future internet

MDPI

Article

A Large-Scale Empirical Study of LLM Orchestration and Ensemble Strategies for Sentiment Analysis in Recommender Systems

Konstantinos I. Roumeliotis [1,2,*], Dionisis Margaris [3], Dimitris Spiliotopoulos [2] and Costas Vassilakis [1]

[1] Department of Informatics and Telecommunications, University of the Peloponnese, 221 31 Tripoli, Greece; costas@uop.gr
[2] Department of Management Science and Technology, University of the Peloponnese, Sehi Location (Former 4th Shooting Range), 221 31 Tripoli, Greece; dspiliot@uop.gr
[3] Department of Digital Systems, University of the Peloponnese, Valioti's Building, Kladas, 231 00 Sparta, Greece; margaris@uop.gr
* Correspondence: k.roumeliotis@uop.gr

Abstract

This paper presents a comprehensive empirical evaluation comparing meta-model aggregation strategies with traditional ensemble methods and standalone models for sentiment analysis in recommender systems beyond standalone large language model (LLM) performance. We investigate whether aggregating multiple LLMs through a reasoning-based meta-model provides measurable performance advantages over individual models and standard statistical aggregation approaches in zero-shot sentiment classification. Using a balanced dataset of 5000 verified Amazon purchase reviews (1000 reviews per rating category from 1 to 5 stars, sampled via two-stage stratified sampling across five product categories), we evaluate 12 different leading pre-trained LLMs from four major providers (OpenAI, Anthropic, Google, and DeepSeek) in both standalone and meta-model configurations. Our experimental design systematically compares individual model performance against GPT-based meta-model aggregation and traditional ensemble baselines (majority voting, mean aggregation). Results show statistically significant improvements (McNemar's test, $p < 0.001$): the GPT-5 meta-model achieves 71.40% accuracy (10.15 percentage point improvement over the 61.25% individual model average), while the GPT-5 mini meta-model reaches 70.32% (9.07 percentage point improvement). These observed improvements surpass traditional ensemble methods (majority voting: 62.64%; mean aggregation: 62.96%), suggesting potential value in meta-model aggregation for sentiment analysis tasks. Our analysis reveals empirical patterns including neutral sentiment classification challenges (3-star ratings show 64.83% failure rates across models), model influence hierarchies, and cost-accuracy trade-offs ($130.45 aggregation cost vs. $0.24–$43.97 for individual models per 5000 predictions). This work provides evidence-based insights into the comparative effectiveness of LLM aggregation strategies in recommender systems, demonstrating that meta-model aggregation with natural language reasoning capabilities achieves measurable performance gains beyond statistical aggregation alone.

Keywords: large language models; zero-shot sentiment analysis; ensemble learning; recommender systems; meta-model aggregation; LLM orchestration; sentiment classification; multi-model aggregation; reasoning models; empirical evaluation

Academic Editors: Stanimir Stoyanov, Asya Stoyanova Doycheva and Veneta Tabakova-Komsalova

Received: 23 December 2025
Revised: 10 February 2026
Accepted: 16 February 2026
Published: 20 February 2026

1. Introduction

One of the major factors in modern digital platforms, nowadays, is undoubtedly recommender systems (RecSys). RecSys enable personalized experiences across everyday aspects of life, from entertainment and social media to employment search and e-commerce. Interpreting user feedback represents a critical challenge in RecSys research. Research in sentiment analysis may prove extremely useful, since, on the one hand, it can contribute to understanding customer feedback and reviews about products they have already seen, purchased, or listened to; on the other hand, it can help predict user satisfaction with products they have not yet experienced [1].

This work focuses specifically on rating prediction from review text, a fundamental component of modern recommender systems. While a complete RecSys encompasses multiple interconnected modules (user profiling, item representation, ranking algorithms, cold-start handling), rating prediction serves as a critical building block that directly influences recommendation quality in collaborative filtering, hybrid systems, and content-based approaches [1,2]. Our investigation of LLM-based rating prediction and meta-model aggregation addresses a core RecSys challenge: accurately inferring user preferences from textual feedback, which subsequently feeds into downstream recommendation algorithms. This component-focused approach aligns with established RecSys literature, where specialized studies on rating prediction, review mining, and sentiment-aware recommendation have consistently contributed to advancing the field's understanding of preference modeling [3,4].

In this direction, the rapid evolution of LLMs has caused a major shift in sentiment classification, moving from traditional feature-based machine learning approaches to more advanced, context-aware reasoning systems that can handle complex human language [2–4].

While recent approaches to RecSys sentiment analysis focus on enhancing model architectures or training data quality [5,6], an emerging direction explores ensemble approaches and meta-model aggregation strategies for combining multiple models [7]. Two evaluation approaches are relevant: independent LLMs operating individually on classification tasks, and meta-model aggregation systems that combine predictions from multiple independent LLMs through statistical or reasoning-based methods [8,9]. Despite growing interest in these aggregation strategies, systematic comparative evaluations of their effectiveness in real-world applications—such as sentiment prediction in RecSys—remain limited.

In contrast to prior studies that typically evaluate a small number of LLMs [10]—often fine-tuned for specific tasks—this study examines 12 leading pre-trained models across four major providers (OpenAI, Anthropic, Google, DeepSeek) in a zero-shot setting. This cross-company comparative design enables systematic analysis of architectural differences, provider-specific biases, and performance heterogeneity across model families, providing insights into which design paradigms (reasoning-optimized vs. chat-oriented architectures) best capture sentiment nuances without task-specific adaptation.

This focus on zero-shot evaluation aligns with a growing trend in the AI community, where labeled data for specific tasks is often limited, and contemporary LLMs are increasingly designed with advanced capabilities—including self-improvement through iterative refinement and instruction-following abilities based on their pre-trained knowledge [11,12]. In order to evaluate the baseline capabilities of these models, which are critical for determining their potential in real-world applications and for downstream adaptation via few-shot learning or instruction tuning, it is essential to evaluate these models in their default and unmodified state.

This empirical evaluation examines all 12 pre-trained models in both standalone and aggregated configurations. In the aggregation approach, a GPT-based meta-model

processes predictions from all base models alongside the original review text to produce a final classification. This experimental design enables direct comparison of individual model performance against meta-model aggregation strategies and traditional ensemble baselines (majority voting, mean aggregation), investigating whether reasoning-based aggregation provides measurable accuracy improvements over statistical combination methods.

To ensure reliable sentiment signals from authentic customer experiences, we created a balanced dataset of 5000 verified reviews from the Amazon Reviews '23 dataset [13], where each LLM predicted sentiment ratings using combined title and text inputs in a zero-shot setting.

By comparing standalone LLMs, traditional ensemble methods, and meta-model aggregation across key metrics—accuracy, precision, recall, F1-score, and computational cost—this study investigates which strategies provide measurable performance improvements and whether observed gains justify increased operational complexity.

The evaluation objectives of this research are threefold. First, we measure the performance of 12 leading LLMs in zero-shot review sentiment classification, establishing baseline capabilities without task-specific fine-tuning. Second, empirically compare whether meta-model aggregation with natural language reasoning capabilities yields accuracy improvements over standalone models and traditional ensemble baselines. Third, we investigate several research questions (RQs) that remain underexamined in the literature, particularly concerning model behavior, aggregation effectiveness, and practical trade-offs, including:

1. Which LLM achieves the highest accuracy in zero-shot sentiment classification?
2. At which rating levels (1–5) do LLMs most frequently make incorrect predictions, and at which levels do they perform most accurately?
3. How does meta-model aggregation compare to traditional ensemble baselines in terms of observed accuracy improvements?
4. Which LLMs contribute positively or negatively to the meta-model's performance?
5. How does the meta-model handle independent models' recommendations—specifically, how frequently does it accept (revise) versus disregard them, and are these decisions beneficial?
6. Which models most strongly influence the meta-model's decisions, and which are the least trusted?
7. How does a model's performance as a meta-model (in a meta-model aggregation system setup) compare to its performance when operating independently?
8. Does the meta-model primarily rely on majority voting, or does it reason directly over textual content? How often does each occur?
9. Which models behave as outliers, potentially disrupting the overall meta-model aggregation system?
10. Are pre-trained LLMs without fine-tuning capable of accurately capturing sentiment from user feedback?
11. Do observed accuracy improvements justify the added computational complexity and cost of meta-model aggregation?
12. How close are the predictions of different LLMs, and can we reduce costs by omitting models with similar outcomes?
13. How does the performance of zero-shot LLMs and the meta-model aggregation approach compare to traditional fine-tuned transformer models?
14. What patterns of agreement exist among the 12 base models, and how does the consensus level relate to prediction accuracy?
15. Finally, under what conditions do meta-models override the majority vote, and how does this relate to prediction accuracy?

The empirical findings provide evidence-based insights into the comparative effectiveness of different aggregation strategies for sentiment analysis in recommender systems, informing practitioners about potential benefits and limitations of meta-model aggregation approaches.

The rest of the paper is structured as follows: in Section 2, the contemporary literature review is summarized. In Section 3, we first present the dataset selection and preprocessing procedures and then we introduce the model deployment and evaluation procedures. In Section 4, the experimental findings are presented, and, lastly, Section 5 discusses this study.

2. Literature Review

This section reviews the literature across three interconnected domains that inform our study: sentiment analysis in recommender systems (Section 2.1), ensemble and aggregation methods in NLP and machine learning (Section 2.2), and emerging LLM collaboration frameworks (Section 2.3). We conclude by positioning our contribution within these established research streams (Section 2.4).

2.1. Sentiment Analysis in Recommender Systems

Sentiment analysis is increasingly applied in RecSys to capture emotional responses and better tailor suggestions based on user sentiments [14–16]. Recent research explores novel approaches such as emotion-aware models, which improve the prediction accuracy of personalized recommendations by considering not only user preferences but also emotional states.

The work in [17] introduces SAMF, a recommendation system combining sentiment analysis and matrix factorization to address data sparsity and credibility in collaborative filtering (CF). By using Latent Dirichlet Allocation for topic modeling and BERT (Bidirectional Encoder Representations from Transformers) for sentiment analysis, the system enhances the rating matrix with implicit information from reviews. Experiments on Amazon datasets show that SAMF outperforms traditional algorithms in recommendation performance. The work in [18] introduces a RecSys that combines sentiment analysis using the Bi-LSTM deep learning model with CF techniques in the e-commerce domain. The objective of the proposed RecSys is to improve the personalization and accuracy of RecSys for online shopping. Its architecture offers flexibility in blending various RecSys methods, like CF, preprocessing strategies and sentiment analysis with Glove + Bi-LSTM.

The work in [19] introduces a RecSys for rice crop disease products. It performs sentiment analysis on data, computes the rating, and reviews text sentiment polarities. Afterwards, it applies K-means clustering on products and disease-wise product data frames, splits the product clusters regions into four quartiles, and prepares the labeled dataset for a recommendation. Lastly, it evaluates and analyzes labeled datasets using an SVM Kernel classifier. The work in [20] presents a RecSys that blends sentiment analysis with CF to generate accurate and personalized recommendations. It creates a sentiment classification model that includes BERT fine-tuning, it develops a hybrid CF-based RecSys Model, and it enhances the RecSys selection process using BERT to upgrade the accuracy of e-commerce recommendations.

The rise in Agentic AI marks a shift towards autonomous agents that enhance personalization by understanding users' emotions. These agents are capable of improving both the accuracy and relevance of recommendations, being able to successfully process emotional feedback, especially in dynamic environments, such as real-time content recommendations and personalized media streaming. In this direction, the work in [21] proposes a dual-control mechanism that upgrades user agency and thus gives users the ability to manage both the degree and data collection of algorithmically tailored content they get.

Its experiments indicate that, to foster a sense of agency, transparency is not sufficient on its own, and may even make disempowerment worse than displaying the results directly. However, by combining user controls with transparency, the user agency is importantly upgraded. The work in [22] discusses the importance of Agentic AI in text classification. It not only examines applications and tools but also addresses the general framework for agentic-based text classification and indicates open challenges, like privacy, ethical issues, and knowledge transfer. Furthermore, it discusses future research to upgrade the transparency, flexibility and robustness of Agentic AI systems. The work in [23] introduces an agent-based CF agent, namely AgentCF, that simulates user-item interactions in RecSys. AgentCF considers both items and users as agents and builds a CF learning method that simultaneously enhances both types of agents. The enhanced agents are given the ability to also send their preferences to other agents, and hence implicitly develop a CF scheme. The work in [24] tests different framings of sources in recommendation explanations, like machine-oriented and human-oriented, impacting thus trusting beliefs and trusting behavioral intentions. Its outcomes indicate that, regardless of their true technical mechanism understanding, subtle wording variations may drive users to various recommendation sources.

AI agents, especially those powered by Agentic AI, are revolutionizing RecSys that autonomously adapt to user preferences and emotional cues. These agents can process both behavioral and sentimental data in order to enhance the relevance and tailoring of recommendations. Particularly in dynamic environments, such as real-time content and media streaming, these agents can offer more personalized and context-aware user experiences, improving both engagement and satisfaction. In this direction, the work in [25] introduces an AI agent-based framework that produces real-time music recommendations by upgrading adaptive learning and hybrid modeling. This work uses a multi-model ensemble to combine contextual features, content-based analysis and CF. Furthermore, it includes SHAP-based explainability to enhance trust and interpretability and FAISS-powered memory for long-term user profiling. The work in [26] develops three experiments, targeting at investigating the impact of linguistic styles on the acceptance of (AI)-generated recommendations and more specifically, the use of figurative versus literal language in ChatGPT. The outcomes of this work show that (a) figurative language can affect visit intention, with imagery vividness, (b) the users who perceived AI as human-like perceived a stronger effect of figurative language on ChatGPT recommendations, and (c) the upgrade of the figurative language used by ChatGPT is less pronounced when recommendations are made by a human agent, and (c) while the figurative language used by ChatGPT significantly boosted visit intention compared with literal language, this enhancement was less pronounced when recommendations were made by a human agent. The work in [27] introduces a framework, namely AIA-PAL, targeting improving student results through advanced Human–Agent Interaction. This framework addresses current intelligent tutoring systems' limitations by enabling personalized learning paths, as well as dynamic scaffolding. Furthermore, it also uses CrewAI and LangGraph, for real-time adaptation learning and decision-making tasks, respectively. Lastly, it encompasses retrieval-augmented generation in order to minimize hallucinations and ensure pedagogical accuracy. The work in [28] presents an AI Agent system which upgrades personalized education by utilizing automated design of agentic systems and multimodal AI methods. This AI Agent system supports diverse students' learning needs across various disciplines, like English as a second language, physics and mathematics. It also combines personalized learning pathways with adaptive assessments and dynamic content delivery and includes a mechanism that is able to successfully design and optimize agentic systems.

AI agents, especially the ones using Agentic AI, have advanced in personalizing recommendations through behavioral and emotional data. The inclusion of LLMs in Sentiment Analysis enhances the RecSys capacity to pick up on emotional signals and respond accordingly. As a result, AI systems can present users with more relevant emotional content, which can improve user engagement and enhance their overall satisfaction with an even more contextually aware experience. With this in mind, ref. [29] have developed a scalable drug recommendation personalization framework that uses both sentiment-aware analysis and advanced NLP techniques. It demonstrates advanced accuracy and contextual understanding by incorporating embeddings from LLMs into sequential and non-sequential algorithms. It also combines an adaptive Confidence-Weighted scoring mechanism with the Llama-3.2-3B-Instruct model, targeting at enhancing user expectation alignment through structured validation. The work in [30] introduces a Sentiment Extraction LLM for Personalized Recommendations, namely KLLMs4Rec, that targets at providing users with more novel, accurate, and diverse recommendations. It does so by solving the LLM's hallucination problem and the noise problem caused by the fusion of heterogeneous information in RecSys. Furthermore, this work presents a hierarchical sentiment attention graph convolutional network, which propagates user personalized preferences on the knowledge graph by integrating three sentiment weight schemes. The work in [31] explores methods to personalize LLMs, comparing zero-shot and fine-tuning reasoning approaches on subjective tasks. The outcomes of this work indicate that model reasoning is improved more than non-personalized models, with the use of personalized fine-tuning processes, underscoring the importance of LLM personalization in subjective text perception tasks, to deliver more accurate and context-aware sentiment-based recommendations.

2.2. Ensemble and Aggregation Methods in Machine Learning and NLP

Ensemble learning, which combines predictions from multiple models to achieve superior performance compared to individual models, represents a foundational paradigm in machine learning [32]. Traditional ensemble methods include bagging, boosting, and stacking, each employing different strategies for model combination [33–35].

In the context of text classification and sentiment analysis, ensemble methods have demonstrated consistent improvements over single-model approaches [36]. Majority voting, where the final prediction is determined by the most frequent class prediction among base models, represents the simplest aggregation strategy and has shown robust performance across diverse NLP tasks [37]. Weighted averaging extends this approach by assigning differential importance to models based on validation performance, allowing the ensemble to prioritize more accurate base learners [38].

Stacking, or stacked generalization, involves training a meta-learner that learns to optimally combine base model predictions [33,39]. Unlike simple voting schemes, stacking can capture complex interactions between base model outputs, though it requires additional training data for the meta-learner and may be prone to overfitting [40]. Recent work has explored neural stacking architectures for NLP tasks, where the meta-learner is implemented as a neural network trained on base model probability distributions [41].

In recommender systems specifically, ensemble methods have been applied to combine diverse recommendation algorithms [42]. Hybrid recommendation approaches often employ weighted combinations of collaborative filtering, content-based, and knowledge-based methods [43]. For sentiment-aware recommendation, ensemble techniques have been used to aggregate predictions from multiple sentiment classifiers to improve robustness [44].

However, traditional ensemble methods face limitations when applied to modern LLM-based systems. First, they typically aggregate only final predictions without accessing or reasoning over the original input, potentially missing cases where all base

models err systematically [45]. Second, they provide limited interpretability regarding why certain predictions were trusted or disregarded for specific instances [46]. Third, they require fixed combination strategies (voting weights, meta-learner parameters) determined during training, lacking the flexibility to dynamically assess model reliability on a per-instance basis.

2.3. LLM Collaboration and Multi-Model Frameworks

The emergence of large language models has introduced new paradigms for multi-model collaboration that extend beyond traditional ensemble learning. Several recent frameworks explore how multiple LLMs can be coordinated to improve performance, reliability, and robustness.

- Mixture of Experts (MoE) for LLMs: MoE architectures partition the input space and route different inputs to specialized expert models, with a gating network determining which experts to activate [47,48]. Recent work has adapted MoE principles for LLMs, where different expert models specialize in distinct domains or task types [49]. Unlike static ensembles, MoE systems dynamically select which models contribute to each prediction, though expert selection is typically based on learned routing rather than explicit reasoning.
- LLM Debate and Self-Consistency: Several studies have explored debate-based approaches where multiple LLM instances generate independent responses and then engage in iterative refinement through argumentation [50]. Self-consistency methods leverage the observation that correct reasoning paths tend to produce consistent answers, aggregating multiple reasoning traces to identify the most reliable solution [51]. These approaches demonstrate that diversity in reasoning processes can improve reliability, though they typically operate within a single model architecture rather than across heterogeneous models.
- LLM-as-Judge and Meta-Evaluation: Recent work has investigated using LLMs as judges to evaluate or aggregate outputs from other models [52,53]. This paradigm is particularly relevant for tasks where ground truth is unavailable or subjective, such as open-ended generation or creative writing. LLM judges can assess response quality, factual accuracy, or alignment with instructions, providing an alternative to human evaluation [53]. However, these approaches face challenges, including judge bias, limited calibration, and difficulty handling disagreements among high-quality but divergent responses [52].
- Model Cascades and Adaptive Routing: Cascade architectures route queries through a sequence of models of increasing capability and cost, invoking more powerful (and expensive) models only when simpler models exhibit low confidence [54]. This approach optimizes the accuracy-cost trade-off by leveraging cheaper models for easy instances while reserving expensive models for difficult cases [54]. Query routing systems extend this concept by using a classifier or heuristic to direct different input types to specialized models [55].
- Multi-Agent LLM Systems: Emerging frameworks coordinate multiple LLM agents with distinct roles, prompts, or tools to collaboratively solve complex tasks [8,56,57]. These systems typically involve iterative communication between agents, where outputs from one agent inform the inputs to others [8]. Applications include software development, scientific reasoning, and planning tasks where decomposition and specialization improve outcomes [8].

Despite these advances, existing LLM collaboration frameworks exhibit several gaps that motivate our work. First, most approaches operate within homogeneous model families (e.g., multiple instances of GPT-4) rather than leveraging heterogeneous models from

different providers with distinct architectural biases [58]. Second, many frameworks focus on generative tasks (code generation, creative writing, question answering) with limited exploration of classification tasks like sentiment analysis, where ground truth enables rigorous evaluation [59]. Third, existing meta-evaluation approaches typically assess response quality holistically rather than reasoning explicitly over prediction disagreements in the context of the original input. Our work addresses these gaps by evaluating meta-model aggregation across 12 heterogeneous models from four providers on a well-defined sentiment classification task, where the meta-model reasons over both base predictions and original review text to produce interpretable final decisions.

2.4. Research Gaps and Our Contribution

While the literature demonstrates strong foundations in sentiment analysis for RecSys (Section 2.1), ensemble methods (Section 2.2), and emerging LLM collaboration frameworks (Section 2.3), empirical evidence comparing these approaches in large-scale, multi-provider LLM evaluations remains limited. Specifically, the following gaps motivated our study:

1. Cross-Provider Heterogeneous Evaluation: Existing studies typically evaluate models from a single provider (e.g., OpenAI) or compare a small number of models [58]. Systematic evaluation across 12+ models from multiple providers (OpenAI, Anthropic, Google, DeepSeek) with distinct architectural paradigms (reasoning-optimized vs. chat-oriented) remains underexplored.
2. Ensemble Baselines for LLM Aggregation: While traditional ensemble methods are well-established, their effectiveness when applied to modern LLMs in zero-shot settings is not well-documented. Specifically, it remains unclear whether simple statistical aggregation (majority voting, mean aggregation) provides comparable benefits to more sophisticated aggregation approaches when combining diverse LLM predictions.
3. Reasoning-Based Meta-Aggregation: Existing LLM collaboration frameworks typically aggregate predictions statistically or through learned meta-models, without explicit natural language reasoning over prediction disagreements. The value of having a meta-model that can access both base model predictions and the original input to perform independent sentiment analysis remains unexplored.
4. Cost-Accuracy Trade-offs in Multi-LLM Systems: While individual papers report inference costs, systematic analysis of cost-accuracy trade-offs when aggregating multiple expensive LLMs is limited. Understanding whether accuracy improvements justify aggregation overhead is critical for production deployment.
5. Zero-Shot Comparative Benchmarking: Most sentiment analysis studies focus on fine-tuned models or few-shot prompting. Rigorous zero-shot evaluation across diverse model families on standardized datasets enables fair architectural comparison without task-specific adaptation confounds.

Conceptual Positioning: Our work provides systematic empirical evidence comparing one specific implementation of meta-model aggregation (where a reasoning-capable LLM combines predictions from multiple base models) against traditional statistical ensemble methods (majority voting, mean aggregation) and standalone model performance. Our meta-model represents a form of ensemble meta-learning, distinguished by its natural language reasoning capabilities over prediction disagreements, evaluated within the specific domain of sentiment analysis for recommender systems. We clarify that this study focuses on empirical evaluation rather than introducing a novel agent architecture or theoretical framework for ensemble learning.

To address these gaps, we conduct a systematic empirical study that: (1) evaluates 12 leading LLMs from four providers in a controlled zero-shot setting, (2) compares meta-

model aggregation against both individual model baselines and traditional ensemble methods (majority voting, mean aggregation), (3) investigates reasoning-based aggregation where a meta-model accesses original review text alongside base predictions, (4) provides comprehensive cost analysis including per-request costs and production-scale projections, and (5) conducts qualitative error analysis to understand why certain models and aggregation strategies succeed or fail on specific sentiment patterns (particularly neutral 3-star ratings). The following sections detail our methodology, results, and implications for RecSys practitioners and LLM evaluation researchers.

3. Materials and Methods

3.1. Dataset Selection and Preprocessing

3.1.1. Dataset Selection

For this work, the source dataset was the Amazon Review Dataset (2023), offered by UCSD [13], which is one of the most extensive and up-to-date product reviews compilations, containing millions of user reviews across various product categories. This dataset is considered very rich in metadata information, including (for each record) attributes like user and product identifiers, reviews and numerical ratings, textual content, verified purchase status, etc., making it highly appropriate for tasks involving multimodal RecSys and sentiment analysis.

However, due to the prohibitively high computational and monetary costs associated with processing the entire dataset through multiple LLMs (as detailed in our cost analysis in Section 4.11), we designed a rigorous sampling strategy to create a representative subset suitable for comprehensive evaluation while maintaining financial feasibility. From the Amazon Review Dataset collection, we selected five distinct product categories, each containing millions of reviews, so as to cover a wide and diverse range of product types and hence consumer behaviors. These five categories, as well as the number of entries included in each, are shown in Table 1.

Table 1. The source categories from the Amazon Reviews '23 dataset and their available review counts.

Category Name	Reviews (Available)
Fashion	2,500,939
Automotive	19,955,450
Books	29,475,453
Electronics	43,886,944
Videogames	4,624,615
	Total: 100,443,401

These source categories provide over 100 million reviews with significant diversity in both product features and user engagement patterns. The selection of these five specific categories from the 33+ available Amazon Reviews 2023 categories was based on three strategic criteria: (1) Product Type Diversity—the categories span fundamentally different product characteristics including physical goods (Fashion, Automotive), informational content (Books), technical products (Electronics), and digital entertainment (Video Games), ensuring our evaluation captures sentiment patterns across heterogeneous domains rather than a single product type; (2) Review Volume and Quality—each category contains millions of reviews (Table 1), providing sufficient data for stratified sampling while maintaining adequate minority class representation (2-star and 3-star reviews) after verified purchase filtering; and (3) E-commerce Representativeness—these categories represent major online retail segments spanning different price ranges ($5–$5000+), purchase frequencies (consumables vs. durables), and user engagement patterns (utilitarian vs. hedonic

consumption), creating a generalizable foundation for RecSys applications beyond niche product domains. This selection strategy prioritizes cross-domain generalizability over single-category depth, aligning with our research objective of evaluating LLM sentiment analysis capabilities across diverse consumer contexts. The subsequent preprocessing and sampling procedures (detailed in Section 3.1.2) reduced this initial pool to a final balanced dataset of 5000 reviews suitable for comprehensive LLM evaluation within practical cost constraints.

3.1.2. Dataset Preprocessing

The preprocessing pipeline was designed to facilitate tackling of important issues associated with large-scale user-generated content, such as noise reduction, quality assurance, class balance, etc. In order to manage computational demands and significantly reduce processing time, compared to traditional computing resources, due to the vast size of reviews, we utilized the Google Colab cloud-based Jupyter Notebook (v.7.5.3) environment, equipped with an L4 GPU and 53 GB of RAM. This infrastructure facilitated the effective handling of massive datasets that would have been prohibitively time-consuming on personal computers, even on the most powerful ones.

The preprocessing methodology involved the following five sequential steps:

1. Establishing a filtering process to guarantee the quality and relevance of the data;
2. Conducting a deduplication phase to guarantee that each product is represented only once within the dataset of each category;
3. Normalizing the textual content for natural language processing and LLM consumption;
4. Eliminating the records where either their title or their text fields were left empty;
5. Applying a stratified sampling approach to construct perfectly balanced datasets in order to overcome the issue of class imbalance.

In the first step, we established a thorough filtering process to guarantee the quality and relevance of the data. Each review had to strictly meet the following three criteria:

1. A valid numerical rating, ranging from 1 to 5,
2. At least one accompanying product image, and
3. A verified purchase status.

The requirement for verified purchases and the presence of product images were especially important, as they serve as strong signals of review authenticity and address the well-known issue of fake or incentivized reviews on e-commerce platforms. Note that the presence of product images provides visual proof (unboxing, usage) for other shoppers, enhancing review authenticity. By limiting our datasets to verified purchases only, we managed to significantly minimize the probability of including fraudulent or biased content in our data.

The filtering method was carried out using a memory-efficient chunked processing technique, where, in order to avoid memory overflow (due to the size of the source datasets), reviews were loaded and handled in batches of 100 K entries. We fully completed the processing of each category, aiming to guarantee adequate representation of minority rating classes, especially the 2-star and 3-star reviews, which are typically less common in e-commerce datasets, where ratings tend to accumulate at the extremes of the rating scale (either 1-star or 5-star ratings).

Table 2 depicts the retention rates of the aforementioned five datasets/categories, along with the respective number of reviews.

The complete category processing led to substantially different retention rates across them. These varying retention rates indicate different quality standards and user behavior patterns associated with each product category. The filtering step reduced the dataset from

100,443,401 initial reviews to 3,450,387 verified purchases (96.56% reduction), ensuring review authenticity and quality.

Table 2. The retention rates of the datasets/categories and the number of reviews concerning verified purchases.

Category Name	Retention Rate	Reviews with Verified Purchases
Fashion	5.30%	133,984
Automotive	5.63%	1,123,953
Books	1.03%	303,268
Electronics	3.91%	1,717,328
Videogames	3.72%	171,854
		Total: 3,450,387

In the second step, we conducted a deduplication phase, based on product identifiers (ASIN), to guarantee that each product is represented only once within the dataset of each category. After deduplication, each category maintained sufficient unique product reviews for balanced sampling.

In the third step, we implemented text preprocessing to standardize the textual content across the dataset. While modern LLMs are trained on raw text and can process natural formatting, we applied traditional normalization techniques for consistency across our dataset. This preprocessing included:

- Conversion to lowercase for case normalization;
- Removal of URLs and HTML tags to eliminate non-textual artifacts;
- Elimination of special characters, while preserving basic punctuation marks (periods, commas, exclamation points, question marks, apostrophes, and hyphens);
- Consolidation of multiple consecutive whitespace characters into single spaces.

We acknowledge that this normalization approach represents a methodological trade-off: while it standardizes inputs and reduces variance from inconsistent formatting artifacts across reviews, it may remove sentiment-relevant cues such as capitalization for emphasis (e.g., "AMAZING product"), repeated punctuation indicating emotion (e.g., "terrible!!!"), or acronyms carrying sentiment weight. The decision prioritized formatting consistency for traditional NLP compatibility, though we recognize that contemporary LLMs' native capability to process raw text formatting could potentially capture additional sentiment signals. This limitation is discussed further in Section 5.7.

The fourth step was designed to remove those records where either the title field or the text field was left blank because of the cleaning processes implemented in the previous steps. After this cleaning, each category retained sufficient high-quality reviews for balanced dataset creation. Following this step, we have guaranteed that all of the remaining reviews will include meaningful, valuable and informative text content.

The fifth and final step employed a two-stage stratified sampling approach to create a perfectly balanced final dataset across all five rating categories (1-star, 2-star, 3-star, 4-star, and 5-star). First, we created intermediate balanced datasets for each of the five product categories, sampling 2000 reviews per rating (10,000 total per category), resulting in 50,000 balanced reviews across all categories. Second, we applied cross-category stratified sampling by randomly selecting 200 reviews per rating from each category's balanced dataset (5 categories $\times$ 5 ratings $\times$ 200 samples = 5000 final reviews). This two-stage approach ensures both within-category balance and cross-category representativeness, with the final dataset containing exactly 1000 reviews for each of the five star ratings.

The decision to limit the final experiment to 5000 samples (0.14% of the 3.45 million verified purchases) was motivated by three primary considerations: (1) Com-

putation Cost Constraints: Our architecture involves 12 independent LLMs processing each review, followed by meta-model aggregation. Given API costs ranging from $0.24 to $43.97 per 5000 reviews per model (see Section 4.11), processing even 50,000 reviews would cost approximately $2496–$4576 for Phase I alone. Processing the full 3.45 M verified dataset would exceed $345,000, making it prohibitively expensive for an exploratory study. (2) Class Balance Requirements: The two-stage stratified sampling (10,000 per category → 5000 final) ensures both adequate minority class representation (1000 samples per rating) and cross-category diversity while maintaining computational feasibility. (3) Benchmark Alignment: Our 5000-sample size aligns with established LLM evaluation benchmarks in sentiment analysis literature, where samples of 1000–10,000 reviews are standard for comparative model assessment. While we acknowledge that a larger test set would strengthen generalizability claims, the 5000-sample design provides sufficient statistical power to detect meaningful performance differences between models (as confirmed by our results showing clear accuracy distinctions ranging from 61.25% to 71.40%).

The above 5-step preprocessing pipeline, starting from the initial dataset download and ending with the creation of the final balanced dataset, (a) ensures data quality, (b) maintains class balance, and (c) optimizes computational efficiency. The full code for this preprocessing procedure is provided as a concise Jupyter notebook file [60].

3.2. Model Deployment and Evaluation

3.2.1. Model Selection

As mentioned in the Introduction, in this study, we used 12 leading closed-source LLMs, all widely used in both commercial and industrial applications at this time. In Phase I, each served as an independent base model for zero-shot sentiment prediction. In Phase II, these models functioned within a meta-model aggregation framework where their predictions were aggregated by a meta-model. In this study, we prioritized industry-leading models that are extensively integrated into production environments due to their robustness, reliability, and operational efficiency, even though a number of open-source and closed-source alternatives (such as LLaMA and Qwen) demonstrate strong performance and rapid advancement. This approach follows current business trends, as shown by recent collaborations, such as the PayPal–OpenAI alliance [61].

The selected models represent the flagship LLMs from four major international companies at the time of this research (late 2025), chosen based on their designation by their respective providers as premier models for production deployment. These include OpenAI's GPT-5 family (reasoning-optimized models), Anthropic's Claude 4.x series (advanced language understanding), Google's Gemini 2.5 variants (multimodal capabilities), and DeepSeek's latest chat and reasoning models (competitive open-weight alternatives). The selected models are shown in Table 3, along with the names of their respective companies.

These 12 models represent a wide variety of modern LLM architectures, each with unique architectural features, capabilities and domain-specific strengths. For example, some of these models are optimized for tasks like language comprehension, summarization, and multi-turn dialogue coherence, while others are particularly strong in code generation and structured reasoning. As a result, detailed comparisons can be made between training methods and design approaches, as well as a more thorough and balanced assessment of meta-model aggregation behavior, reasoning proficiency, and adaptability across a set of LLM architectures with heterogeneous characteristics.

Table 3. The selected LLMs, along with their respective companies.

Model	Company
GPT-5-2025-08-07	OpenAI OpCo, LLC, San Francisco, CA, USA [62]
GPT-5-mini-2025-08-07	
GPT-5-nano-2025-08-07	
GPT-4.1-2025-04-14	
Claude Sonnet-4.5	Anthropic PBC, San Francisco, CA, USA [63]
Claude-Haiku-4.5	
Claude-Opus-4.1	
Gemini-2.5-pro	Google, Mountain View, CA, USA [64]
Gemini-2.5-flash	
Gemini-2.5-flash-lite	
DeepSeek-chat	Hangzhou DeepSeek Artificial Intelligence Basic Technology Research Co., Ltd., Hangzhou, China [65]
DeepSeek-reasoner	

3.2.2. Model Deployment

The 12 LLMs described above are accessed through their official APIs, where for each one, we developed a specific client to ensure consistent usage during inference, first as individual base models and afterwards as components of a meta-model aggregation system.

The developed clients standardize interactions with each model by providing (a) prompt creation, (b) dataset preparation, (c) prediction, (d) response cleaning, and (e) performance tracking methods. Specifically, we initialized each model client with a set of parameters that define the system message, input characteristics, label columns, prompt templates and response keys, ensuring that all outputs are provided in a structured JSON format suitable for downstream analysis. These clients, realized as Python classes, are made publicly available on GitHub [60] by the authors.

The implementation uses Python 3.11 with key dependencies including Pandas (v2.3.3) for dataset management, official API client libraries (OpenAI for GPT and DeepSeek models, anthropic for Claude models, Google-GenerativeAI for Gemini models), and the built-in JSON module for response parsing. The system employs a batch processing architecture that handles dataset iteration, dynamic prompt generation, API invocation, and structured response extraction within a unified pipeline.

Execution time for each prediction is measured using Python's time module, capturing the complete duration from API call initiation to processed response retrieval. Specifically, the timer starts immediately before the API request and ends after the model's response has been received and parsed into a structured format. This measurement includes: (1) network latency for request transmission and response retrieval, (2) server-side model inference time, and (3) client-side JSON parsing (typically <10 ms, representing <0.5% of total time for most requests). Importantly, execution time excludes retry attempts—when a model returns an invalid response format, the timer is reset and the request is retried, ensuring measurements reflect successful prediction time only.

The modular class-based design ensures consistent treatment of all 12 LLMs, facilitating direct comparison of reasoning capabilities, sentiment prediction performance, and meta-model aggregation behaviors across heterogeneous model architectures and API interfaces.

3.2.3. Phase I: LLMs for Sentiment Analysis

Phase I of this study aimed to assess the base prediction capabilities of the 12 selected LLMs when functioning as independent base models. During this phase, each model was tasked with performing sentiment prediction on the test dataset in a zero-shot learning setting—a term we define precisely to avoid methodological confusion. For each model, an instance of its dedicated class was initialized, ensuring that the same standardized pipeline was applied across all experiments (as described in the previous subsection).

Zero-Shot Learning Definition: In the context of LLM evaluation, "zero-shot" refers to inference without task-specific training examples, meaning the model has not been fine-tuned on labeled sentiment analysis data nor provided with in-context exemplars (labeled review-rating pairs) in the prompt. This contrasts with few-shot learning, where the prompt includes several labeled examples (e.g., "Review: 'Great product!' → Rating: 5"), and fine-tuning, where the model is trained on thousands of task-specific instances. Critically, zero-shot learning does NOT preclude structured instructions—modern LLMs require task descriptions to understand the prediction objective. An instruction-free prompt (e.g., simply presenting review text without specifying the task) would yield undefined behavior rather than meaningful sentiment predictions. Our zero-shot approach follows established LLM evaluation practices where models receive clear task specifications (predict 1–5-star ratings for product reviews) and output format constraints (JSON structure) but no labeled training examples. This methodology isolates the models' pre-trained capabilities for sentiment understanding from task-specific adaptation effects.

For each model, an instance of its dedicated class was initialized, ensuring that the same standardized pipeline was applied across all experiments (as described in the previous subsection). The inference protocol employed a standardized prompt template that combines each review's title and text into a unified input field, instructing models to predict the numerical sentiment rating (1–5 stars) with output constrained to JSON format containing only the predicted value. This uniform prompt design (illustrated in Figure 1) ensures response consistency across all 12 models, eliminates prompt-based bias, and enables direct quantitative performance comparison through automated parsing and metric computation.

```
'prompt': '''You are an expert sentiment analysis system for product reviews.
   Your task is to predict the star rating (1-5) that a user would give based on their review text.
   Rating scale:
   - 5 stars: Extremely positive, enthusiastic praise
   - 4 stars: Positive with minor reservations
   - 3 stars: Mixed feelings, balanced pros and cons
   - 2 stars: Mostly negative with few positives
   - 1 star: Extremely negative, strong dissatisfaction

   Analyze the review's sentiment, key phrases, and overall tone. Consider:
   - Emotional language (love, hate, disappointed, amazing)
   - Specific praise or complaints
   - Comparison words (better, worse, expected more)
   - Problem severity mentions
   - Recommendation likelihood

   Output only valid JSON with no additional text:
   {"rating": <number between 1-5>}

   Review:  {title} {text}''',
```

Figure 1. The prompt delivered to independent LLMs for zero-shot product reviews sentiment analysis.

Subsequently, the prompt was processed using a specialized parsing function, which extracted the LLM's predicted values from its response. The same parsing function would be able to handle situations where the LLM would generate additional or unnecessary text surrounding the JSON output. In addition to the predicted scores for each input, the total inference time was also recorded so as to enable evaluation of both prediction accuracy and efficiency. Lastly, the results of the predictions (predictions) along with their corresponding total execution times were placed in a single file/dataset, enabling simple comparison of the model's performance on different architectures and platforms.

In conclusion, this phase established a consistent baseline for evaluating each model's performance in zero-shot sentiment prediction and efficiency, before advancing to the meta-model aggregation configurations explored in the following phases (Sections 3.2.4 and 3.2.5).

3.2.4. Phase II: Meta-Model Aggregation for Sentiment Analysis

In Phase II, we evaluate a meta-model aggregation approach that combines predictions from multiple LLMs for sentiment analysis of product reviews. This approach represents a specific implementation of ensemble meta-learning where a reasoning-capable LLM (the meta-model) aggregates predictions from 12 independent base models. The system is built on a hierarchical framework with two main components: (a) a distributed network of the 12 independent base models (described in Section 3.2.1) and (b) a meta-model aggregator responsible for interpreting and aggregating the independent predictions from all 12 models.

In the first component, each model receives the same product review input (including the title and text body, as described in the previous subsection) and independently generates a sentiment rating on a 1–5 scale (illustrated at the top of Figure 2). To account for the unique interface and capabilities of each model family, the 12 models are implemented using the respective specialized method classes (GPTmethods, GEMINImethods, CLAUDEmethods, and DEEPSEEKmethods). These method classes standardize API interactions, prompt generation, and response parsing across heterogeneous architectures.

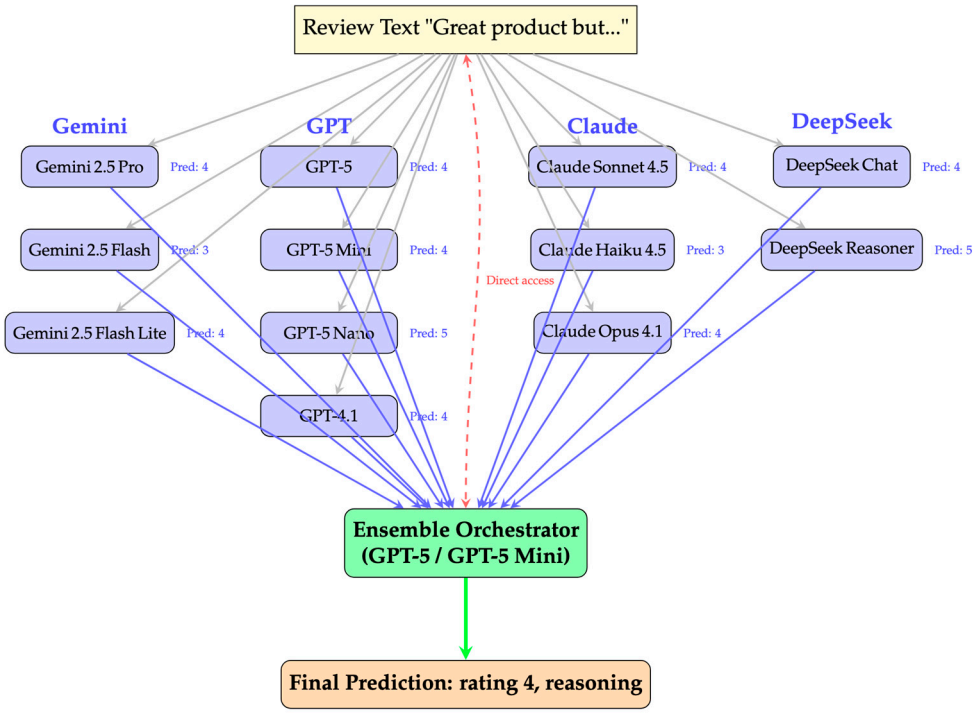

Figure 2. Ensemble aggregation architecture with meta-model reasoning for sentiment analysis.

Meta-model aggregator is a GPT-based reasoning model and represents the decision-making hub of the system (as shown at the bottom of Figure 2). The meta-model aggregator collects the rating recommendations made by each of the twelve models that comprise the network and then compares them in order to assess whether they are consistent with one another, determine if there are any possible outliers present among the recommendations and combine these into a single unified sentiment prediction, accompanied by explanatory reasoning and trust hierarchies. This design uses ensemble-style aggregation to improve the reliability and interpretability of predictions.

The meta-model's prompt template explicitly defines the 5-star rating scale and provides detailed sentiment analysis guidelines. It is capable of recognizing emotional tones, praise or criticism, comparative expressions, and the severity of reported issues (as shown in Figure 3). The final prompt enables the meta-model to reason holistically about both the input content and the diversity of model opinions by combining the original review text (red line in Figure 2) with the aforementioned criteria and the aggregated model outputs (blue lines in Figure 2).

```
self.orchestrator_prompt_template = '''You are an orchestrator on an agentic ai sentiment analysis system
for product reviews. Your task is to predict the star rating (1-5) that a user would give based on their review text.

Rating scale:
- 5 stars: Extremely positive, enthusiastic praise
- 4 stars: Positive with minor reservations
- 3 stars: Mixed feelings, balanced pros and cons
- 2 stars: Mostly negative with few positives
- 1 star: Extremely negative, strong dissatisfaction

Analyze the review's sentiment, key phrases, and overall tone. Consider:
- Emotional language (love, hate, disappointed, amazing)
- Specific praise or complaints
- Comparison words (better, worse, expected more)
- Problem severity mentions
- Recommendation likelihood

Your AI agents have already made their own predictions:

Your task is to assess their predictions based on the Review given and conclude to the appropriate
prediction of the rating for this review.

Below I am providing you AI Agents' predictions:
{predictions}

Output only valid JSON with no additional text:
{{"rating": <number between 1-5>, "reasoning": <reason what you detect on ai agents' predictions,
which agent deviates from the others and how you conclude to your rating prediction>}}

Review: {title} {text}
'''
```

Figure 3. The meta-model prompt template used in the meta-model aggregation sentiment analysis system.

In this study, we used two GPT-based, reasoning-optimized models as meta-models, namely GPT-5-2025-08-07 and GPT-5-mini-2025-08-07, both recognized for their strong reasoning performance [62]. The first model inherits its reasoning framework from the o3 model, while the second is derived from the o4-mini family. We also evaluated an alternative meta-model, the DeepSeek-reasoner; however, it was found to require significantly longer inference times, making it impractical for large-scale experimentation (comparative timing results are discussed in the Results section).

The selection of GPT-based models as meta-models was motivated by their explicit classification as "reasoning models" with dedicated reasoning token support, as documented in OpenAI's official model specifications [62]. While Claude Sonnet 4.5 achieved the highest standalone accuracy (65.02%, Section 4.1) and demonstrated strong perfor-

mance as a base model, it is not officially designated as a reasoning-optimized model by Anthropic. Given that aggregation tasks require explicit multi-step reasoning over diverse predictions—including conflict resolution, outlier detection, and trust assessment—we prioritized models with documented reasoning capabilities for this role. However, evaluating Claude Sonnet 4.5 and other high-performing models (such as Claude Opus 4.1) as meta-models represents an important direction for future research, particularly to assess whether standalone accuracy translates to superior aggregation performance or whether specialized reasoning architectures provide distinct advantages.

The overall processing pipeline begins by loading the product reviews from a structured dataset. Then, the meta-model sends the reviews to the 12 models and collects their results (i.e., their predicted ratings). The results are subsequently gathered, validated, aggregated, and analyzed by the meta-model to produce the final sentiment prediction and reasoning output. Finally, these results are stored along with comprehensive metadata, including individual model predictions, final aggregated outcomes, meta-model reasoning, and detailed processing-time metrics.

In order for the system to be resilient, it incorporates reliability mechanisms, including automated error handling for API failures, retry logic for failed requests, and validation checks for malformed responses. Moreover, the modular design of the system's architecture allows for easy integration of additional models and/or meta-models in future expansions.

Overall, this meta-model aggregation system represents an approach to ensemble reasoning with LLMs, demonstrating how multiple LLMs can be aggregated to achieve more efficient, reliable and interpretable sentiment analysis, compared to single-model baselines.

3.2.5. Evaluation Framework and Metrics

The evaluation framework combines quantitative metrics, qualitative analysis and assessment of computational and monetary costs, in order to provide a comprehensive understanding of the model capabilities. Therefore, it is capable of evaluating both the performance of the individual models and the effectiveness and added value of the aggregated LLM system at the same time.

Individual LLM Evaluation. The performance of each model was evaluated using four main metrics, computed with the sklearn package: accuracy, precision, recall, and F1-Score. The accuracy metric is used as the main performance indicator. It measures the overall prediction correctness across the full range of ratings (from 1 to 5 stars). The precision and recall metrics evaluate the prediction correctness and completeness, respectively, for each rating category. Lastly, the F1-Score metric provides a balanced measure of precision and recall. F1-score is computed using macro averaging (i.e., computed separately for each class and then averaging the individual metrics) [66] to account for the multi-class nature of the sentiment classification task.

The framework is also capable of (a) examining zero-shot sentiment classification capabilities, (b) identifying the models that achieve optimal performance in detecting specific sentiment intensities, and (c) measuring the cost for each model to assess computational efficiency, all at a single rating level. Furthermore, we employ visualization tools, such as confusion matrices, heatmaps, and performance charts, to reveal prediction patterns and distributions, gaining insight into the strengths and weaknesses of each individual model.

Meta-Model Aggregation System Evaluation. The evaluation of this setting, i.e., the aggregated system, mainly focuses on the meta-model's ability to synthesize and refine the predictions of the individual models included. The framework analyzes how the meta-model integrates, weighs, and revises model recommendations in order to produce final sentiment ratings. The framework also compares the meta-model's performance

against individual model averages and majority voting baselines, to evaluate any accuracy improvements or degradations resulting from the aggregation procedure. During the evaluation, patterns of agreement and disagreement between models and the meta-model are also examined to highlight cases of consensus, divergent predictions, and the impact of outlier models. Furthermore, the framework measures the overhead incurred by the meta-model and compares it to the gains in predictive performance to analyze the computational and monetary cost. Lastly, the framework provides insight into the efficiency and overall value of the meta-model aggregation approach by using metrics such as cost per accuracy point improvement.

3.2.6. Comparison with Traditional Ensemble Methods

Our meta-model represents a specific instantiation of ensemble meta-learning, distinguished by natural language reasoning over predictions. To clarify the relationship between this meta-model-based approach and established ensemble learning techniques, we explicitly compare our meta-model aggregation system with three widely used ensemble methods: majority voting, mean aggregation, and stacking.

Majority Voting is the simplest ensemble approach, where the final prediction is determined by the most frequent prediction among base models. In our context, this would select the rating (1–5) that receives the most votes from the 12 independent LLMs. While computationally efficient and easy to interpret, majority voting treats all models equally and cannot leverage differential model expertise or account for prediction confidence.

Mean Aggregation computes the arithmetic mean of all base model predictions and rounds to the nearest valid rating. This unweighted averaging approach treats all models equally, similar to majority voting, but utilizes the full numerical scale rather than discrete votes. While more sophisticated variants exist—such as weighted averaging, which assigns different importance to each model based on validation performance—we employ unweighted mean aggregation as a straightforward baseline that requires no additional training data or hyperparameter tuning. Like majority voting, this remains a purely statistical aggregation technique without contextual awareness of the specific input being classified.

Stacking (or stacked generalization) involves training a meta-learner that takes base model predictions as input features and learns to combine them optimally. The meta-learner is typically trained on a held-out validation set. While more sophisticated than voting schemes, traditional stacking methods produce predictions without explicit reasoning or interpretability regarding why certain base models were trusted or disregarded for specific instances.

Our meta-model-based approach differs from these traditional methods in several key aspects:

- Access to Original Input: Unlike standard stacking, where the meta-learner typically sees only base model predictions, our meta-model receives both the aggregated predictions and the original review text. This enables it to perform independent sentiment analysis and detect cases where base models may have collectively erred.
- Explicit Reasoning: The meta-model generates natural language explanations for its decisions, including which models deviated from the consensus and why certain predictions were trusted or disregarded. This interpretability is absent in statistical ensemble methods.
- Dynamic Trust Assessment: Rather than relying on fixed weights learned during training, the meta-model dynamically evaluates model reliability on a per-instance basis, identifying outliers and adjusting its trust accordingly based on the specific linguistic characteristics of each review.

- Override Capability: As demonstrated in Section 4.6, the meta-model can disregard all base model recommendations when it determines that its own analysis of the review text yields a more accurate assessment. This capability goes beyond traditional meta-learning, which is constrained to combining existing predictions.
- Zero-Shot Meta-Reasoning: The meta-model operates without task-specific training on sentiment analysis examples. Traditional stacking requires training the meta-learner on labeled validation data, whereas our approach leverages the pre-trained reasoning capabilities of the meta-model.

To empirically validate these distinctions, we computed the performance of baseline ensemble methods on our dataset. Table 4 presents a comparison between our meta-model and traditional aggregation techniques.

Table 4. Comparison of Ensemble Methods and meta-model Performance.

Method	Accuracy	Precision	Recall	F1-Score
Majority Voting	62.64%	0.6332	0.6264	0.6206
Mean Aggregation (Rounded)	62.96%	0.6374	0.6296	0.6245
GPT-5 Meta-Model	71.40%	0.7229	0.714	0.7092
GPT-5 Mini Meta-Model	70.32%	0.7119	0.7032	0.6982
Average Individual Model	61.25%	-	-	-

As shown, simple majority voting achieves 62.64% accuracy and mean aggregation achieves 62.96% accuracy, both closely matching the average performance of individual models (61.25%). This minimal improvement (+1.39 to +1.71 percentage points) indicates that statistical aggregation alone provides limited benefit when base models exhibit similar error patterns.

In contrast, our meta-model-based approach achieves 71.40% accuracy, representing:

- An 8.76 percentage point improvement over majority voting (13.98% relative improvement)
- An 8.44 percentage point improvement over mean aggregation (13.41% relative improvement)
- A 10.15 percentage point improvement over the average individual model (16.57% relative improvement)

These gains demonstrate that the meta-model's ability to reason over both base predictions and original text provides additional value beyond statistical aggregation. The meta-model not only aggregates predictions but actively analyzes the review content to identify cases where the collective judgment of base models may be incorrect.

We acknowledge that our approach shares structural similarities with ensemble learning—both combine multiple models to improve predictive performance. However, the explicit reasoning, dynamic trust assessment, and ability to override collective predictions distinguish our meta-model from traditional meta-learning techniques. We position our contribution not as a replacement for ensemble methods, but as a complementary approach that trades computational cost for interpretability and context-aware reasoning in high-stakes prediction scenarios.

4. Results

This section presents empirical findings from our comparative evaluation. The findings address the 15 research questions posed in the Introduction section. For each RQ, we restate the question (for the reader's convenience), present the relevant findings, and then provide an answer to the RQ. In all result tables, boldface is used to mark the highest performance.

4.1. Individual Model Performance Analysis

RQ1: Which LLM achieves the highest accuracy in zero-shot sentiment classification?

The evaluation of the 12 models shows clear differences in their zero-shot sentiment classification performance (illustrated in Table 5). At the company level, in this task, the Claude models achieved the best performance overall, while the Google models achieved the lowest performance. At the model level, Claude Sonnet 4.5 achieved the highest accuracy at 65.02%, followed closely by Claude Opus 4.1 at 64.48% and GPT-4.1 at 63.54%.

Table 5. Model Performance Comparison.

Company	Model	Accuracy	Precision	Recall	F1-Score
Google	Gemini-2.5-pro	0.5906	0.6001	0.5906	0.5792
	Gemini-2.5-flash	0.5686	0.5735	0.5686	0.5427
	Gemini-2.5-flash-lite	0.5844	0.5995	0.5844	0.5802
OpenAI OpCo, LLC	GPT-5	0.6240	0.6330	0.6240	0.6126
	GPT-5-mini	0.6234	0.6271	0.6234	0.6107
	GPT-5-nano	0.5984	0.6131	0.5984	0.5963
	GPT-4.1	0.6354	0.6400	0.6354	0.6326
Anthropic PBC	Claude Sonnet-4.5	**0.6502**	0.6548	0.6502	**0.6475**
	Claude-Haiku-4.5	0.5950	0.6163	0.5950	0.5914
	Claude-Opus-4.1	0.6448	**0.6607**	**0.6448**	0.6463
Hangzhou DeepSeek Artificial Intelligence Basic Technology Research Co., Ltd.	DeepSeek-chat	0.6334	0.6341	0.6334	0.6270
	DeepSeek-reasoner	0.6022	0.6053	0.6022	0.5945
OpenAI OpCo, LLC	GPT-5 as a meta-model	0.7140	0.7229	0.7140	0.7092
	GPT-5-mini as a meta-model	0.7032	0.7119	0.7032	0.6982

Moderate performance was achieved by the DeepSeek Chat (63.34%), GPT-5 (62.40%), and GPT-5 Mini (62.34%) models, while the worst results were observed for the Gemini 2.5 Flash model (56.86%).

An interesting finding is that, even though DeepSeek Reasoner required a lot more computational resources than DeepSeek Chat (c.f. Section 4.12), due to its reasoning-based architecture, it failed to surpass its accuracy.

A similar situation was observed between GPT-5 and GPT-4.1, where GPT-5 (designed as a reasoning model) achieved 1.14 percentage points lower accuracy than the older GPT-4.1 model. This suggests that, for zero-shot sentiment classification, reasoning-oriented models may not offer an advantage over their chat-oriented counterparts, despite their higher associated costs.

On the other hand, when these models are incorporated into a meta-model aggregation system, we observe a significant improvement in performance (as shown in the last two rows of Table 5). More specifically, the GPT-5 meta-model achieved an accuracy of 71.40%, while the GPT-5 mini meta-model reached 70.32%, representing improvements of 10.15% and 9.07%, respectively, compared to the average accuracy of the individual models, which is measured at 61.25%.

4.2. Rating Prediction Challenges

RQ2: At which rating levels (1–5) do LLMs most frequently make incorrect predictions, and at which levels do they perform most accurately?

Most models showed a similar pattern of difficulty with 3-star ratings, with failure rates typically ranging from 55% to 71% (as shown in Table 6). Interestingly, DeepSeek Reasoner was the only model to deviate from this pattern, showing its highest failure rate with 2-star ratings (62.50%). This unique behavior indicates that the DeepSeek Reasoner may be processing negative sentiment differently to the other models in our study.

Table 6. Hardest and easiest rating levels (1–5) for independent base models in sentiment prediction tasks.

Model	Hardest Rating (Failure %)	Easiest Rating (Success %)
Gemini-2.5-pro	3 (65.30%)	**1 (92.00%)**
Gemini-2.5-flash	**3 (77.40%)**	**1 (92.00%)**
Gemini-2.5-flash-lite	3 (64.50%)	1 (86.50%)
GPT-5	3 (71.50%)	1 (85.00%)
GPT-5-mini	3 (69.90%)	1 (83.70%)
GPT-5-nano	3 (68.40%)	5 (73.40%)
GPT-4.1	3 (58.30%)	5 (81.70%)
Claude Sonnet-4.5	3 (55.70%)	5 (84.30%)
Claude-Haiku-4.5	3 (70.00%)	1 (79.30%)
Claude-Opus-4.1	3 (57.50%)	5 (76.10%)
DeepSeek-chat	3 (61.20%)	5 (83.80%)
DeepSeek-reasoner	2 (62.50%)	1 (87.50%)
Overall Average	3 (64.83% failures)	1 (82.73% success)

Gemini 2.5 Flash produced the highest number of failures (measured at 77.4%) for 3-star ratings (indicating a "fair" user opinion), whereas Claude Sonnet-4.5 produced the fewest (measured at 55.7%). As a result, the difference between these two extremes was a significant 21.7% and shows that there is a clear distinction between the ability of each model to understand "neutral" or "mediocre" sentiment expressions.

Therefore, the above measurements highlight a serious problem that all researchers face when studying sentiment analysis: accurately classifying neutral or mediocre sentiment, which can be determined using data collected from 3-star ratings on a 5-star rating system. To understand why models systematically fail on 3-star reviews, we conducted a qualitative analysis examining misclassified instances across three dimensions: linguistic ambiguity, rating scale interpretation, and contextual insufficiency.

- Linguistic Ambiguity: 3-star reviews frequently contain hedging language ("decent but…", "okay for the price", "works fine, I guess") that lacks the definitive sentiment markers present in extreme ratings. Many misclassified 3-star reviews contain contradictory statements (e.g., "Great quality but arrived late"—should be 3-star but predicted as 4-star due to positive "great quality" framing), where the coexistence of positive and negative phrases within single sentences creates feature-level ambiguity that models struggle to balance appropriately. Additionally, comparative qualifiers ("better than expected" without establishing baseline expectations) and sarcastic or ironic expressions that LLMs may interpret literally ("Oh, perfect, another broken charger") further complicate accurate classification.
- Rating Scale Interpretation: The semantic space between 2-star ("poor"), 3-star ("acceptable"), and 4-star ("good") contains finer-grained distinctions than extreme ratings. When models misclassify 3-star reviews, they frequently predict extreme ratings (1-star or 5-star) rather than adjacent ratings, suggesting difficulty in calibrating "middling" sentiment (Table 7). The negative bias observed in Table 8 (mean bias of -0.39 to -0.81 for 3-star) suggests models apply a "negativity heuristic"—treating any criticism as evidence for lower ratings, even when reviews explicitly state "met basic expectations" or "acceptable quality." This aligns with the anchoring effect, where models pre-trained predominantly on polarized sentiment (more common in training corpora) struggle with the nuanced midpoint calibration required for 3-star classification.

Table 7. Mean Absolute Error (MAE) per Rating Level.

Model	Overall	Rating 1	Rating 2	Rating 3	Rating 4	Rating 5
Gemini-2.5-pro	0.491	0.096	0.634	0.868	0.606	0.253
Gemini-2.5-flash	0.536	0.105	0.703	1.037	0.681	0.152
Gemini-2.5-flash-lite	0.485	0.154	0.547	0.796	0.547	0.379
GPT-5	0.435	0.166	0.456	0.836	0.51	0.209
GPT-5-mini	0.441	0.197	0.56	0.832	0.393	0.225
GPT-5-nano	0.46	0.306	0.432	0.78	0.481	0.299
GPT-4.1	0.409	0.232	0.441	0.667	0.492	0.212
Claude Sonnet-4.5	0.392	0.234	0.404	0.639	0.497	0.186
Claude-Haiku-4.5	0.46	0.225	0.388	0.793	0.502	0.393
Claude-Opus-4.1	0.392	0.294	0.327	0.632	0.443	0.265
DeepSeek-chat	0.423	0.203	0.473	0.726	0.519	0.196
DeepSeek-reasoner	0.467	0.166	0.636	0.746	0.495	0.291

Table 8. Mean Error (Bias) per Rating Level.

Model	Overall	Rating 1	Rating 2	Rating 3	Rating 4	Rating 5
Gemini-2.5-pro	−0.355	0.096	−0.5	−0.756	−0.362	−0.253
Gemini-2.5-flash	−0.326	0.105	−0.531	−0.809	−0.243	−0.152
Gemini-2.5-flash-lite	−0.327	0.154	−0.365	−0.646	−0.397	−0.379
GPT-5	−0.232	0.166	−0.308	−0.602	−0.208	−0.209
GPT-5-mini	−0.16	0.197	−0.248	−0.39	−0.133	−0.225
GPT-5-nano	−0.183	0.306	−0.164	−0.5	−0.259	−0.299
GPT-4.1	−0.157	0.232	−0.181	−0.449	−0.174	−0.212
Claude Sonnet-4.5	−0.149	0.234	−0.166	−0.439	−0.189	−0.186
Claude-Haiku-4.5	−0.272	0.225	−0.234	−0.609	−0.35	−0.393
Claude-Opus-4.1	−0.148	0.294	−0.105	−0.43	−0.235	−0.265
DeepSeek-chat	−0.179	0.203	−0.221	−0.488	−0.195	−0.196
DeepSeek-reasoner	−0.252	0.166	−0.346	−0.52	−0.269	−0.291

- Contextual Insufficiency: Short-form reviews often lack sufficient context for models to disambiguate neutral sentiment. Reviews such as “Product is fine” (actual 3-star, predicted as 4-star) or “Didn’t work for me” (actual 3-star, predicted as 1-star) provide minimal information about whether dissatisfaction stems from personal preferences (warranting 3-star) or objective product failures (warranting 1-star).

As such, this finding has critical implications for how sentiment analysis systems are designed and indicates that developing the ability to detect and classify neutral sentiment expressions must receive greater attention. Potential mitigation strategies include: (1) incorporating lexical diversity and sentiment contrast features to flag ambiguous cases for human review, (2) fine-tuning models specifically on neutral sentiment with contrastive learning objectives that emphasize 2-vs.-3-vs.-4 distinctions, and (3) implementing confidence thresholds that escalate uncertain 3-star predictions to secondary verification.

On the other hand, all independent base models achieved their best performance when predicting values at the extremes of the rating scale (either 1-star or 5-star classifications), representing clear positive or negative sentiments. This concurs with the results of previous studies [15,67] which assert that ratings at the bounds of the scale are easier to predict.

Interestingly, we observe that while the models that achieved their best performance on 1-star predictions succeeded more frequently on the ‘easiest rating’ cases, on average, the models that excelled on 5-star predictions performed better (i.e., had the lowest failure percentage) on the “hardest rating” cases. Based on the combined results, the models that achieved higher accuracy on 5-star ratings, such as the Claude Sonnet 4.5 and the GPT-

4.1, tended to exhibit stronger overall performance and thereby confirming the previous observation. Furthermore, we also observe significant differences between models from the same company. For example, while GPT-5 and GPT-5 Mini achieved their best performance on 5-star predictions, the GPT-4.1 model performed better on 1-star ratings. A similar situation occurred within the DeepSeek family, where the reasoning variant (the DeepSeek-reasoner model) proved to be more accurate on 1-star predictions than on 5-star ones, as opposed to the DeepSeek-chat model. This pattern strongly suggests that the reasoning-oriented models exhibit better performance in identifying negative sentiment cues, while the chat-oriented models balance sentiment extremes more effectively.

To further investigate the magnitude and direction of prediction errors, we calculated the Mean Absolute Error (MAE) and Mean Error (Bias) for each model across different rating levels. Table 7 presents the MAE for each model, highlighting the error magnitude on the easiest (1-star and 5-star) and hardest (3-star) ratings.

The results in Table 7 confirm that 3-star ratings are indeed the most challenging, with MAE values ranging from 0.632 to 1.037, significantly higher than the errors observed at the extremes. Conversely, the 1-star and 5-star ratings exhibit much lower MAE values, typically below 0.3, reinforcing the finding that models are more adept at identifying strong sentiments.

We also analyzed the systematic bias of the models using the Mean Error metric, as shown in Table 8.

Table 8 reveals a consistent negative bias across almost all models, indicating a general tendency to underestimate ratings. This is particularly pronounced for 3-star ratings, where models frequently predict lower values (1 or 2). The positive bias observed for 1-star ratings is expected, as predictions can only be equal to or higher than the true value. Similarly, the negative bias for 5-star ratings is due to the upper bound of the scale. However, the strong negative bias in the intermediate ratings suggests that models may be overly sensitive to negative sentiment cues, leading them to classify neutral or mixed reviews as more negative than they actually are.

4.3. Impact of Aggregation on System Performance

RQ3: How does meta-model aggregation compare to traditional ensemble baselines in terms of observed accuracy improvements?

When the LLMs are incorporated into a meta-model aggregation system, we observe significant improvements in sentiment analysis accuracy in both meta-model variants, albeit with notable differences in their performance. More specifically, when using the GPT-5 meta-model, we observe an accuracy of 71.4%, representing a 10.15% increase over the average model performance (61.25%). When using the GPT-5 mini meta-model, we observe an accuracy of 70.32%, which corresponds to a 9.07% improvement over the average model performance.

The difference in accuracy between these two meta-models (measured at 1.08%) suggests that the more advanced GPT-5 model is slightly better at synthesizing and analyzing model recommendations. However, the GPT-5 Mini model delivers results that are comparable to those of the GPT-5 model. Therefore, we can conclude that the aggregation architecture itself, rather than the choice of model, is the key factor driving performance enhancement. This suggests that even when using lightweight models, we can achieve very good results, as long as the aggregation architecture remains intact.

Statistical Significance Validation: To rigorously validate that these improvements are not due to random variance in our 5000-sample test set, we performed comprehensive statistical significance testing using McNemar's test for paired classifier comparisons. McNemar's test is specifically designed for evaluating whether two classifiers have sig-

nificantly different error rates on the same test set, making it ideal for our paired prediction scenario. All comparisons demonstrate highly statistically significant results: when comparing the GPT-5 meta-model against individual models, all 12 comparisons yielded $p < 0.001$ (chi-square test statistics ranging from $\chi^2 = 110.04$ to $\chi^2 = 482.23$), indicating that the meta-model's superior performance is not attributable to sampling variance. Similarly, the GPT-5 mini meta-model achieved $p < 0.001$ for all individual model comparisons ($\chi^2 = 79.47$ to $\chi^2 = 381.08$). When compared against the majority voting baseline (62.64% accuracy), both meta-models demonstrated highly significant improvements: GPT-5 meta-model ($\chi^2 = 279.19$, $p < 0.001$) and GPT-5 mini meta-model ($\chi^2 = 210.16$, $p < 0.001$). Bootstrap confidence intervals (10,000 iterations) further confirm the reliability of these estimates: GPT-5 meta-model accuracy is 71.40% with 95% CI [70.16%, 72.66%], and GPT-5 mini meta-model is 70.32% with 95% CI [69.08%, 71.54%]. The narrow confidence intervals (margin of error $\pm$1.23–1.25 percentage points) indicate that our 5000-sample test set provides sufficient statistical power to detect meaningful performance differences. These statistical tests conclusively establish that the meta-model improvements are both statistically significant and practically meaningful, addressing concerns about potential sampling variance in our experimental design.

4.4. LLM Influence Analysis

RQ4: Which LLMs contribute positively or negatively to the meta-model's performance?

To rigorously assess the influence of each independent model on the meta-model's decision-making process, we employ a classification scheme that evaluates recommendations relative to the meta-model's standalone performance. This approach focuses on the potential influence of each model—counting all recommendations that provide better or worse estimates—rather than only those that successfully altered the final aggregated decision.

Specifically, we categorize each model's recommendation as follows:

- **Positive Influence**: The model provides a more accurate estimate than the meta-model (i.e., $|\text{Pred(A)} - \text{True}| < |\text{Pred(M)} - \text{True}|$). This includes cases where the model pulls the estimate closer to the ground truth, even if it overshoots or does not perfectly match the true value.
- **Negative Influence**: The model provides a less accurate estimate than the meta-model (i.e., $|\text{Pred(A)} - \text{True}| > |\text{Pred(M)} - \text{True}|$). This captures cases where the model would mislead the meta-model further from reality.
- **Neutral Influence**: The model provides an estimate with the same error magnitude as the meta-model (i.e., $|\text{Pred(A)} - \text{True}| = |\text{Pred(M)} - \text{True}|$). This typically occurs when the model and meta-model output the same rating, thereby reinforcing the meta-model's initial belief (whether correct or incorrect).

This definition allows us to measure the intrinsic value of each model's input, independent of the aggregation mechanism's final output.

Based on this analysis, significant differences can be observed in the potential influence of individual models, as shown in Table 9. When GPT-5 serves as the meta-model, the models with the highest number of positive influences are Claude Opus 4.1 and Claude Sonnet 4.5, with 587 and 516, respectively. These models demonstrate a strong capacity to correct the meta-model's errors. Interestingly, the GPT-4.1 model shows the lowest number of negative influences (288), suggesting it is the least likely to mislead the meta-model, followed closely by Claude Sonnet 4.5 (300).

When using the GPT-5 Mini as a meta-model, we observe a similar pattern, where the Claude Opus 4.1 achieves the highest number of positive influences (659), followed by

Claude Sonnet 4.5 (594). In terms of negative influence, Claude Sonnet 4.5 and GPT-4.1 remain the safest options, with 355 and 361 negative influences, respectively.

Table 9. Comparison of the base models' positive and negative influence on the meta-model performance for the GPT-5 and GPT-5 mini meta-models.

Model	GPT-5 Meta-Model		GPT-5 Mini Meta-Model	
	Positive	**Negative**	**Positive**	**Negative**
Gemini-2.5-pro	300	577	438	677
Gemini-2.5-flash	299	785	408	849
Gemini-2.5-flash-lite	375	616	448	656
GPT-5	0	0	428	396
GPT-5-mini	396	428	0	0
GPT-5-nano	446	562	478	564
GPT-4.1	417	288	518	361
Claude Sonnet-4.5	516	300	594	355
Claude-Haiku-4.5	410	529	519	610
Claude-Opus-4.1	587	374	659	419
DeepSeek-chat	445	372	508	415
DeepSeek-reasoner	386	542	417	547

The high positive influence counts for the Claude models across both meta-models highlight their complementary strength to the GPT-based meta-models. By frequently providing more accurate predictions when the GPT models make mistakes, they act as valuable correctors within the meta-model aggregation system. In contrast, models like Gemini 2.5 Flash show high negative influence counts (785 and 849), indicating a greater risk of introducing errors if their recommendations are followed blindly.

To systematically validate the relationship between standalone model performance and meta-model influence, we conducted a correlation analysis synthesizing the performance results from Table 5 with the influence metrics presented above. The analysis reveals exceptionally strong correlations between model accuracy and influence patterns: For the GPT-5 meta-model, standalone accuracy exhibits a very strong positive correlation with net influence (Pearson r = 0.974, $p < 0.0001$; Spearman $\rho = 0.964$, $p < 0.0001$) and an equally strong negative correlation with negative influence (Pearson r = −0.961, $p < 0.0001$). The GPT-5 mini meta-model shows nearly identical patterns (net influence: Pearson r = 0.978, $p < 0.0001$; negative influence: Pearson r = −0.952, $p < 0.0001$). These statistically significant correlations show that higher-performing standalone models consistently provide more beneficial guidance to the meta-model, while lower-performing models systematically introduce more errors.

Notably, the three highest-accuracy standalone models—Claude Sonnet 4.5 (65.02%), Claude Opus 4.1 (64.48%), and GPT-4.1 (63.54%)—also achieve the strongest net positive influences (+216, +213, and +129, respectively, for the GPT-5 meta-model) and highest positive influence ratios (63.24%, 61.08%, and 59.15%). Conversely, the three lowest-accuracy models—Gemini 2.5 Flash (56.86%), Gemini 2.5 Flash-Lite (58.44%), and Gemini 2.5 Pro (59.06%)—exhibit the most negative net influences (−486, −241, and −277), with positive influence ratios below 40%. This direct correspondence between standalone performance and meta-model contribution shows that the meta-model effectively recognizes and leverages high-quality predictions while appropriately discounting unreliable ones.

Two notable anomalies merit discussion: GPT-5 Mini (62.34% accuracy) shows a slightly negative net influence (−32) despite above-average performance, and DeepSeek Reasoner (60.22% accuracy) exhibits substantial negative influence (−156) despite moderate standalone performance. These anomalies suggest that prediction quality alone does not

guarantee positive meta-model influence—factors such as prediction confidence calibration, output format consistency, and alignment with the meta-model's reasoning style may also play critical roles.

Based on the above findings, we conclude that the effectiveness of a meta-model aggregation system relies heavily on the selection of models that not only perform well individually but also possess the specific capability to correct the meta-model's weaknesses. The strong empirical correlation between standalone accuracy and meta-model influence provides quantitative validation that high-performing base models function as effective "error correctors" within the meta-model aggregation framework.

4.5. Meta-Model Decision-Making: Revisions and Independence

RQ 5: How does the meta-model handle independent models' recommendations—specifically, how frequently does it accept (revise) versus disregard them, and are these decisions beneficial?

Revision Behavior (Accepting Recommendations). The GPT-5 meta-model made changes to its original predictions, as directed by independent models, 760 times (15.20%) during the course of this study. Of the 760 changes made to its predictions, 583 were correct (a success rate of more than 76%) and 133 were incorrect. These two statistics result in a positive revision ratio of approximately 4.4 adjustments for each incorrect adjustment. The GPT-5 mini meta-model, on the other hand, made changes to its original predictions 788 times (15.76%), showing a more active revision behavior. Of the 788 changes made to its predictions, 551 were correct (a success rate of 70%) and 152 were incorrect. As such, the GPT-5 mini meta-model has a slightly lower, but still positive, revision ratio of approximately 3.6 adjustments per incorrect revision.

Disregarding Behavior (Independent Decision-Making). Beyond accepting recommendations, we also analyzed how often meta-models completely disregard them. We define "disregarding" recommendations as instances where the meta-model's final prediction does not match any of the ratings provided by the 12 base models. This represents a strong form of independence, where the meta-model rejects the entire set of proposed values—including the recommendation from its own underlying model acting as a model—and generates a unique conclusion based on its own reasoning over the source text (presumably taking into account to some extent the estimations of individual models).

More specifically, the GPT-5 meta-model disregarded its base models' recommendations in 231 cases (4.62% of total predictions), while the GPT-5 mini meta-model did so in 175 cases (3.50% of total predictions). This indicates that the GPT-5 meta-model proved to be more confident in its independent decision-making, while the GPT-5 mini meta-model proved to be more conservative.

Considering the GPT-5 meta-model, all 231 disregard cases were correct (100% accuracy), while the GPT-5 Mini achieved 96.57% accuracy (169 correct and 6 incorrect cases). This behavior suggests that both meta-models can successfully operate independently of base models' recommendations, with the GPT-5 meta-model showing superior judgment in knowing when to trust its independent analysis.

The GPT-5 mini meta-model demonstrated more active revision behavior (15.76% vs. 15.20%), indicating it is more likely to follow the recommendations of its independent models. However, the GPT-5 meta-model achieved both a higher improvement-to-error ratio (4.4 vs. 3.6) and perfect accuracy (100%) when disregarding all recommendations, demonstrating that the quality of meta-model decisions may be more important than the quantity of revisions in aggregation tasks.

4.6. Model Trust and Influence Patterns

RQ6: Which models most strongly influence the meta-model's decisions, and which are least trusted?

Based on our experiments, the GPT-5 meta-model shows the strongest alignment with the predictions from its own base model (GPT-5), while showing the least reliance on the Gemini 2.5 Flash model. This behavior suggests an intrinsic connection with its own model family (the GPT family from OpenAI OpCo, LLC), while being more skeptical of models that may differ in their reasoning and result formulation processes.

Similarly, the GPT-5 mini meta-model also shows the strongest alignment with the predictions from its own model (GPT-5 Mini), while demonstrating the least reliance on the Gemini 2.5 Flash model. However, the key difference is that the GPT-5 mini meta-model takes the predictions from GPT-5 more seriously, indicating that, while it retains its decision-making autonomy, it values the results produced by its more advanced counterpart to a greater extent. It is worth noting that the meta-model prompt includes both the name and the rating of each model, ensuring that the meta-model is aware of which model generated each prediction.

The fact that both meta-models identify the Gemini 2.5 Flash model as the least trusted one indicates that, regardless of the meta-model's sophistication, certain patterns of model reliability are universally recognized. This inter-agreement evaluation (derived from both the GPT-5 meta-model and the GPT-5 mini meta-model) practically shows the ability of the meta-models to identify and appropriately adjust the importance of less reliable predictions, regardless of their trust and influence approach.

4.7. Comparative Analysis of Standalone Versus Aggregated Performance

RQ7: How does a model's performance as a meta-model (in a meta-model aggregation system setup) compare to its performance when operating independently?

Based on the experiments, the GPT-5 model achieved an accuracy of 62.4% as a standalone model, while its accuracy increased to 71.4% when acting as a meta-model in a meta-model aggregation system (a relative increase of 14.4%). Similarly, the GPT-5 Mini model achieved an accuracy of 62.34% as a standalone model, while its accuracy also increased to 70.32% (a relative increase of 12.8%) when acting as a meta-model.

These results show the beneficial influence of the base models' recommendations on both models, indicating that they can be positively influenced by the recommendations of other models in a meta-model aggregation setup.

4.8. Decision-Making Strategy Analysis

RQ8: Does the meta-model primarily rely on majority voting, or does it reason directly over textual content? How often does each occur?

The GPT-5 meta-model shows a predominant tendency to align with majority decisions, with 4292 cases matching majority vote, compared to 708 cases of independent reasoning. When considering other central tendency metrics, the meta-model's predictions align with the rounded mean of the models' outputs in 4241 cases and with the rounded median in 4252 cases (Table 10). Importantly, its performance actually improves when diverging from the majority, achieving an accuracy of 77.68% in divergent cases compared to 70.36% when following the majority.

The GPT-5 mini meta-model shows a slightly different pattern, with 4227 cases matching the majority vote and 773 cases of independent reasoning. In terms of central tendency, it aligns with the rounded mean in 4229 cases and the rounded median in 4235 cases. However, its performance pattern inverts when diverging from the majority, achieving 68.69% accuracy in divergent cases compared to 70.62% when following the majority. This

indicates that the GPT-5 Mini tends to make independent decisions more often than the GPT-5; however, these decisions are proven to be less successful than the ones from its more sophisticated counterpart.

Table 10. Meta-Model Alignment with Central Tendency Metrics (Number of Matches).

Metric	GPT-5 Meta-Model	GPT-5 Mini Meta-Model
Mean (Rounded)	4241	4229
Median (Rounded)	4252	4235
Mode (Majority)	4281	4218

The different behaviors reveal that the two meta-models used in our study exhibit a fundamental divergence in their decision-making capabilities. The GPT-5 meta-model is more adept at recognizing when the majority opinion may be misleading and, as a result, independent reasoning is more likely to yield the correct results. In contrast, the GPT-5 mini meta-model performs better when aligning with majority opinions, demonstrating a more conservative approach in its independent reasoning.

4.9. Outlier Model Behavior Analysis

RQ9: Which models behave as outliers, potentially disrupting the overall meta-model aggregation system?

When using the GPT-5 meta-model, we identified four models as potential system disruptors, due to their consistent lower agreement with the meta-model's decisions, while simultaneously exerting a stronger negative influence on system performance. These models are the Gemini 2.5 Flash, Gemini 2.5 Flash Lite, Claude Haiku 4.5, and DeepSeek Reasoner.

On the other hand, when GPT-5 Mini served as the meta-model, we identified a different and larger set of five outlier models: Gemini 2.5 Pro, Gemini 2.5 Flash Lite, GPT-5 Nano, Claude Haiku 4.5 and Claude Opus 4.1. This broader identification of outliers suggests that the GPT-5 mini meta-model may be more sensitive to patterns of disagreement or may have a lower threshold for what constitutes disruptive behavior.

Notably, some models appear as outliers in both configurations, particularly the Gemini 2.5 Flash Lite and Claude Haiku 4.5, suggesting that these models may have inherent characteristics that make them more likely to diverge from the consensus, regardless of the meta-model. However, the difference in outlier identification between the meta-models is particularly interesting. The GPT-5 meta-model appears more tolerant of variation in model behavior, identifying fewer outliers and primarily focusing on models from the Gemini family and lighter versions of other model families.

4.10. Pre-Trained LLM Capability Assessment

RQ10: Are pre-trained LLMs without fine-tuning capable of accurately capturing sentiment from user feedback?

Based on the experiments, we observe an average accuracy of 61.25% across all models. The highest and lowest accuracies were achieved by the Claude Sonnet 4.5 and the Gemini 2.5 Flash models, at 65.02% and 56.86%, respectively. The significant difference of 8.16% between the pre-trained LLMs shows that pre-trained LLMs, without fine-tuning, exhibit varying capabilities when handling sentiment analysis tasks.

Additionally, both meta-model configurations produced similar and consistent results, showing that pre-trained LLMs possess innate capabilities to analyze sentiment. Due to their high level of accuracy (greater than 60% in most instances), we can infer that the

pre-trained LLMs developed a robust general language understanding that contributed to their capability to analyze sentiment in spite of no task-specific fine-tuning.

Further, based upon the results, we are able to observe that the performance ranking of the LLMs is consistent across both meta-model configurations, with the highest performance being the Claude models (Claude Sonnet-4.5 at 65.02% and Claude-Opus-4.1 at 64.48%), followed by the OpenAI GPT models. Interestingly, the Google Gemini models demonstrated the greatest variability in performance.

Finally, as the performance order of the LLMs was constant, it is possible to infer that architectural and training choices made in developing the LLMs have a lasting impact on their capability to analyze sentiment, regardless of whether the LLM has been optimized for the particular task.

Also, there is a large performance gap between the average individual model performance of 61.25% and the two aggregated performances of 71.40% for GPT-5 and 70.32% for GPT-5 Mini. As such, while the pre-trained LLMs demonstrated their ability to analyze meaningful sentiment, this significant performance gap suggests that pre-trained LLMs can greatly benefit from aggregation. Thus, collaboration with pre-trained models can provide an effective method of overcoming limitations in analyzing sentiment without the requirement of fine-tuning the models.

4.11. Cost-Effectiveness Analysis of Meta-Model Aggregation Approaches

RQ11: Do observed accuracy improvements justify the added computational complexity and cost of meta-model aggregation?

Figure 4 and Table 11 present a direct comparison of accuracy and the associated cost per model for 5000 sample predictions. We observe that the Gemini-2.5-Pro is the most expensive model while delivering relatively low accuracy, whereas the Claude Sonnet-4.5 achieves a better balance between accuracy and cost, making it more cost-effective in this evaluation.

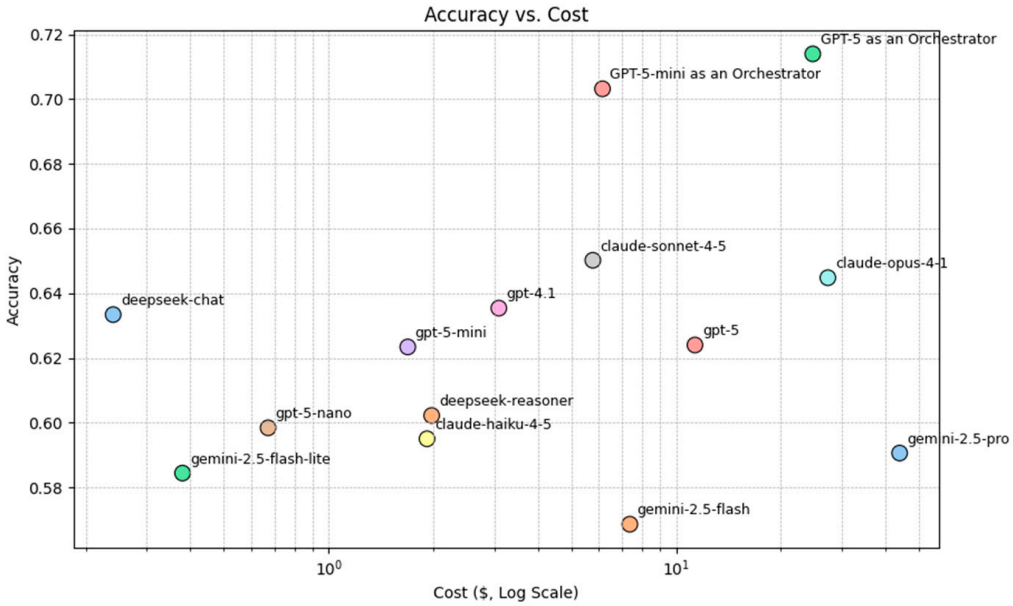

Figure 4. Comparison of model accuracy versus cost of computations per 5000 predictions.

For these models, the cost shown in Figure 4 and Table 11 reflects the expense of API calls for the meta-model itself. While not explicitly depicted in the figure, the total cost of a meta-model aggregation system includes the cost of the meta-model plus the cost of each individual model under aggregation. More specifically, for the GPT-5 meta-model,

the total system cost (including all models' API costs) is $130.45, while for the GPT-5 mini meta-model, the total system cost is $111.86.

Table 11. Comparison of models' accuracy, cost and execution time.

Model	Accuracy	Cost (USD)	Total Time (min)
Gemini 2.5 Pro	0.5906	43.97	819.43
Gemini 2.5 Flash	0.5686	7.36	444.43
Gemini 2.5 Flash Lite	0.5844	0.38	64.34
GPT-5	0.6240	11.33	784.09
GPT-5 Mini	0.6234	1.69	1032.75
GPT-5 Nano	0.5984	0.67	339.62
GPT-4.1	0.6354	3.09	118.36
Claude Sonnet 4.5	0.6502	5.76	221.25
Claude Haiku 4.5	0.5950	1.92	117.09
Claude Opus 4.1	0.6448	27.33	320.86
DeepSeek Chat	0.6334	0.24	139.77
DeepSeek Reasoner	0.6022	1.98	2609.27
GPT-5 as meta-model	0.7140	130.45	7795.36
GPT-5 Mini as meta-model	0.7032	111.86	8044.02

Comparing these costs with the corresponding increases in prediction accuracy (up to a 10% improvement), the value of meta-model aggregation ultimately depends on how much a business prioritizes additional accuracy relative to its operational budget.

4.12. Model Similarity and Redundancy Analysis

RQ12: How close are the predictions of different LLMs, and can we reduce costs by omitting models with similar outcomes?

To address this question, we investigated the similarity between the predictions of the 12 LLMs using Normalized Mean Absolute Error (NMAE) and Agreement Rate. The analysis results, visualized as a confusion matrix in Figure 5, revealed that the most similar pair of models is Claude Sonnet 4.5 and Claude Opus 4.1, with an NMAE of 0.030 and an agreement rate of 88%. This high level of similarity suggests that these two models often provide similar predictions.

Based on this finding, we simulated a cost-reduction strategy where the more expensive model, Claude Opus 4.1, was omitted from the ensemble. To maintain the numerical balance of the voting system, the vote of the remaining similar model, Claude Sonnet 4.5, was doubled. The simulation results showed that the baseline accuracy of the 12-model ensemble (62.64%) was maintained, and even slightly improved to 62.80% in the reduced 11-model configuration. This finding suggests that identifying and removing redundant models is a viable strategy for optimizing the cost-efficiency of meta-model aggregation systems without compromising performance. Table 12 presents the top 5 most similar model pairs identified in our analysis, from which the most prominent candidates for removal can be drawn. Interestingly, maintaining GPT-4.1 could allow the removal of up to three other models (Claude Sonnet-4.5, GPT-5 and Claude-Opus-4.1), while another option would be to maintain Claude Sonnet-4.5 are remove up to three different models (Claude-Opus-4.1, GPT-4.1 and DeepSeek-chat). The choice could take into account both cost and accuracy factors (Claude Sonnet-4.5 is more expensive and more accurate than GPT-4.1).

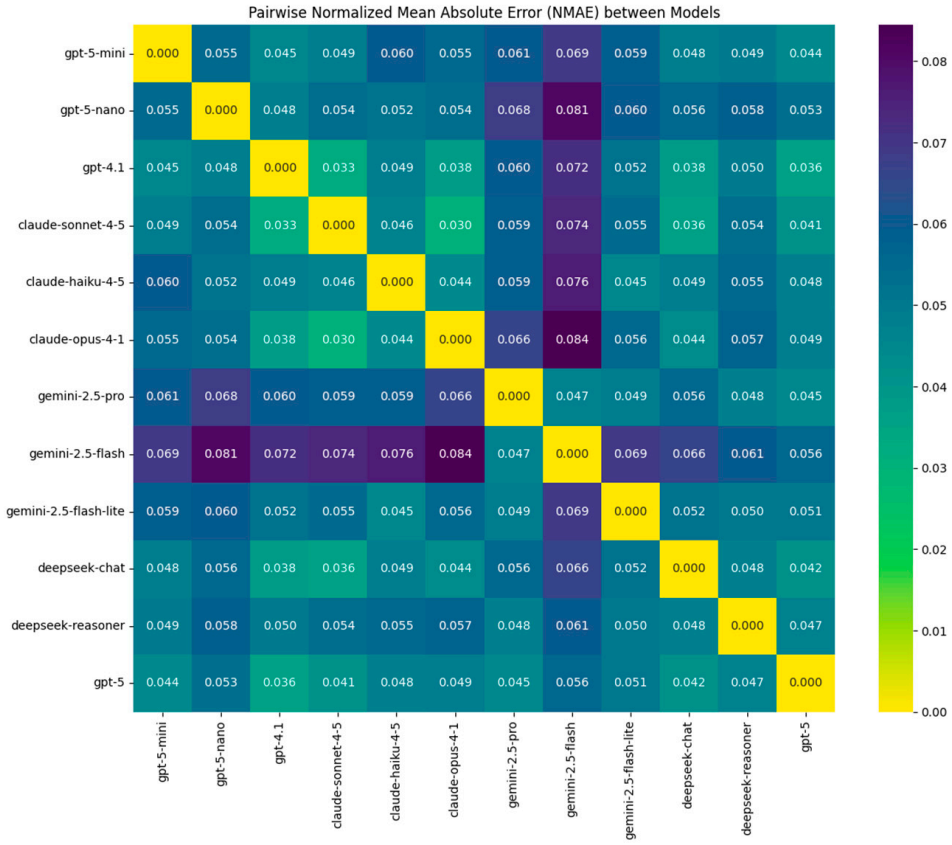

Figure 5. Pairwise Normalized Mean Absolute Error (NMAE) between Models.

Table 12. Top 5 Most Similar Model Pairs based on NMAE.

Model A	Model B	NMAE	Agreement
Claude Sonnet-4.5	Claude-Opus-4.1	0.03	0.88
GPT-4.1	Claude Sonnet-4.5	0.033	0.87
GPT-4.1	GPT-5	0.036	0.859
Claude Sonnet-4.5	DeepSeek-chat	0.036	0.859
GPT-4.1	Claude-Opus-4.1	0.038	0.85

4.13. Comparison with Fine-Tuned Baseline

RQ13: How does the performance of zero-shot LLMs and the meta-model aggregation approach compare to traditional fine-tuned transformer models?

To contextualize the zero-shot LLM performance and aggregation effectiveness, we fine-tuned RoBERTa-base as a supervised learning baseline. Using the remaining 45,000 balanced reviews (excluding the 5000 test set), we performed hyperparameter tuning on Google Colab Pro (A100 GPU) with a stratified 80/20 train-validation split. Six configurations were tested (learning rates: 2×10^{-5}, 3×10^{-5}, 5×10^{-5} $\times$ batch sizes: 16, 32) with a 3-epoch maximum and early stopping (patience = 2). The best configuration (learning rate 5×10^{-5}, batch size 16) achieved 63.03% validation accuracy and 64.00% test accuracy.

RoBERTa's performance positioned it within the range of zero-shot LLMs (56.86–65.02%) but substantially below aggregation approaches. Specifically: (1) RoBERTa outperformed 7 of 12 zero-shot LLMs, (2) underperformed the top 5 zero-shot models (Claude Sonnet 4.5: 65.02%, Claude Opus 4.1: 64.48%, GPT-4.1: 63.54%), and (3) trailed meta-models by 7.40 and 6.32 percentage points, respectively. Despite direct exposure to 36,000 training examples (7200 per rating), RoBERTa's performance remained comparable

to mid-tier zero-shot LLMs, suggesting fundamental challenges in rating prediction—particularly for neutral sentiment (2-star: 53% recall, 3-star: 53% recall)—that supervised fine-tuning alone cannot fully resolve. It is crucial, though, to mention that by experimenting more with extensive hyperparameter tuning through Grid-search or Bayesian optimization, the above results could be optimized, bringing greater accuracy.

This comparison reveals three key findings: (1) aggregation provides meaningful accuracy improvements even versus specialized fine-tuned models, (2) zero-shot LLM performance is competitive with supervised approaches despite lacking task-specific training, and (3) the 10.15 percentage point aggregation improvement compounds architectural benefits with model diversity. While RoBERTa offers substantially lower inference cost (milliseconds vs. seconds per prediction), the accuracy-cost trade-off depends on deployment context. Future work could explore whether advanced prompting techniques (Few-Shot, Chain-of-Thought) further widen performance gaps or whether hybrid approaches combining fine-tuned models with aggregation yield additional gains.

4.14. Model Agreement and Consensus Patterns

RQ14: What patterns of agreement exist among the 12 base models, and how does the consensus level relate to prediction accuracy?

To rigorously quantify inter-model agreement, we computed Fleiss' Kappa across all 12 models, yielding $\kappa = 0.5921$ (moderate agreement according to Landis & Koch interpretation guidelines [68]). This indicates that while models show statistically significant agreement beyond chance (observed agreement $\overline{P} = 0.7036$, expected agreement $P_e = 0.2385$), substantial disagreement persists across the ensemble. Agreement varied significantly by rating level: 1-star ($\kappa = 0.5782$) and (especially) 5-star ($\kappa = 0.6294$) reviews achieved higher consensus compared to neutral 3-star reviews ($\kappa = 0.5542$), consistent with the neutral sentiment challenge discussed in Section 4.2.

Pairwise Cohen's Kappa analysis revealed model-specific agreement patterns. The highest-agreeing pairs were Claude Sonnet 4.5–Claude Opus 4.1 ($\kappa = 0.849$, 88.00% raw agreement) and GPT-4.1–Claude Sonnet 4.5 ($\kappa = 0.836$, 87.04%), suggesting architectural similarity within model families. The lowest-agreeing pairs involved Gemini 2.5 Flash with other models ($\kappa = 0.582$–0.671), indicating distinct prediction behavior. Individual model average kappa ranged from 0.659 (Gemini 2.5 Flash) to 0.768 (GPT-5), with higher average agreement correlating with individual model accuracy (Pearson's $r = 0.72$, $p < 0.01$).

Consensus level analysis categorized predictions into five tiers based on majority vote strength: Unanimous (12/12 models agree: 43.36% of cases), Strong Majority (10–11/12: 25.84%), Majority (8–9/12: 18.58%), Weak Majority (6–7/12: 12.02%), and No Majority (<6/12: 0.20%). Prediction accuracy increased monotonically with consensus: unanimous cases achieved 80.12% accuracy versus 47.26% for standard majority and 30.00% for no-majority cases. This pattern is consistent with the aggregation rationale—high consensus provides signal quality, whereas disagreement flags ambiguous cases requiring meta-model reasoning.

Extreme divergence cases (predictions spanning 1–5 stars, n = 4, 0.08%) highlighted linguistic complexity. For example, the rating text "I'm used to new products being the same price, functionally, as used items. I'm just always blown away when something like this happens" (true: 2-star) elicited predictions from 1 to 5 stars due to sarcasm and implicit negativity contradicting surface-level neutral phrasing. Such cases underscore why pure majority voting (62.64% accuracy) underperforms meta-model reasoning (71.40%), as aggregation must identify when consensus itself is unreliable due to shared systematic errors across base models.

4.15. Meta-Model Override Behavior and Decision Triggers

RQ15: Under what conditions do meta-models override the majority vote, and how does this relate to prediction accuracy?

Meta-models exhibited distinct override behaviors with critical accuracy implications. GPT-5 overrode the majority in 14.38% of cases (719/5000), achieving 78.03% accuracy in those instances compared to 70.29% when following the majority—a 7.74 percentage point improvement. Conversely, GPT-5 Mini overrode more frequently (15.64%, 782/5000) but with lower success (69.18% accuracy when overriding versus 70.53% when following, a −1.35 percentage point deficit). This reveals that override frequency alone does not guarantee improvement; decision quality matters more than decision volume.

Override success varied systematically by rating level. GPT-5 achieved its highest override accuracy on extreme ratings: 95.12% (5-star, 78/82 overrides) and 90.77% (1-star, 59/65 overrides), versus 70.40% on neutral 3-star reviews (176/250 overrides). This aligns with the neutral sentiment challenge—meta-models struggle to correct majority errors when the ground truth itself is ambiguous. By contrast, extreme ratings provide clearer linguistic signals that enable confident deviation from consensus.

Override triggers analysis revealed that meta-models disproportionately override under high base-model disagreement. When GPT-5 overrode the majority, base models exhibited a mean prediction standard deviation of 0.408, compared to 0.215 when following (difference = 0.193, t-test $p < 0.001$). Similarly, GPT-5 Mini showed a mean disagreement of 0.421 (override) versus 0.210 (follow), a difference of 0.211 ($p < 0.001$). This indicates that disagreement among base models serves as an implicit confidence signal: meta-models are more likely to trust their independent analysis when the ensemble lacks strong consensus.

Majority strength analysis quantified this pattern: when all 12 models agreed (unanimous majority), GPT-5 overrode in 4.5% of cases (97/2168) with 100% accuracy—exclusively correcting rare systematic errors. As the majority strength weakened, override rates increased: 6–7 models agreeing (weak majority) triggered 18.9% override rate (183/969), though accuracy dropped to 73.8%. This suggests optimal aggregation involves calibrated override thresholds: strong consensus should be trusted absent compelling contradictory evidence, whereas weak consensus warrants skeptical re-evaluation.

Qualitative override examples illustrate reasoning sophistication. Correct override (GPT-5): True rating = 2 stars, majority predicted 3 stars, GPT-5 predicted 2 stars. For instance, when processing the review: "It comes with a lot of items related to the game. Very satisfied with the quality of items; however, the packaging is a different story. The box was damaged…" GPT-5 reasoning identified that negative framing of packaging issues outweighed positive product mentions, whereas the majority of models averaged positive and negative signals toward neutrality. On the other hand, an incorrect override (GPT-5 Mini) occurred for the following case, where the True rating was 1 star, the majority predicted 1 star, but GPT-5 Mini overrode the decision, predicting 2 stars. With the review text being "Right off the bat, before I even use this thing, it's missing the thumb stick covers…" GPT-5 Mini reasoning treated missing components as a moderate defect rather than a critical failure, misaligning with reviewer intent. These cases demonstrate that meta-model override success depends on correctly weighting sentiment intensity, not merely detecting valence.

5. Discussion

This study evaluates meta-model aggregation systems versus independent LLMs in sentiment analysis for RecSys, uncovering key insights into performance, cost-efficiency, and practical implications of LLM aggregation.

5.1. Performance Enhancement Through Aggregation

The most important finding is the considerable improvement in performance achieved through meta-model aggregation. Both meta-models used in this study (GPT-5 and GPT-5 Mini) succeeded in achieving considerable accuracy gains, measured at 10.15% and 9.07% in absolute terms (or 16.6% and 14.8% as relative improvements), respectively, when compared to the average performance of individual models, which was measured at 61.25%. The small (almost negligible) difference between the two meta-models indicates that effectiveness can be achieved even with lightweight, and thus cost-effective, models, meaning there is no need for organizations to deploy the most expensive models as meta-models.

5.2. The Neutral Sentiment Challenge

All 12 LLMs faced extreme challenges in predicting neutral (i.e., 3-star) ratings, showing an average failure rate of 64.83% (almost 2 out of every 3 predictions are misclassified), which is considered significant. This highlights the fundamental challenge of classifying neutral or mixed sentiment expressions in sentiment analysis research. Furthermore, the substantial range in failure rates (ranging from 55.7% to 77.4%) indicates that certain architectural approaches perform considerably better in handling neutral sentiment compared to others. 3-star reviews often contain more balanced or contradictory statements requiring a more detailed understanding, unlike extreme ratings (either praising or strongly unfavorable), which have clear linguistic characteristics. Although meta-model aggregation contributes to resolving this challenge, it remains a critical area that future research should focus on.

5.3. Model Specialization and Reasoning Trade-Offs

Surprisingly, reasoning-oriented models performed worse than their chat-oriented counterparts. More specifically, DeepSeek Reasoner underperformed compared to DeepSeek Chat, while GPT-5 showed 1.14% lower accuracy than GPT-4.1. Based on these findings, we can conclude that the additional computational overhead of reasoning-focused architectures in sentiment analysis tasks does not result in corresponding benefits. This could also be because reviews may typically be of short length, which appears to be more akin to a conversation rather than an essay or a complex document; therefore, the style and content of a review may be a better fit for conversational models. Furthermore, there appears to be a potential bias in reasoning models towards negative sentiments, which is inconsistent with the emotional distribution in real-world reviews, which tend to be much more balanced. In conclusion, organizations should carefully consider whether the additional cost and latency of reasoning models are justified for specific use cases.

5.4. Meta-Model Decision-Making and Independent Models' Influence

The meta-models demonstrated sophisticated behavior, which was superior to simple majority voting. More specifically, when the GPT-5 meta-model diverged from the consensus, it achieved an accuracy of 77.68%, whereas it achieved 70.36% accuracy when it followed the majority. This suggests that effective aggregation also involves critically evaluating potentially misleading consensus. Furthermore, the meta-models established implicit trust hierarchies, consistently identifying outlier models (Gemini 2.5 Flash Lite and Claude Haiku 4.5). Finally, the perfect accuracy (100%) achieved by GPT-5, when it ignored the models' recommendations, indicates an advanced meta-cognitive ability to recognize when collective input is less reliable than its own analysis.

An evaluation of the reasoning behind the decision-making process was made using the reasoning text that was generated from the meta-models through a systematic collection

and storage of the reasoning text in designated columns within the prediction phase. A comparison of the reasoning style of the two models has shown significant differences.

The GPT-5 mini meta-model produces considerably more text than the GPT-5 meta-model, as evidenced by a 62% greater mean reasoning length (91.15 words) than the GPT-5 meta-model (56.44 words). In many instances, this additional text is directed toward the detailed examination of outliers, as indicated by the term "outlier" being referenced 1693 times as opposed to 376 times in GPT-5.

A qualitative review of the reasoning logs revealed how the meta-models handled conflict resolution. For example, in cases where there was a split decision among the models, both the GPT-5 Mini and GPT-5 meta-models provided explicit comparisons of the reviewers' textual information versus their own predictive assessments.

- Example of Consensus (GPT-5): "All models unanimously predicted 1 star with no deviations. The review is strongly negative, citing safety concerns. . . and explicit non-recommendation. . . The severity of the issue. . . clearly aligns with a 1-star rating."
- Example of Conflict Resolution (GPT-5): "Models split evenly between 1-star and 2-star. . . The 2-star predictors. . . deviate from the strongly negative tone. The review reports failure after limited use. . . clear non-recommendation. Despite a brief 'worked great' at first, overall severity and dissatisfaction align with 1 star."

These examples demonstrate that the meta-models do not merely aggregate votes but actively reason about the content to resolve ambiguities, explaining why certain models (e.g., those predicting 2 stars) might have been misled by specific phrases (e.g., "worked great").

5.5. Cost–Benefit Analysis

Overall, the cost of the entire system was $130.45 & $111.86 using GPT-5 & GPT-5 Mini as aggregation models (using 5000 predictions). That is substantially more than the cost of the individual LLMs, whose cost ranged from $0.24 (DeepSeek Chat) to $43.97 (Gemini 2.5 Pro). In addition, the total time to execute the model increased drastically, taking 7795 & 8044 min in comparison to the time taken by each individual model (taking anywhere from 64 to 2609 min). On average, this translates into a latency of approximately 1.6 min per request. It is worth noting, however, that the above-stated latency can be greatly improved by making simultaneous calls to the base models. As such, if the system were to make simultaneous calls on all of the LLMs, the total execution time would theoretically reach the amount of time taken by the slowest model (DeepSeek Reasoner), plus the time taken by the meta-model to aggregate the results (approximately 784 min for GPT-5 and 1032 min for GPT-5 mini). The time taken to serve the 5000 requests would therefore be approximately 3393 min for GPT-5 and 3642 min for GPT-5 mini, with the corresponding latencies per request being 0.68 and 0.73 min, respectively. Although there is a greater latency, the accuracy improvement was approximately 10%, representing nearly 500 additional correct predictions, which are very important in applications where the accuracy of the prediction directly influences business results. The meta-model aggregation method provides a "training-free" way to enhance performance without requiring the substantial labeling of data sets or computational resources needed to fine-tune the models. In summary, the accuracy gain may outweigh the cost increase in high-stakes applications; whereas in low-stakes applications, individual models will provide the best value based on cost vs. benefit.

To contextualize these costs in production-scale RecSys environments, we calculate the cost-per-accuracy-point improvement: The GPT-5 meta-model achieves a 10.15% accuracy gain over the average individual model (61.25%) at $130.45 per 5000 reviews, yielding $12.86 per percentage point improvement. For the GPT-5 mini meta-model, the 9.07%

improvement at $111.86 translates to $12.33 per percentage point. When compared to the best-performing individual model (Claude Sonnet 4.5 at 65.02% accuracy, costing $5.76), the meta-model's marginal gain of 6.38 percentage points (from 65.02% to 71.40%) costs an additional $124.69, or $19.54 per marginal percentage point. These metrics reveal that the cost efficiency of aggregation depends heavily on the baseline being compared.

Regarding production scalability, processing millions of reviews daily would indeed incur substantial costs. Extrapolating to 1 million reviews (200× our test set), the GPT-5 meta-model would cost approximately $26,090, and the GPT-5 mini meta-model approximately $22,372. For organizations processing 10 million reviews daily, annual costs would exceed $95 million (GPT-5) or $81 million (GPT-5 Mini)—financially prohibitive for most real-world deployments. However, several practical deployment strategies can mitigate these costs: (1) Selective Aggregation: Apply the full 12-model ensemble only to high-stakes scenarios (e.g., disputed reviews, flagged content, product launches) while using individual models or reduced ensembles for routine classification. (2) Model Pruning: As demonstrated in Section 4.12, removing redundant models (e.g., highly similar pairs like Claude Sonnet 4.5 and Claude Opus 4.1) maintains performance while reducing costs proportionally. A 6-model ensemble would halve aggregation expenses. (3) Confidence-Based Routing: Implement a two-tier system where individual models handle high-confidence predictions (e.g., >90% softmax probability), escalating only uncertain cases to aggregation. (4) Batch Optimization: Amortize meta-model overhead across larger batches and leverage parallel API calls to reduce per-review latency from 1.6 min to <1 min. (5) Hybrid Architectures: Fine-tune smaller models (e.g., BERT, RoBERTa) on high-confidence meta-model predictions, creating cost-effective specialized classifiers for production while reserving aggregation for challenging cases.

In conclusion, while the current full-ensemble aggregation is economically challenging for large-scale continuous deployment, the architecture's primary value lies in (a) research benchmarking to establish performance ceilings, (b) high-stakes applications where accuracy justifies cost (e.g., content moderation, fraud detection, medical sentiment analysis), and (c) training data generation where meta-model predictions can be used to fine-tune efficient production models. Organizations must carefully assess their accuracy-cost trade-offs: for applications where a 10% accuracy improvement translates to significant business value (e.g., reducing false negatives in safety-critical domains), aggregation becomes economically viable; for low-stakes sentiment analysis at scale, individual models or pruned ensembles offer better cost–benefit ratios.

5.6. Practical Implications

The main practical implications are the following:

1. Organizations should deploy multiple models instead of focusing on finding a single 'optimal' model, as lower-performing models can still provide valuable insights.
2. Lightweight meta-models (like the GPT-5 Mini) offer attractive performance-cost balance and may prove most suitable for lower-stakes applications.
3. Due to the high failure rate in classifying neutral or mediocre opinions (i.e., 3-star ratings), production systems should implement additional safeguards when encountering these cases (e.g., higher confidence thresholds or human review queues).
4. Organizations should carefully assess the business value of accuracy improvements in relation to aggregation costs, depending on the application's stakes.

5.7. Limitations and Future Directions

This study provides a comprehensive evaluation of meta-model aggregation for product review sentiment analysis using Amazon reviews. While our findings demonstrate

clear performance improvements within this context, generalizability to other tasks and domains requires empirical validation. Specifically, our results are constrained to:

- Task Scope: Our evaluation focuses exclusively on product review sentiment analysis—the task of predicting numerical star ratings (1–5) from textual reviews in an e-commerce context. This focused scope allows rigorous controlled evaluation within a single well-defined task. Future work could investigate whether meta-model aggregation provides similar benefits for other NLP tasks, including text summarization, question answering, dialogue generation, named entity recognition, or open-domain text classification.
- Output Format Specificity: The task involves predicting discrete ordinal ratings on a bounded 1–5 scale, providing a clear quantitative evaluation metric. This structured output format enables rigorous statistical analysis and direct performance comparison. Extensions to other sentiment analysis formulations would be valuable future work, including: (a) open-ended sentiment description where models generate free-text sentiment explanations rather than scale-based ratings, (b) aspect-based sentiment analysis where multiple sentiment dimensions must be evaluated simultaneously (e.g., product quality, shipping speed, customer service), or (c) fine-grained emotion detection that extends beyond the positive-negative-neutral spectrum to identify specific emotional states (joy, anger, frustration, disappointment).
- Data Source Specificity: All reviews originate from the Amazon e-commerce platform across five product categories (Fashion, Automotive, Books, Electronics, Video Games), providing substantial domain coverage within e-commerce contexts. This domain focus ensures consistent evaluation conditions and reduces confounding variables. Validation with datasets from different providers and of varying form or nature represents a natural extension of this work, with promising directions including: (a) movie reviews (e.g., MovieLens, IMDb) where sentiment may be expressed differently than in product reviews, (b) restaurant reviews (e.g., Yelp) which often emphasize subjective experiences like ambiance and service quality, (c) social media content (e.g., Twitter, Reddit) characterized by informal language, hashtags, and cultural references, or (d) news article sentiment where political bias and journalistic framing introduce distinct challenges.
- Language Limitation: This study evaluates English-language reviews exclusively. The performance of zero-shot LLMs and the effectiveness of meta-model aggregation may differ substantially across languages due to: (a) training data imbalance, where English typically dominates pre-training corpora, (b) linguistic structure differences (e.g., sentiment expression in morphologically rich languages, context-dependent languages like Greek, Chinese and Japanese), and (c) cultural variation in how sentiment and product satisfaction are expressed textually. Multilingual evaluation and cross-lingual transfer experiments would be required to assess generalizability beyond English.
- Temporal and Versioning Constraints: Our measurements represent performance snapshots using specific model versions available in late 2025 (Table 3). As acknowledged in our API-based evaluation limitations discussion, model providers continuously update their systems, which may affect both absolute performance levels and relative rankings. Additionally, future model architectures with improved reasoning capabilities, longer context windows, or domain-specific fine-tuning may alter the cost–benefit calculus of meta-model aggregation. Our findings should be interpreted as characterizing the specific model ecosystem evaluated rather than establishing permanent truths about LLM meta-model aggregation effectiveness.

The above scope decisions enable rigorous controlled evaluation within a well-defined context. Our findings demonstrate that meta-model aggregation with natural language

reasoning outperforms traditional ensemble methods and individual models for Amazon product review sentiment prediction. These results establish a foundation for future work examining whether similar benefits extend to other tasks, domains, languages, or future model generations.

Data Contamination Considerations: We evaluate models on the Amazon Reviews '23 dataset, a publicly available benchmark. While LLMs with 2024/2025 training cutoffs may have encountered these reviews during pre-training, the observed zero-shot accuracies (ranging from 56.86% to 65.02% for individual models) suggest limited memorization; extensive training on this specific test set would likely produce substantially higher performance approaching near-perfect accuracy. This performance range is consistent with genuine zero-shot evaluation, though as with all contemporary LLM studies on public datasets, complete certainty about training data overlap remains infeasible.

API-Based Evaluation Limitations: This study evaluates commercial API-based LLM services, documenting all model versions with timestamps (Table 3) and using temperature = 0 for deterministic sampling. API providers may modify underlying models, update endpoints, or adjust pricing structures over time. Our measurements represent performance indicators captured at a specific point in time (late 2025) using the model versions listed in Table 3, enabling rigorous comparative evaluation under identical conditions. This approach reflects realistic production deployment scenarios where organizations typically access LLMs through commercial APIs rather than hosting models locally. While API-based inference may exhibit residual non-determinism due to distributed computing architectures and inference optimizations, all models were evaluated under identical conditions within the same temporal window, ensuring valid comparative conclusions.

Ensemble-Based Nature of Approach: Our approach represents a specific instantiation of meta-model aggregation within the established ensemble learning paradigm, where the meta-model processes predictions through natural language reasoning rather than learned weights. This reasoning capability distinguishes our approach from traditional stacking methods while remaining grounded in ensemble learning principles. To clarify scope: we do not introduce novel agent architectures, autonomous goal-pursuit mechanisms, distributed decision-making protocols, or tool-use capabilities that characterize true agentic AI systems. Rather, the contribution lies in the empirical demonstration that meta-model reasoning provides measurable accuracy improvements over statistical aggregation methods when applied to sentiment classification.

Absence of Formal Agent Model: This work focuses on meta-model aggregation for sentiment analysis, positioning itself within ensemble learning and LLM evaluation literature rather than multi-agent systems paradigms. To clarify terminology: our use of "orchestration" in the title refers specifically to meta-model aggregation—the process by which a reasoning-capable LLM combines predictions from multiple independent models. This differs from multi-agent coordination in which autonomous agents negotiate, communicate, and collaborate. The base models function as independent zero-shot inference systems accessed via APIs, processing inputs and generating predictions without maintaining state or learning from experience. For completeness, we note that our system does not implement formal agent properties such as: (1) belief-desire-intention (BDI) models or other formal agent architectures that maintain internal mental states, (2) autonomous learning loops that enable adaptation without human intervention, (3) environment interaction cycles where agents perceive, reason, and act dynamically, or (4) dynamic goal adjustment mechanisms. This positioning clarifies that our contribution lies in the empirical evaluation of meta-model aggregation effectiveness rather than agent architecture innovation.

Text Normalization Trade-offs: Our preprocessing pipeline applied traditional text normalization (lowercase conversion, special character removal) to standardize inputs

across 100M+ source reviews, reducing variance from inconsistent formatting artifacts. While modern LLMs are trained on raw text and can process natural formatting directly, normalization ensures consistent experimental conditions across all models and domains. An interesting direction for future work involves investigating whether preserving natural text formatting (capitalization for emphasis like "AMAZING product," repeated punctuation indicating strong emotion like "terrible!!!", or acronyms carrying sentiment weight like "OMG") yields improved sentiment classification accuracy with contemporary LLMs.

Execution Time Measurement and Network Variability: Our execution time measurements capture the complete operational cost of obtaining predictions (network latency, server-side inference, and minimal JSON parsing overhead), providing realistic production-deployment estimates. All measurements were conducted from a single geographic location with consistent network infrastructure, ensuring fair within-study relative comparisons. These measurements serve as relative performance indicators; actual execution times may vary across different deployment environments due to network conditions. Future work comparing execution efficiency across models could explore controlled network conditions or isolated inference-only measurements to further reduce latency variance.

Prompting Methodology: Our study employs zero-shot prompting with structured JSON output for both individual models and the meta-model, prioritizing a fair comparative baseline across heterogeneous model families (GPT, Claude, Gemini, DeepSeek) while isolating the architectural contribution of aggregation from prompt optimization effects. This approach provides production realism and unbiased model comparison. The meta-model prompt elicits reasoning implicitly by requiring structured explanations of model deviations, consensus assessment, and decision justification, sharing conceptual similarities with Chain-of-Thought while embedding reasoning requirements within the task structure. Advanced prompting techniques—including explicit Chain-of-Thought (CoT) reasoning, Few-Shot exemplars, Self-Consistency with multiple sampling paths, and Self-Correction through iterative refinement—represent promising directions for future work. Preliminary literature suggests these techniques could improve individual model accuracy by 5–15 percentage points, which would correspondingly affect meta-model performance. Our zero-shot aggregation baseline establishes a foundation for systematically comparing future prompt-engineered approaches and investigating whether architectural and prompt optimization benefits are additive, multiplicative, or exhibit diminishing returns.

Future research may consider:

1. Inclusion of datasets from a variety of sources to improve diversity;
2. Exploration of more advanced aggregation strategies, such as hierarchical aggregation, iterative refinement, and weighted voting;
3. Integration of fine-tuned models;
4. Evaluation of advanced prompting techniques (Few-Shot with rating exemplars to reduce 3-star errors, explicit CoT for improved transparency, Self-Consistency for multiple reasoning paths, multi-turn agent-to-agent communication);
5. Longitudinal tracking of aggregation effectiveness across model updates;
6. Understanding and employing user trust and enhancing explainability in ensemble-based recommendations;
7. Investigation of raw text processing without normalization to preserve sentiment cues; and
8. Evaluation on datasets with verifiable post-training-cutoff timestamps to eliminate data contamination concerns.

5.8. Broader Implications

This empirical evaluation shows that meta-model aggregation can achieve substantial performance improvements over individual models. The findings are consistent with ensemble approaches in the literature; however, these approaches operate at a semantic level (i.e., reasoning with respect to explanations as opposed to probability aggregation). The results indicate that model diversity may provide more value than individual excellence, suggesting that employing only the best-performing models is not necessarily always the best approach. A significant difference was observed between the performance of the zero-shot implementation (a 61.25% average) and the meta-model aggregation implementations (a 71.40% average). This shows that aggregation approaches can achieve strong results without requiring the need for task-specific training, which may be beneficial for organizations with limited machine learning experience and/or no access to large amounts of labeled data. Furthermore, the aggregation patterns exhibited by the meta-model aggregation mechanisms, which include the ability to synthesize the perspective of multiple models, override the consensus of the models, and develop hierarchical trust relationships among the models, exhibit characteristics similar to effective human team dynamics.

Author Contributions: Conceptualization, K.I.R., D.M., D.S. and C.V.; methodology, K.I.R., D.M., D.S. and C.V.; software, K.I.R. and D.M.; validation, K.I.R., D.M., D.S. and C.V.; formal analysis, K.I.R., D.M., D.S. and C.V.; investigation, K.I.R., D.M., D.S. and C.V.; resources, K.I.R., D.M., D.S. and C.V.; data curation, K.I.R., D.M., D.S. and C.V.; writing—original draft preparation, K.I.R. and D.M.; writing—review and editing, K.I.R., D.M., D.S. and C.V.; visualization, K.I.R. and D.M.; supervision, C.V.; project administration, K.I.R., D.M.; funding acquisition, K.I.R., D.M., D.S. and C.V. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Data supporting the reported results can be found at [60].

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Zhang, K.; Cao, Q.; Sun, F.; Wu, Y.; Tao, S.; Shen, H.; Cheng, X. Robust Recommender System: A Survey and Future Directions. *ACM Comput. Surv.* **2026**, *58*, 1–38. [CrossRef]
2. Shehmir, S.; Kashef, R. LLM4Rec: A Comprehensive Survey on the Integration of Large Language Models in Recommender Systems—Approaches, Applications and Challenges. *Future Internet* **2025**, *17*, 252. [CrossRef]
3. Zhao, Z.; Fan, W.; Li, J.; Liu, Y.; Mei, X.; Wang, Y.; Wen, Z.; Wang, F.; Zhao, X.; Tang, J.; et al. Recommender Systems in the Era of Large Language Models (LLMs). *IEEE Trans. Knowl. Data Eng.* **2024**, *36*, 6889–6907. [CrossRef]
4. Deldjoo, Y.; He, Z.; McAuley, J.; Korikov, A.; Sanner, S.; Ramisa, A.; Vidal, R.; Sathiamoorthy, M.; Kasirzadeh, A.; Milano, S. A Review of Modern Recommender Systems Using Generative Models (Gen-RecSys). In Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Barcelona, Spain, 25–29 August 2024; pp. 6448–6458.
5. Kiziltepe, R.S.; Ezin, E.; Yentür, Ö.; Basbrain, A.M.; Karakus, M. Advancing Sentiment Analysis for Low-Resource Languages Using Fine-Tuned LLMs: A Case Study of Customer Reviews in Turkish Language. *IEEE Access* **2025**, *13*, 77382–77394. [CrossRef]
6. Chinnalagu, A. Comparative Analysis of Fine-Tuned LLM, BERT and DL Models for Customer Sentiment Analysis. In Proceedings of the 2024 13th International Conference on System Modeling & Advancement in Research Trends (SMART), Moradabad, India, 6–7 December 2024; pp. 255–259.
7. Bandi, A.; Kongari, B.; Naguru, R.; Pasnoor, S.; Vilipala, S.V. The Rise of Agentic AI: A Review of Definitions, Frameworks, Architectures, Applications, Evaluation Metrics, and Challenges. *Future Internet* **2025**, *17*, 404. [CrossRef]
8. Sapkota, R.; Roumeliotis, K.I.; Karkee, M. AI Agents vs. Agentic AI: A Conceptual Taxonomy, Applications and Challenges. *Inf. Fusion* **2026**, *126*, 103599. [CrossRef]
9. Shavit, Y.; Agarwal, S.; Brundage, M.; Adler, S.; O'Keefe, C.; Campbell, R.; Lee, T.; Mishkin, P.; Eloundou, T.; Hickey, A.; et al. *Practices for Governing Agentic AI Systems*; OpenAI White Pap.; OpenAI: San Francisco, CA, USA, 2023.
10. Ghatora, P.S.; Hosseini, S.E.; Pervez, S.; Iqbal, M.J.; Shaukat, N. Sentiment Analysis of Product Reviews Using Machine Learning and Pre-Trained LLM. *Big Data Cogn. Comput.* **2024**, *8*, 199. [CrossRef]

11. Tian, Y.; Peng, B.; Song, L.; Jin, L.; Yu, D.; Han, L.; Mi, H.; Yu, D. Toward Self-Improvement of LLMs via Imagination, Searching, and Criticizing. In *Proceedings of the Advances in Neural Information Processing Systems*; Globerson, A., Mackey, L., Belgrave, D., Fan, A., Paquet, U., Tomczak, J., Zhang, C., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2024; Volume 37, pp. 52723–52748.
12. Sapkota, R.; Roumeliotis, K.I.; Pokhrel, S.; Karkee, M. From Self-Learning to Self-Evolving Architectures in Large Language Models: A Short Survey. *Authorea Prepr.* **2025**. [CrossRef]
13. Hou, Y.; Li, J.; He, Z.; Yan, A.; Chen, X.; McAuley, J. Bridging Language and Items for Retrieval and Recommendation. *arXiv* **2024**, arXiv:2403.03952. [CrossRef]
14. Zhang, F.; Luo, W.; Yang, X. Mitigating Selection Bias in Recommendation Systems Through Sentiment Analysis and Dynamic Debiasing. *Appl. Sci.* **2025**, *15*, 4170. [CrossRef]
15. Margaris, D.; Vassilakis, C.; Spiliotopoulos, D. What Makes a Review a Reliable Rating in Recommender Systems? *Inf. Process. Manag.* **2020**, *57*, 102304. [CrossRef]
16. Awati, C.J.; Shirgave, S.K.; Thorat, S.A. Improving Performance of Recommendation Systems Using Sentiment Patterns of User. *Int. J. Inf. Technol.* **2023**, *15*, 3779–3790. [CrossRef]
17. Liu, N.; Zhao, J. Recommendation System Based on Deep Sentiment Analysis and Matrix Factorization. *IEEE Access* **2023**, *11*, 16994–17001. [CrossRef]
18. Karabila, I.; Darraz, N.; El-Ansari, A.; Alami, N.; El Mallahi, M. Enhancing Collaborative Filtering-Based Recommender System Using Sentiment Analysis. *Future Internet* **2023**, *15*, 235. [CrossRef]
19. Sharma, A.; Vora, D.; Shaw, K.; Patil, S. Sentiment Analysis-Based Recommendation System for Agricultural Products. *Int. J. Inf. Technol.* **2024**, *16*, 761–778. [CrossRef]
20. Karabila, I.; Darraz, N.; EL-Ansari, A.; Alami, N.; EL Mallahi, M. BERT-Enhanced Sentiment Analysis for Personalized e-Commerce Recommendations. *Multimed. Tools Appl.* **2023**, *83*, 56463–56488. [CrossRef]
21. Wu, M.; Liu, W.; Wang, Y.; Yao, M. Negotiating the Shared Agency between Humans & AI in the Recommender System. In Proceedings of the Extended Abstracts of the CHI Conference on Human Factors in Computing Systems, Yokohama, Japan, 26 April–1 May 2025; pp. 1–9.
22. Sharma, V.; Limbachiya, Y.; Oza, D. Empowering Text Classification with Agentic AI: A Systematic Review. In Proceedings of the 2025 International Conference on Artificial Intelligence and Machine Vision (AIMV), Gandhinagar, India, 16–17 August 2025; pp. 1–6.
23. Zhang, J.; Hou, Y.; Xie, R.; Sun, W.; McAuley, J.; Zhao, W.X.; Lin, L.; Wen, J.-R. AgentCF: Collaborative Learning with Autonomous Language Agents for Recommender Systems. In Proceedings of the ACM Web Conference 2024, Singapore, 13–17 May 2024; pp. 3679–3689.
24. Liu, W.; Wang, Y. Evaluating Trust in Recommender Systems: A User Study on the Impacts of Explanations, Agency Attribution, and Product Types. *Int. J. Hum.–Comput. Interact.* **2025**, *41*, 1280–1292. [CrossRef]
25. Sharanarthi, T. AI Agent-Based Framework for Personalized Music Recommendation. In Proceedings of the 2025 World Skills Conference on Universal Data Analytics and Sciences (WorldSUAS), Indore, India, 22–23 August 2025; pp. 1–5.
26. Baek, T.H.; Kim, H.J.; Kim, J. AI-Generated Recommendations: Roles of Language Style, Perceived AI Human-Likeness, and Recommendation Agent. *Int. J. Hosp. Manag.* **2025**, *126*, 104106. [CrossRef]
27. Tebourbi, H.; Nouzri, S.; Mualla, Y.; Abbas-Turki, A. Artificial Intelligence Agents for Personalized Adaptive Learning. *Procedia Comput. Sci.* **2025**, *265*, 252–259. [CrossRef]
28. Xiao, K.; He, Y. Adaptive AI Agent Systems for Personalized Learning: Frameworks, Algorithms, and Practical Applications in Education. In Proceedings of the 2025 5th International Conference on Artificial Intelligence and Education (ICAIE), Suzhou, China, 14–16 May 2025; pp. 108–112.
29. Matia Kangoni, S.; Tshimanga Tshipata, O.; Sedi Nzakuna, P.; Paciello, V.; Mbula Mboma, J.-G.; Makulo, J.-R.; Kyamakya, K. Enhancing Sentiment-Driven Recommender Systems With LLM-Based Feature Engineering: A Case Study in Drug Review Analysis. *IEEE Access* **2025**, *13*, 130304–130322. [CrossRef]
30. Cui, Y.; Wang, K.; Yu, H.; Guo, X.; Cao, H. KLLMs4Rec: Knowledge Graph-Enhanced LLMs Sentiment Extraction for Personalized Recommendations. *Expert Syst. Appl.* **2025**, *282*, 127430. [CrossRef]
31. Woźniak, S.; Koptyra, B.; Janz, A.; Kazienko, P.; Kocoń, J. Personalized Large Language Models. In Proceedings of the 2024 IEEE International Conference on Data Mining Workshops (ICDMW), Abu Dhabi, United Arab Emirates, 9 December 2024; pp. 511–520.
32. Dietterich, T.G. Ensemble Learning. In *The Handbook of Brain Theory and Neural Networks*; Arbib, M.A., Ed.; MIT Press: Cambridge, MA, USA, 2002; pp. 110–125.
33. Wolpert, D.H. Stacked Generalization. *Neural Netw.* **1992**, *5*, 241–259. [CrossRef]
34. Breiman, L. Bagging Predictors. *Mach. Learn.* **1996**, *24*, 123–140. [CrossRef]
35. Freund, Y.; Schapire, R.E. A decision-theoretic generalization of on-line learning and an application to boosting. *J. Comput. Syst. Sci.* **1997**, *55*, 119–139. [CrossRef]

36. Tiwari, D.; Nagpal, B. Ensemble methods of sentiment analysis: A survey. In Proceedings of the 7th International Conference on Computing for Sustainable Global Development (INDIACom 2020), New Delhi, India, 12–14 March 2020; pp. 150–155. [CrossRef]
37. Kyritsis, K.; Liapis, C.M.; Perikos, I.; Paraskevas, M.; Kapoulas, V. From Transformers to Voting Ensembles for Interpretable Sentiment Classification: A Comprehensive Comparison. *Computers* **2025**, *14*, 167. [CrossRef]
38. Pawlikowski, M.; Chorowska, A. Weighted ensemble of statistical models. *Int. J. Forecast.* **2020**, *36*, 93–97. [CrossRef]
39. Saleh, H.; Mostafa, S.; Gabralla, L.A.; Aseeri, A.O.; El-Sappagh, S. Enhanced Arabic Sentiment Analysis Using a Novel Stacking Ensemble of Hybrid and Deep Learning Models. *Appl. Sci.* **2022**, *12*, 8967. [CrossRef]
40. Ting, K.M.; Witten, I.H. Issues in stacked generalization. *J. Artif. Intell. Res.* **1999**, *10*, 271–289. [CrossRef]
41. Jia, J.; Liang, W.; Liang, Y. A review of hybrid and ensemble in deep learning for natural language processing. *arXiv* **2024**, arXiv:2312.05589.
42. Monti, D.; Palumbo, E.; Rizzo, G.; Lisena, P.; Troncy, R.; Fell, M.; Cabrio, E.; Morisio, M. An ensemble approach of recurrent neural networks using pre-trained embeddings for playlist completion. In Proceedings of the ACM Recommender Systems Challenge 2018 (RecSys Challenge '18), Vancouver, BC, Canada, 2–7 October 2018; pp. 1–6. [CrossRef]
43. Çano, E.; Morisio, M. Hybrid recommender systems: A systematic literature review. *Intell. Data Anal.* **2017**, *21*, 1487–1524. [CrossRef]
44. Akhtar, M.S.; Ghosal, D.; Ekbal, A.; Bhattacharyya, P.; Kurohashi, S. All-in-one: Emotion, sentiment and intensity prediction using a multi-task ensemble framework. *IEEE Trans. Affect. Comput.* **2022**, *13*, 285–297. [CrossRef]
45. Mienye, I.D.; Swart, T.G. Ensemble Large Language Models: A Survey. *Information* **2025**, *16*, 688. [CrossRef]
46. Bassan, S.; Amir, G.; Zehavi, M.; Katz, G. What makes an ensemble (un) interpretable? *arXiv* **2025**, arXiv:2506.08216. [CrossRef]
47. Shazeer, N.; Mirhoseini, A.; Maziarz, K.; Davis, A.; Le, Q.; Hinton, G.; Dean, J. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv* **2017**, arXiv:1701.06538.
48. Fedus, W.; Zoph, B.; Shazeer, N. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *J. Mach. Learn. Res.* **2022**, *23*, 1–39.
49. Yuan, Y.; Ma, L.; Talati, N. MoE-Lens: Towards the hardware limit of high-throughput MoE LLM serving under resource constraints. *arXiv* **2025**, arXiv:2504.09345.
50. Yang, Y.; Ma, Y.; Feng, H.; Cheng, Y.; Han, Z. Minimizing Hallucinations and Communication Costs: Adversarial Debate and Voting Mechanisms in LLM-Based Multi-Agents. *Appl. Sci.* **2025**, *15*, 3676. [CrossRef]
51. Kumar, A.; Kim, H.; Nathani, J.S.; Roy, N. Improving the reliability of LLMs: Combining CoT, RAG, self-consistency, and self-verification. *arXiv* **2025**, arXiv:2505.09031.
52. Szymanski, A.; Ziems, N.; Eicher-Miller, H.A.; Li, T.J.-J.; Jiang, M.; Metoyer, R.A. Limitations of the LLM-as-a-judge approach for evaluating LLM outputs in expert knowledge tasks. In Proceedings of the 30th International Conference on Intelligent User Interfaces (IUI 2025), Sydney, Australia, 27–30 January 2025; pp. 952–966. [CrossRef]
53. Roumeliotis, K.I.; Sapkota, R.; Karkee, M.; Tselikas, N.D. Agentic AI with orchestrator-agent trust: A modular visual classification framework with trust-aware orchestration and RAG-based reasoning. *IEEE Access* **2026**. [CrossRef]
54. Chen, L.; Zaharia, M.; Zou, J. FrugalGPT: How to use large language models while reducing cost and improving performance. *arXiv* **2023**, arXiv:2305.05176. [CrossRef]
55. Varangot-Reille, C.; Bouvard, C.; Gourru, A.; Ciancone, M.; Schaeffer, M.; Jacquenet, F. Doing more with less: A survey on routing strategies for resource optimisation in large language model-based systems. *arXiv* **2025**, arXiv:2502.00409.
56. Microsoft. *AutoGen: Stable Documentation*; Microsoft: Redmond, WA, USA, 2026. Available online: https://microsoft.github.io/autogen/stable//index.html (accessed on 18 January 2026).
57. Hong, S.; Zhuge, M.; Chen, J.; Zheng, X.; Cheng, Y.; Zhang, C.; Wang, J.; Wang, Z.; Yau, S.K.S.; Lin, Z.; et al. MetaGPT: Meta programming for a multi-agent collaborative framework. *arXiv* **2024**, arXiv:2308.00352.
58. Yang, F.; Liu, X.C.; Lu, L.; Wang, B.; Liu, C.D. Independent Mobility GPT (IDM-GPT): A self-supervised multi-agent large language model framework for customized traffic mobility analysis using machine learning models. *arXiv* **2025**, arXiv:2502.18652.
59. Zhang, Z.; Wang, J.; Li, Z.; Wang, Y.; Zheng, J. AnnCoder: A Mti-Agent-Based Code Generation and Optimization Model. *Symmetry* **2025**, *17*, 1087. [CrossRef]
60. Applied–AI–Research–Lab. Agentic AI vs. AI Agents in Recommender Systems: A Comparative Study on Sentiment Analysis with Large Language Models. Available online: https://github.com/Applied-AI-Research-Lab/AgenticRecommender-Agentic-AI-vs-AI-Agents-in-Recommender-Systems-Sentiment-Analysis-with-LLMs (accessed on 11 October 2025).
61. PayPal, Inc. OpenAI and PayPal Team up to Power Instant Checkout and Agentic Commerce in ChatGPT. Press Release. 2025. Available online: https://newsroom.paypal-corp.com/2025-10-28-OpenAI-and-PayPal-Team-Up-to-Power-Instant-Checkout-and-Agentic-Commerce-in-ChatGPT (accessed on 11 October 2025).
62. OpenAI. Models—OpenAI API Documentation. Online Developer Documentation. 2025. Available online: https://platform.openai.com/docs/models (accessed on 11 October 2025).

63. Anthropic PBC. Models Overview—Claude API Documentation. Online Developer Documentation. 2025. Available online: https://platform.claude.com/docs/en/about-claude/models/overview (accessed on 11 October 2025).
64. Google LLC. Gemini Models—Gemini API Documentation. Online Developer Documentation. 2025. Available online: https://ai.google.dev/gemini-api/docs/models (accessed on 11 October 2025).
65. Hangzhou DeepSeek Artificial Intelligence Basic Technology Research Co., Ltd. Models and Pricing—DeepSeek API Documentation. Online Developer Documentation. 2025. Available online: https://api-docs.deepseek.com (accessed on 11 October 2025).
66. Takahashi, K.; Yamamoto, K.; Kuchiba, A.; Koyama, T. Confidence Interval for Micro-Averaged F1 and Macro-Averaged F1 Scores. *Appl. Intell.* **2022**, *52*, 4961–4972. [CrossRef]
67. Margaris, D.; Spiliotopoulos, D.; Sgardelis, K.; Vassilakis, C. Using Prediction Confidence Factors to Enhance Collaborative Filtering Recommendation Quality. *Technologies* **2025**, *13*, 181. [CrossRef]
68. Landis, J.R.; Koch, G.G. The Measurement of Observer Agreement for Categorical Data. *Biometrics* **1977**, *33*, 159–174. [CrossRef]