design of the app," or "This game is really fun but not addictive." These reviews are often more subjective and varied, making them less reliable for measuring app quality but important for understanding the overall user sentiment.

The filtered reviews were scored using the same trust evaluation process as in previous steps. Figure 13 illustrates the resulting rankings for the four snooker apps. The corresponding Kendall Tau distances between the models and the actual Play Store rankings are provided in Table XXII.

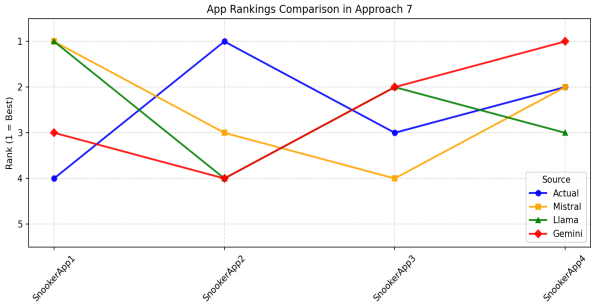

Fig. 13. Non-functionals

TABLE XXII
KENDALL DISTANCE BETWEEN MODELS AND ACTUAL (APPROACH 7)

Model Comparison	Distance
Mistral vs LLaMA	1
Mistral vs Gemini	3
LLaMA vs Gemini	3
Mistral vs Actual	4
Gemini vs Actual	3
LLaMA vs Actual	6

Discussion of Results: Compared to functional reviews, non-functional feedback led to greater divergence among LLMs, particularly between Gemini and the others (distance = 3). This variation likely stems from the subjective nature of non-functional sentiment, where expressions are more ambiguous and open to interpretation. Additionally, distances between LLM rankings and actual app ratings remained moderate to high (up to 6), underscoring again that app store ratings may not accurately reflect the sentiment trends found in user feedback. This reinforces the value of a trust-based approach in interpreting nuanced user preferences beyond just quantitative ratings.

D. Insights from Experiment 2

The goal of Experiment2 was to validate our methodology in a different app category—snooker-related mobile games—and assess whether the insights from chess apps generalized to this new context. While the overall patterns were consistent, a key difference was that the AI-based filters removed fewer reviews in this experiment, particularly in approaches 4 and 5. Upon manual inspection, many of the reviews flagged as AI-generated did not exhibit typical AI-like phrasing as defined in the prompt, suggesting that the filters may be overly conservative in this domain.

We also introduced two additional filters based on review type: functional and non-functional. These helped distinguish between feedback grounded in app performance (e.g., crashes, interface issues, etc.) versus subjective impressions (e.g., theme, enjoyment, etc.). Although these steps were exploratory, they lay the foundation for deeper, review-type-specific trust assessments in future work.

As in Experiment 2, the Kendall Tau distances between LLM-derived rankings and actual Google Play rankings remained moderate (typically 3 to 4), reinforcing the observation that platform star ratings do not fully reflect user sentiment. Reviews with detailed explanations and high engagement often portrayed sentiment that diverged from the app's star rating. This highlights the potential of sentiment-based trust scoring to provide a more nuanced and user-aligned view of app quality.

V. Limitations

One of the main challenges we faced during this process was related to both computational and security constraints. Due to organizational security policies, we were unable to install the Ollama package, which could have enabled more efficient execution of multiple LLMs. Additionally, limited computational resources required us to install and run one LLM at a time. Once one LLM finished processing the dataset, the next one had to wait for its turn to process the same data. This sequential execution significantly increased the overall processing time and limited our ability to scale the analysis or perform parallel experiments effectively. The other limitation is that we focused only a few apps from on two categories of and that too using a moderate number of reviews of those apps.

VI. Conclusion and Future Work

Mobile app ratings are influenced by numerous factors, many of which may not directly reflect actual user satisfaction. Elements such as appealing visuals or promotional content can boost ratings, while the detailed content of user reviews might indicate a different reality. In our analyses, we observed several instances where apps maintained high ratings despite having a substantial proportion of negative user reviews, highlighting a significant inconsistency.

To address this issue, we introduced an approach where multiple LLMs independently evaluate user sentiments from reviews. By integrating these assessments using a set of filtering approaches, we computed a holistic trust score for each app. That score was used to rank similar apps. Our method provides deeper insights into perception about apps, as indicated by the users, and offers a comprehensive picture of user preferences. In addition, since our method allows comparing different apps which offer a similar functionality, it also empowers users to select a particular app, for their needs, from a set of available choices instead of solely relying on the default star ratings available in the play store.

We plan to extend our method by incorporating review content categorization into **functional** (e.g., app features, stability, performance, etc.) and **non-functional** (e.g., user interface, speed, battery usage, customer support, etc.) aspects. These dimensions will help differentiate reviews based on the type of user concern. We will also explore opportunities for real-time sentiment analysis, creating a more dynamic and responsive system for evaluating and ranking mobile applications.

VII. Declarations

Funding: This research received no external funding.
Ethical approval: Not applicable. This study analyzes publicly available app reviews and does not involve human participants, patient data, or any intervention.
Informed consent: Not applicable.
Author Contributions:

- **Muhammad Hassam Aslam Khan**: Conceptualization, data collection, software implementation, experiments, analysis, and drafting the manuscript.
- **Rajeev R. Raje**: Conceptualization, methodology guidance, supervision, review and project oversight.

Data Availability Statement: The data used in this study were collected from publicly available app reviews. The processed datasets and scripts can be made available by the corresponding author upon reasonable request.
Conflict of Interest: The authors declare that they have no conflict of interest.
Clinical trial number: Clinical trial number: not applicable.

References

[1] Nahida Chowdhury, Ayush Maharjan, and Rajeev Raje. E-sers: An enhanced approach to trust-based ranking of apps. *Software*, 3:250–270, 07 2024.

[2] Chess.com. Chess – Play Learn [Mobile application software]. https://play.google.com/store/apps/details?id=com.chess, 2025. Accessed: 2025-06-24.

[3] AI Factory Limited. Chess [Mobile application software]. https://play.google.com/store/apps/details?id=uk.co.aifactory.chess, 2025. Accessed: 2025-06-24.

[4] Ehsan Noei and Kelly Lyons. A survey of utilizing user-reviews posted on google play store. In *Proceedings of the 29th Annual International Conference on Computer Science and Software Engineering*, CASCON '19, page 54–63, USA, 2019. IBM Corp.

[5] Gayatree Ganu, Noémie Elhadad, and Amélie Marian. Beyond the stars: Improving rating predictions using review text content. In *12th International Workshop on the Web and Databases (WebDB)*, 2013.

[6] Nahida Sultana Chowdhury and Rajeev R. Raje. A holistic ranking scheme for apps. In *2018 21st International Conference of Computer and Information Technology (ICCIT)*, pages 1–6, 2018.

[7] AppTweak. A comprehensive guide to google play store optimization. https://www.apptweak.com/en/aso-blog/aso-for-google-play-app-store-optimization-guide-for-android, August 2023. Accessed: 2025-05-03.

[8] Ehsan Noei, Feng Zhang, and Ying Zou. Too many user-reviews! what should app developers look at first? *IEEE Transactions on Software Engineering*, 47(2):367–378, 2021.

[9] Emitza Guzman and Walid Maalej. How do users like this feature? a fine grained sentiment analysis of app reviews. In *2014 IEEE 22nd International Requirements Engineering Conference (RE)*, pages 153–162, 2014.

[10] Sebastiano Panichella, Andrea Di Sorbo, Emitza Guzman, Corrado A. Visaggio, Gerardo Canfora, and Harald C. Gall. How can i improve my app? classifying user reviews for software maintenance and evolution. In *2015 IEEE International Conference on Software Maintenance and Evolution (ICSME)*, pages 281–290, 2015.

[11] Rosa Eliviani and Dwi Diana Wazaumi. Exploring sentiment trends: Deep learning analysis of social media reviews on google play store by netizens. *International Journal of Advances in Data and Information Systems*, 5(1):62–70, Mar. 2024. Trans.

[12] Junichiro Niimi. Dynamic sentiment analysis with local large language models using majority voting: A study on factors affecting restaurant evaluation. *arXiv preprint arXiv:2407.13069*, 2024. Evaluates LLaMA-2/3 models for sentiment classification of reviews.

[13] Audun Jøsang. *Subjective Logic: A Formalism for Reasoning Under Uncertainty*. Springer, 2016. Foundational book on subjective logic and trust modeling.

[14] Christian Esposito, Antonio Galli, Vincenzo Moscato, and Giancarlo Sperlí. Multi-criteria assessment of user trust in social reviewing systems with subjective logic fusion. *Inf. Fusion*, 77:1–18, 2022.

[15] Srijan Kumar, Bryan Hooi, Disha Makhija, Mohit Kumar, Christos Faloutsos, and V. Subrahamanian. Fairjudge: Trustworthy user prediction in rating platforms. 03 2017.

[16] Junichiro Niimi. Dynamic sentiment analysis with local large language models using majority voting: A study on factors affecting restaurant evaluation, 2024.

[17] Gemini Team. Gemini family: Highly capable multimodal models. https://ai.google.dev/gemini, 2024. Accessed: 2025-05-03.

[18] Mistral AI Team. Announcing mistral 7b: high-performance 7b-parameter llm. https://mistral.ai/news/announcing-mistral-7b, 2023. Accessed: 2025-06-24.

[19] Edwin van Teijlingen and Vanora Hundley. The importance of pilot studies. *Social Research Update*, (35), 2001.

[20] Adaku Uchendu, Thai Le, Kai Shu, and Dongwon Lee. Authorship attribution for neural text generation. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8384–8395, Online, November 2020. Association for Computational Linguistics.

[21] Boris Škorić, Sebastiaan J.A. de Hoogh, and Nicola Zannone. Flow-based reputation with uncertainty: evidence-based subjective logic. *International Journal of Information Security*, 15(4):381–402, 2016.

[22] Originality.AI. From 2019 to 2024: Ai-generated google reviews increased from 5% to 19%. https://originality.ai/blog/ai-google-reviews-study. Accessed July 2025.

[23] Originality.AI. More than 26% of g2 reviews are ai-generated since chatgpt. https://originality.ai/blog/g2-reviews-ai-generated-case-study. Accessed July 2025.

[24] The Transparency Company via FastCompany. Fake ai-generated reviews are all over the internet, research shows. https://www.fastcompany.com/91252048/fake-ai-generated-reviews-internet-research. Accessed July 2025.

[25] Shayan Gerami. AI vs Human Text Dataset, 2024. Accessed: 2025-05-11.