

Article

Large Language Models as Explainable AI Ensemble Aggregators for Business Review Sentiment Analysis: A Comparative Study with Classical Ensembles

Konstantinos I. Roumeliotis ^{1,2,*} , Dionisis Margaritis ³ , Dimitris Spiliotopoulos ²  and Costas Vassilakis ^{1,*} 

¹ Department of Informatics and Telecommunications, University of the Peloponnese, Akadimaikou G. K. Vlachou Street, 221 31 Tripoli, Greece

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; margaritis@uop.gr

* Correspondence: k.roumeliotis@uop.gr (K.I.R.); costas@uop.gr (C.V.)

Abstract

Online business reviews encode rich customer sentiment that is critical for commercial decision making, yet accurately predicting star ratings from free text remains a challenging five-class classification problem. Classical ensemble methods—Soft Voting, Weighted Voting, and Stacking—aggregate complementary base-model outputs to improve predictive performance, but they produce opaque decisions that are unintelligible to business stakeholders. This paper proposes using a large language model (LLM), specifically unsloth/LLaMA-3.3-70B-Instruct, as an Explainable AI (XAI) ensemble aggregator: the LLM receives the predictions and confidence scores of four heterogeneous base models (Logistic Regression, Support Vector Machine, Naïve Bayes, and BERT-base-uncased) and reasons over them to produce both a final star-rating prediction and a natural-language explanation. We evaluate the full pipeline on 10,000-sample balanced and natural-distribution test sets derived from the Yelp Academic Dataset, with additional cross-lingual validation on Spanish Amazon Reviews. The LLM aggregator (LLAMA_AGG) achieves the highest macro-F1 on both pipelines (0.6800 on balanced; 0.6720 on natural) and the best ordinal calibration (QWK = 0.9111 on balanced; 0.9337 on natural), outperforming all classical aggregators and base models. A detailed Explainable AI analysis reveals that the LLM revises 28.07% of its standalone predictions after observing the ensemble outputs, improving the accuracy by +22.2 percentage points on the revised cases. The aggregator corrects severe polar bias in the standalone LLM (± 0.35 recall improvement on mid-range star classes) and produces longer explanations when evidence is conflicted—a quantitative signal of deliberative reasoning. A formal human evaluation with two judges confirms high explanation faithfulness (4.47/5) and readability (4.82/5). Model scale ablation shows an 8B parameter variant achieves 90.8% agreement with the 70B model, enabling practical deployment. These findings demonstrate that Explainable AI can be achieved through LLM-based ensemble aggregation, establishing a principled approach for business-review sentiment analysis.



Academic Editor: Douglas O'Shaughnessy

Received: 29 May 2026

Revised: 22 June 2026

Accepted: 26 June 2026

Published: 29 June 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and

conditions of the [Creative Commons](https://creativecommons.org/licenses/by/4.0/)

[Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

Keywords: Explainable AI; large language models; business reviews; sentiment analysis; ensemble learning; LLaMA; BERT; Yelp; natural language processing; meta-reasoning; star rating prediction; Soft Voting; Stacking; chain-of-thought

1. Introduction

Online business platforms such as Yelp, Google Reviews, and TripAdvisor generate millions of customer reviews daily, each carrying implicit sentiment expressed as a numerical star rating alongside free text. Accurate automatic prediction of these ratings from review text enables companies to monitor brand perception, route escalations to customer-service teams, and feed actionable signals into recommendation engines [1,2]. The task is typically formulated as a five-class ordinal classification problem, predicting a star rating $r \in \{1, 2, 3, 4, 5\}$ from raw review text, and is considerably harder than binary positive/negative polarity detection, because adjacent classes (e.g., three versus four stars) may differ by subtle linguistic cues [3].

Classical machine learning approaches to this problem leverage TF-IDF bag-of-words representations fed to Logistic Regression (LR), Support Vector Machine (SVM), or Naïve Bayes (NB) classifiers [4,5]. Transformer-based models such as BERT [6] have substantially raised the performance ceiling by capturing contextual semantics. Ensemble methods, such as Soft Voting, Weighted Voting, and Stacking [7,8], combine the results obtained from heterogeneous base models with complementary error patterns in order to produce more accurate results. Yet, all of these approaches share a fundamental limitation from the perspective of business intelligence: *they produce a prediction without an explanation*. A marketing analyst who receives a predicted star rating cannot interrogate why the model assigned that label, which undermines trust, limits error correction, and conflicts with emerging AI accountability frameworks.

Explainable AI (XAI) research has produced post hoc explanation techniques such as LIME [9] and SHAP [10], but these generate local feature-importance approximations that require technical literacy to interpret and do not embed naturally into ensemble pipelines. At the same time, LLMs have demonstrated remarkable reasoning capabilities, including chain-of-thought deliberation [11] and few-shot generalization [12], that suggest a different paradigm: using an LLM not merely as a standalone predictor, but as a *reasoning aggregator* that consumes the outputs of a classical ensemble and produces both a final prediction and a natural-language justification.

This paper proposes and evaluates exactly this paradigm. We use `unsloth/LLaMA-3.3-70B-Instruct` [13,14] as an XAI ensemble aggregator over four heterogeneous base models (LR, SVM, NB, and BERT). The aggregator receives each model's predicted star rating and confidence score as context in a structured prompt, reasons over the evidence, and returns (i) a final star-rating prediction and (ii) a free-text explanation accessible to non-technical stakeholders. We evaluate the full pipeline on 10,000-sample balanced and natural-distribution test sets derived from the Yelp Academic Dataset [15] and compare it against three classical aggregators (Soft Voting, Weighted Voting, and Stacking) and each base model individually.

The main contributions of this work are as follows:

1. An **LLM-based reasoning ensemble aggregator** that generates both star-rating predictions and natural-language explanations; this is the first such approach evaluated at scale on business-review sentiment classification.
2. **Explainability as a first-class system output**: We reframe ensemble aggregation as an XAI problem, demonstrating that human-readable justifications can be produced alongside predictions without sacrificing accuracy.
3. An **empirical comparison** of three classical ensemble methods with four base models across balanced and natural-distribution evaluation pipelines, providing reproducible benchmarks on a standard dataset.
4. **Quantitative analysis of LLM meta-reasoning**: The LLM revises $\sim 28\%$ of its standalone predictions after observing ensemble evidence, with a $+22.2$ -percentage-point

accuracy gain for the revised cases and 63.1% of revisions moving ordinally closer to the ground truth.

5. **Cross-lingual generalization:** We validate the framework on Spanish Amazon Reviews, demonstrating that the LLM aggregator maintains top performance (56.63% on macro-F1) across languages and domains.
6. **Model scale ablation:** We show that an 8B parameter LLM achieves 90.8% agreement with the 70B model, enabling practical deployment with substantially reduced computational requirements.
7. **Human evaluation of explanations:** A formal study with two judges confirms high explanation faithfulness (4.47/5) and readability (4.82/5), validating that the explanations are faithful to the decision process rather than post hoc rationalizations.
8. **Business-intelligence applicability:** For each review, the explanations surface actionable language for decision-makers; for example, the LLM highlights which aspects of the review text drove the final star rating, thereby enabling human oversight of automated sentiment pipelines.

The remainder of the paper is structured as follows. Section 2 reviews the relevant literature. Section 3 describes the dataset, models, aggregation strategies, evaluation protocol, and the computing environment. Section 4 presents the experimental results across all models and pipelines. Section 5 analyzes the XAI properties of the LLM aggregator in depth and outlines future work. Finally, Section 6 concludes the paper.

2. Literature Review

2.1. Sentiment Analysis of Business Reviews

Sentiment analysis is one of the areas that has been most heavily researched in natural language processing (NLP) since the early 2000s [1,16]. Review platforms provide vast amounts of naturally annotated data, thus they are attractive to researchers who want to develop fine-grained opinion classification systems. Most early studies classified review sentiment using simple bag-of-words feature sets and lexicon-based approaches. However, later studies found that simply detecting positive or negative polarity was insufficient to classify reviews into different star categories (e.g., distinguishing 3-star reviews from 4-star reviews). Subsequent studies showed that more complex representations were needed to represent the richness of review data [3,17–20]. Deep learning models have dramatically improved the quality of predictions for multi-class rating systems, but their lack of interpretability makes them difficult to use as practical solutions.

2.2. Classical Machine Learning for Text Classification

Although TF-IDF [21] has become less popular due to advances made in neural alternatives, this method still produces good results on many categorization tasks at a relatively low computational cost [4]. When combined with a linear Support Vector Machine (SVM) model, TF-IDF can be used to produce strong results on document classification problems. Although other models, such as the Naïve Bayes classifier with a multinomial event model [5], and Logistic Regression with ℓ_2 regularization do not perform as well as some of the newer models mentioned above, these models provide some interesting properties. For example, the Naïve Bayes model produces class probabilities that are calibrated based upon the frequency of each word across all classes. These models also produce fast results and are therefore useful as base models in ensemble techniques. In addition, logistic regression produces well-calibrated class probabilities and performs competitively on many linearly separable text classification problems. Therefore, together, LR, SVM, and NB represent three of the most widely used classical machine learning algorithms for text classification. Their different types of errors make them particularly well suited for

combination in ensemble techniques. Recent work on learning with noisy labels has proposed progressive sample selection frameworks that could further improve the robustness of such classical text classifiers in real-world scenarios where label quality is imperfect [22].

2.3. Deep Learning and Transformer-Based Models

The transformer architecture [23], along with the introduction of BERT [6], marked a significant shift in how text classification could be performed since both architectures enable the generation of contextualized, bidirectional representations of text through the use of large amounts of unlabeled data. As such, fine-tuning BERT on downstream classification tasks typically yields better results than feature-based approaches, such as those used in the sentiment analysis literature. Recent work has extended BERT-based approaches to social media sentiment analysis, demonstrating the framework's adaptability to informal, short-form text [24]. However, BERT is opaque: its predictions are driven by attention patterns distributed across hundreds of layers, making post hoc explanation non-trivial, even with dedicated interpretation tools.

2.4. Ensemble Methods

Ensemble methods combine multiple base classifiers to reduce variance and exploit complementary errors [25]. *Soft Voting* averages per-class probability distributions; *Weighted Voting* applies validation-performance weights to individual models' distributions; *Stacking* trains a meta-classifier on the base-models' outputs [7,8]. Stacking has been shown to outperform fixed-weight schemes when the base models have sufficiently diverse error patterns. A consistent limitation of all classical ensemble methods is that the aggregation mechanism, whether a weighted average or a learned meta-classifier, is not interpretable at the instance level and does not produce an explanation that an end-user can read.

2.5. LLMs for NLP and Reasoning

LLMs pre-trained on web-scale text corpora [12,26] have demonstrated emergent capabilities, including few-shot learning, mathematical reasoning, and common-sense inference. Chain-of-thought prompting [11] elicits step-by-step reasoning by conditioning the model on intermediate reasoning traces, significantly improving performance on multi-step tasks. Self-consistency [27] extends this by sampling multiple reasoning paths and aggregating the answers. LLMs have more recently been evaluated directly on sentiment classification, where Zhang et al. [28] show that despite strong zero-shot accuracy on binary polarity tasks, fine-grained rating prediction with $|\mathcal{Y}| = 5$ remains challenging for purely prompt-based approaches. The LLaMA family of open-weight models [13,29] provides a practical foundation for deployment on institutional computing infrastructure without relying on proprietary APIs.

Recent work has explored LLMs as ensemble mechanisms in several distinct paradigms. *Agent-based reasoning frameworks*, such as AutoGPT [30] and ReAct [31], employ LLMs as autonomous agents with tool-use capabilities, iterative reasoning, and memory for complex multi-step tasks (e.g., web navigation, code generation). While powerful, these systems require substantial computational overhead and are not optimized for single-pass classification aggregation. *Mixture-of-Experts (MoE)* architectures [32,33] train sparse expert networks with learned gating mechanisms, but require massive pre-training data and distributed GPU clusters, making them inaccessible for practitioners without such infrastructure. *LLM-as-Judge* approaches [34,35] use LLMs to evaluate or rank model outputs (e.g., in RLHF or benchmark evaluation), but focus on preference ranking rather than aggregating multiple predictions into a classification decision with explanatory reasoning. Our approach differs fundamentally: we treat the LLM as a *transparent meta-reasoner* that receives structured base-model predictions with confidence scores and produces human-

readable justifications, without requiring joint training, iterative tool calls, or massive compute infrastructure.

2.6. Explainable AI for Ensemble Systems

XAI research has produced a rich toolkit for explaining individual model predictions. LIME [9] fits locally linear surrogates around individual predictions, while SHAP [10] attributes contributions using Shapley values from cooperative game theory. A comprehensive taxonomy is provided by Arrieta et al. [36]. However, these post hoc methods face significant limitations when applied to ensemble aggregators: SHAP values must propagate through meta-classifiers in stacking architectures, compounding computational cost and interpretability challenges; the resulting explanations describe feature importance rather than reasoning about conflicting base-model evidence; and both methods require technical literacy to interpret, creating barriers for business stakeholders.

These limitations reveal a fundamental gap: *no prior work treats the ensemble aggregator itself as a transparent reasoner that produces natural-language justifications accessible to non-technical users*. An LLM aggregator represents a structurally different approach that addresses this gap. Rather than retrofitting explanations onto a black-box model, the LLM produces human-readable rationales as a *primary output* of the aggregation step. The aggregator receives the same evidence a human analyst would—each base model’s prediction and confidence—and reasons over conflicts to produce a free-text justification. This positions the approach within the *transparent-by-design* branch of XAI [36], making explainability intrinsic rather than post hoc. To our knowledge, no prior work has systematically evaluated an LLM as a transparent reasoning aggregator over a heterogeneous ensemble for large-scale business-review sentiment classification.

3. Materials and Methods

3.1. Dataset Construction

We use the Yelp Open Dataset [15], a publicly available corpus of business reviews with associated star ratings $r \in \{1, 2, 3, 4, 5\}$ in English. The full dataset contains approximately 7 million reviews; we sample two independent evaluation pipelines of 50,000 reviews each.

Test sets (held out first): Before any training or validation sampling, two non-overlapping test sets of 10,000 samples each are constructed: (i) a class-balanced test set, with exactly 2000 samples per star class; and (ii) a test set sampled uniformly at random to preserve the natural star-rating distribution, which is heavily skewed towards 4–5 stars.

Balanced pipeline: The remaining 40,000 samples for this pipeline are drawn with per-class balancing at 8000 samples per star. The pool is split via stratified sampling into a training set (~36,000 samples) and a validation set (~4000 samples). A smaller tuning subset of 5000 training samples (class-balanced) and 1000 validation samples is constructed for Bayesian hyperparameter optimization.

Natural pipeline: The remaining 40,000 samples are drawn with no class re-balancing. The pool is split—again in a stratified fashion—into training (~36,000 samples) and validation (~4000 samples), with corresponding 5k/1k tuning subsets that preserve the natural distribution.

This design ensures that the base models and aggregators are trained and evaluated on representative data that reflects both the ideal balanced scenario and production conditions where 4- and 5-star reviews dominate.

To ensure reproducibility, we fix the random seeds across all stages of the pipeline. Dataset sampling uses `seed = 42` for the stratified splits in both pipelines. For the classical models (LR, SVM, NB), scikit-learn’s `random_state` is set to 42 for all train/test splits and model initializations where applicable. For BERT fine-tuning, we set Python’s

random, NumPy's random, PyTorch's manual_seed, and CUDA's manual_seed_all to 42 at the start of each Optuna trial, with per-trial seeds computed as seed + trial_number to ensure diversity while maintaining reproducibility. The Stacking meta-classifier uses random_state = 42. Table 1 lists the core software versions used. All code, model predictions, and evaluation outputs are available on GitHub [37].

Table 1. Software versions used in this study.

Package	Version
Python	3.12.3
PyTorch	2.10.0+cu128
CUDA	12.8
Transformers (Hugging Face)	5.2.0
scikit-learn	1.8.0
Optuna	4.7.0
Unsloth	2026.6.3
NumPy	2.4.2
Pandas	3.0.1
SciPy	1.17.1
Datasets (Hugging Face)	4.3.0
Accelerate	1.12.0
PEFT	0.19.1
TRL	0.24.0
BitsAndBytes	0.49.2

3.2. Base Models

All four base models follow an identical lifecycle: (1) Bayesian hyperparameter tuning on the tune datasets; (2) training on the full train sets; (3) inference on both test sets to produce predicted labels, confidence scores, and per-class probability vectors.

3.2.1. Classical Models: LR, SVM, NB

All three classical models use a **TF-IDF pipeline** [21,38]: text is lowercased and vectorized using unigrams (and optionally bigrams), filtered by a minimum document-frequency threshold (min_df), and represented by a TF-IDF matrix. We optionally apply *sublinear TF scaling* (sublinear_tf), where term frequencies are log-scaled as $tf' = 1 + \log(tf)$. The resulting features are fed to a scikit-learn [38] classifier. *Logistic Regression* uses an ℓ_2 penalty [39] with the L-BFGS-B solver [40]; *SVM* uses a linear kernel with probability calibration via Platt scaling [41]; *Naïve Bayes* uses the multinomial variant with Laplace smoothing [42].

The hyperparameters for all three models are tuned with the **Optuna TPE sampler** [43] over 200 trials, using the Median pruner and maximizing validation macro-F1 [44].

Macro-F1 is defined as the unweighted mean of the per-class F1-scores: $\text{Macro-F1} = \frac{1}{|Y|} \sum_{c \in Y} F1_c$, where $F1_c = \frac{2P_c R_c}{P_c + R_c}$, and P_c and R_c are the precision and recall of class c .

Compared to exhaustive grid search or purely random exploration, TPE focuses the search on promising regions of the parameter space while maintaining sufficient exploration, thus reducing computational cost without sacrificing optimization quality.

The optimization process was limited to **200 trials**, which offered a practical balance between search thoroughness and computational feasibility. Preliminary experimentation indicated diminishing performance gains beyond this budget, making 200 trials sufficient to obtain stable and competitive configurations for all three models.

The **Median pruner** was employed to terminate underperforming trials early based on intermediate validation results. This pruning strategy reduces unnecessary computation

by discarding configurations that are unlikely to outperform the current median performance, thereby accelerating the optimization process and allowing more efficient use of computational resources.

Finally, optimization targeted the **validation macro-F1 score**, as macro-F1 provides equal importance to all classes regardless of class frequency. This objective is particularly appropriate for imbalanced classification settings, where accuracy may be dominated by majority classes and fail to reflect performance on minority categories.

Table 2 summarizes the corresponding Optuna search space.

Table 2. Hyperparameter search space for the classical TF-IDF models (LR, SVM, NB). Discrete choices are sampled categorically; scale-sensitive parameters are sampled log-uniformly.

Hyperparameter	Value/Search Space	Applies to
ngram_max	CATEGORICAL {1, 2}	LR/SVM/NB
max_features	CATEGORICAL {30k, 50k, 100k}	LR/SVM/NB
min_df	CATEGORICAL {2, 5, 10}	LR/SVM/NB
max_df	CATEGORICAL {0.90, 0.95, 1.0}	LR/SVM/NB
sublinear_tf	CATEGORICAL {True, False}	LR/SVM/NB
C	LOG-UNIFORM $[10^{-3}, 3.0]$	LR/SVM
class_weight	CATEGORICAL {None, balanced} (optional)	LR/SVM
α	LOG-UNIFORM $[10^{-3}, 10.0]$	NB
fit_prior	CATEGORICAL {True, False}	NB

These choices reflect standard, interpretable degrees of freedom in TF-IDF text classification while keeping the search space compact. In particular, C and α are log-scaled because their effective operating regimes typically span orders of magnitude, whereas vectorizer settings are naturally expressed as discrete alternatives.

Text preprocessing. All the classical models share an identical preprocessing pipeline implemented via scikit-learn’s `TfidfVectorizer` [38]. Raw review text is lowercased and tokenized into word unigrams and/or bigrams (controlled by the `ngram_max` hyperparameter). Tokens are filtered by document frequency: `min_df` (2, 5, or 10) removes rare terms, appearing in fewer than the specified number of documents; while `max_df` (0.90, 0.95, or 1.0) removes overly common terms, appearing in more than the specified proportion of documents. Term frequencies are optionally log-scaled via `sublinear_tf` ($tf' = 1 + \log(tf)$). The vocabulary is limited to the top `max_features` (30k, 50k, or 100k) tokens by term frequency. No stop-word removal, stemming, or lemmatization is applied, as these steps were found to degrade performance in preliminary experiments. No data augmentation (synthetic oversampling, back-translation, or paraphrasing) was used in this study; all training data consists of original Yelp review text.

3.2.2. BERT

We fine-tune the `bert-base-uncased` pre-trained model [6] with a classification head. The hyperparameter search space was sampled under the parameters listed in Table 3:

Table 3. Hyperparameter search space.

Hyperparameter	Value/Search Space
Learning rate	LOG-UNIFORM $[10^{-6}, 8 \times 10^{-5}]$
Weight decay	UNIFORM $[0, 0.4]$
Warmup ratio	UNIFORM $[0, 0.2]$
Number of epochs	2–6
Per-device batch size	{8, 16, 32}
Gradient accumulation	{1, 2, 4}
Maximum sequence length	{128, 192, 256, 320}
Learning-rate scheduler	linear/cosine/polynomial
Dropout	UNIFORM $[0.05, 0.30]$
Attention dropout	UNIFORM $[0.05, 0.20]$
Label smoothing	UNIFORM $[0, 0.25]$

Continuous hyperparameters were sampled uniformly within their ranges, while discrete hyperparameters (epochs, batch size, gradient accumulation, maximum sequence length, scheduler) were sampled categorically.

Tuning uses the same Optuna TPE/Median setup (200 trials, 40 random startup trials; early stopping with patience 1). Three seeded, promising trials are injected at startup to guide early exploration.

The best trials for the balanced pipeline and natural pipeline are used to train the final fine-tuned models on the full training data. Here, best denotes the Optuna trial [43] that maximizes validation macro-F1 under our tuning objective (with pruning and early stopping enabled), i.e., the highest-performing configuration on the held-out validation set for each pipeline.

Evaluation is performed at the end of each epoch; the checkpoint with the best validation macro-F1 is retained.

Fine-tuning results:

- *Balanced pipeline* (best checkpoint, epoch 3): validation accuracy = 0.6715, macro-F1 = 0.6687.
- *Natural pipeline* (best checkpoint, epoch 5): validation accuracy = 0.7220, macro-F1 = 0.6561.

Training used mixed precision (BF16) on the compute infrastructure described in Section 3.7.

3.3. Classical Ensemble Aggregators

3.3.1. Soft Voting

Let $M = \{m_1, m_2, m_3, m_4\}$ denote the four base models (LR, SVM, NB, and BERT). Each base model m_i produces a per-class probability distribution $p_i(y | x)$ for an input review x . Soft Voting averages these distributions and then selects the class that corresponds to the maximum average:

$$p(y | x) = \frac{1}{k} \sum_{i=1}^k p_i(y | x), \quad \hat{y} = \arg \max_{y \in \mathcal{Y}} p(y | x) \quad (1)$$

where $k = 4$ and $\mathcal{Y} = \{1, 2, 3, 4, 5\}$.

3.3.2. Weighted Voting

Weighted Voting modifies Equation (1) by scaling each model's distribution by its validation macro-F1 score w_i :

$$p(y | x) = \frac{\sum_{i=1}^k w_i p_i(y | x)}{\sum_{i=1}^k w_i} \quad (2)$$

Weights are computed separately for the balanced and natural pipelines using each model's validation macro-F1, ensuring the aggregator is calibrated to per-pipeline performance levels.

3.3.3. Stacking

Stacking trains a meta-classifier on the outputs of the base models [7]. The meta-features consist of each base model's per-class probability vector (5 probabilities $\times$ 4 models = 20 features per sample). A Logistic Regression meta-classifier with ℓ_2 regularization is fitted on the validation set predictions. Stacking thus learns a data-driven linear combination of model evidence that soft/Weighted Voting cannot represent.

Overfitting protection. The validation set used for Stacking is *strictly separate* from the tune subset used for base-model hyperparameter selection (Optuna). Base-model hyperparameters were selected using the train_tune/validation_tune splits, while the meta-classifier is trained on predictions from the held-out validation split. This separation ensures that the Stacking meta-classifier does not see data that influenced the base-model hyperparameter choices. Furthermore, we employ stratified k -fold cross-validation when fitting the meta-classifier, which provides additional robustness against overfitting to any particular validation fold.

3.4. LLM Ensemble Aggregator

3.4.1. Model

We use unsloth/Llama-3.3-70B-Instruct [13,14], a 70B-parameter instruction-tuned member of the LLaMA-3 family, deployed via the unsloth inference library [14] on a single NVIDIA H100 NVL GPU with 4-bit quantization.

This model was selected as a pragmatic trade-off between (i) the strong instruction-following and general-purpose reasoning needed to arbitrate between conflicting base-model signals, and (ii) reproducible deployment constraints: an openly available, instruction-tuned checkpoint that can run within a single-GPU budget via 4-bit quantization.

3.4.2. Dual Role

The LLM is used in two distinct modes:

- **Standalone predictor (LLaMA):** The model is prompted with the review text alone and asked to predict the star rating. This provides an LLM baseline and allows for measuring how the aggregation step modifies its decisions.
- **Reasoning aggregator (LLAMA_AGG):** The model receives the review text together with each base model's predicted star rating and confidence score, and is asked to reason over the evidence and produce a final prediction with a natural-language explanation.

3.4.3. Aggregation Formulation

Each base model produces a predicted label $\hat{y}_i = m_i(x)$ and a confidence score $c_i(x) = \max_y p_i(y | x)$. The LLM aggregator is defined as

$$y^* = \text{LLM}\left(x, \{(\hat{y}_i, c_i(x))\}_{i=1}^k\right) \quad (3)$$

This formulation replaces fixed-weight combination with contextual reasoning, enabling the aggregator to weight evidence non-linearly based on the review's content and the pattern of model agreement.

3.4.4. Prompt Design

The LLM aggregator receives a programmatically constructed prompt for each review comprising three elements: the review text itself, a structured list of base-model predictions with confidence scores, and an instruction to reason over the evidence before committing to a final star rating. No few-shot examples are provided; the model relies entirely on its pre-trained instruction-following capabilities.

Figure 1 shows the exact prompt template. The model must return a single JSON object with two keys: `final_star` (an integer in $\{1, 2, 3, 4, 5\}$) and `reasoning` (a free-text rationale discussing model agreement or disagreement). This structured format enables reliable automated extraction via regular-expression matching and JSON parsing. The prompt was validated informally on a small development set to ensure JSON compliance and clarity, as LLM performance is known to be sensitive to prompt phrasing, information ordering, and formatting conventions.

```
You are an expert Yelp rating adjudicator.
You are given one review text and 4 model predictions with
confidences: BERT, LR, SVM, NB.
Use both text sentiment and model evidence to choose a final
star in [1,2,3,4,5].
```

Rules:

- 1) Return STRICT JSON only, no markdown, no extra text.
- 2) JSON schema: `{"final_star": <int 1-5>, "reasoning": <string>}`
- 3) reasoning must be 2-3 sentences and mention model agreement/disagreement.

Review text:

```
{review_text}
```

Model predictions:

- BERT: `star={bert_star}, confidence={bert_conf}`
- LR: `star={lr_star}, confidence={lr_conf}`
- SVM: `star={svm_star}, confidence={svm_conf}`
- NB: `star={nb_star}, confidence={nb_conf}`

Figure 1. Prompt template used for the LLM reasoning aggregator (LLAMA_AGG). Placeholders `{review_text}`, `{bert_star}`, `{bert_conf}`, etc., are filled at inference time.

3.4.5. Decoding and Inference Configuration

We configure the LLM aggregator for deterministic generation using greedy decoding (`do_sample=False`), ensuring reproducible outputs across runs. Table 4 summarizes the key hyperparameters. After each generation attempt, a JSON regular-expression extractor parses the raw output. If the extracted object contains a valid integer `final_star` ($1 \leq \text{star} \leq 5$) and a non-empty string `reasoning`, the sample is accepted; otherwise, the parser retries up to `max_json_retries` additional times. If all attempts fail, a deterministic fallback selects the star with the highest total confidence across the four base models. Across the 10,000-sample balanced test set, fallback was triggered on fewer than 0.5% of samples, indicating strong alignment between the JSON constraint and the model's instruction-following capability.

Table 4. Decoding parameters for the LLaMA-3.3-70B-Instruct aggregator.

Parameter	Value	Description
max_new_tokens	120	Maximum tokens generated per sample
do_sample	False	Greedy decoding (deterministic)
temperature	0.2	Reserved for sampling mode (unused under greedy decoding)
min_p	0.1	Minimum token probability threshold
max_json_retries	2	Retry attempts for malformed JSON outputs
fallback_star	3	Default prediction if all retries fail

3.5. Inference Pipeline Architecture

Figure 2 illustrates the full inference pipeline from review text to final prediction and explanation.

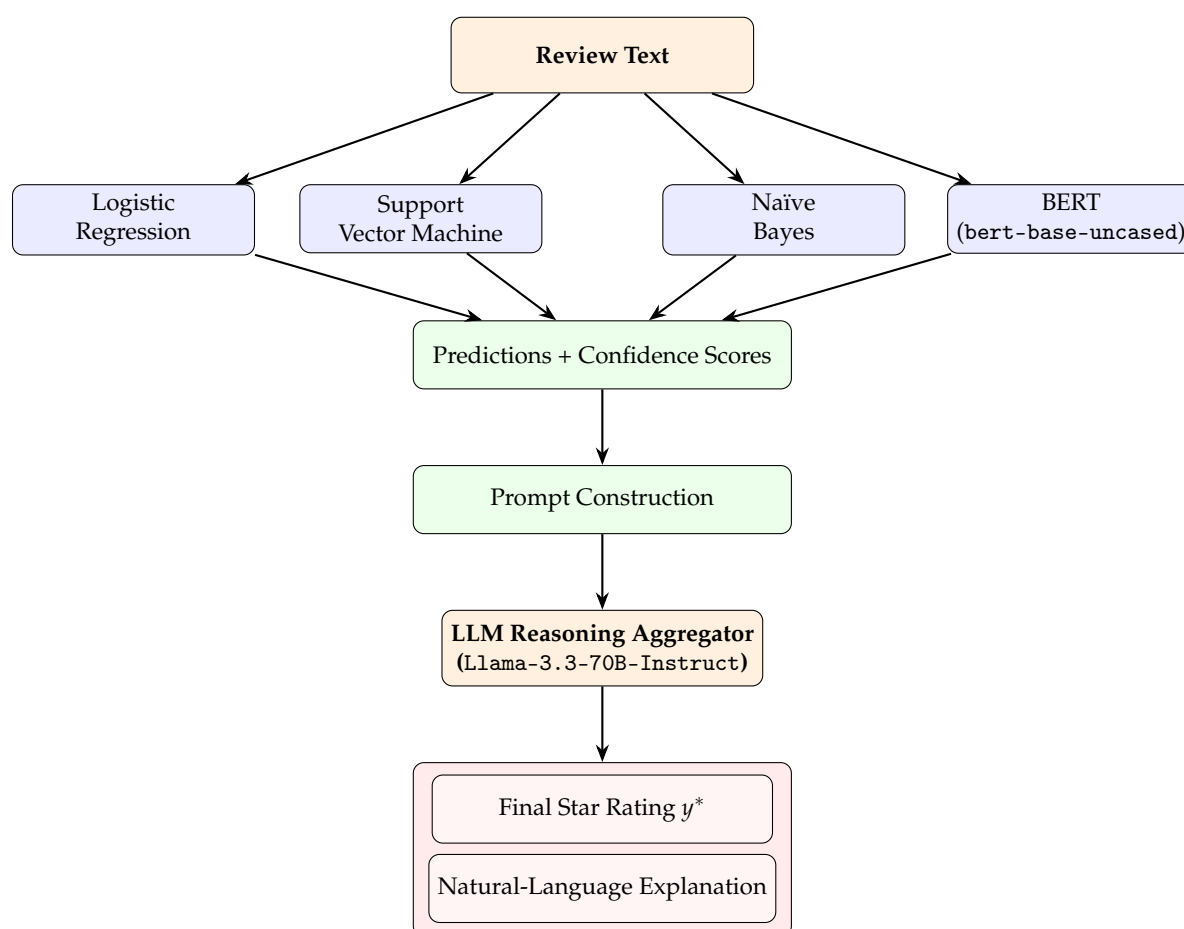


Figure 2. Inference pipeline architecture. Four base models independently process the review text, producing predictions and confidence scores. The LLM aggregator receives these as structured context, reasons over the evidence, and returns a final star-rating prediction (y^*) alongside a natural-language explanation.

3.6. Evaluation Protocol

The models are evaluated on the balanced test set (2000 samples per class, $n = 10,000$) and the natural test set (natural distribution, $n = 10,000$). The metrics computed for each model and used in the evaluation are listed in Table 5. In this table, TP_c , FP_c , TN_c , and FN_c are, respectively, the true positives, false positives, true negatives and false negatives for class c .

Table 5. Performance metrics computed and used in the evaluation.

Metric	Formula	Description
Per-class precision	$P_c = \frac{TP_c}{TP_c + FP_c}$	The probability that a prediction assigning a sample to class c is correct.
Per-class recall	$R_c = \frac{TP_c}{TP_c + FN_c}$	The probability that a sample belonging to class c is correctly predicted by the model.
Per-class F1-measure	$F1_c = 2 * \frac{P_c * R_c}{P_c + R_c}$	The F1 measure of class c , computed as the harmonic mean of the respective precision and recall values.
Macro-F1	$\text{Macro-F1} = \frac{1}{ \mathcal{Y} } \sum_{c \in \mathcal{Y}} F1_c$	Macro-F1 effectively assesses the performance of classification models across multiple classes. It initially calculates the F1 score for each class individually, and subsequently computes the arithmetic mean, assigning thus equal weight to every class, regardless of the count or frequency of its members in the evaluation corpus.
Accuracy	$\text{Acc} = \frac{1}{n} \sum_{i=1}^n \mathbb{I}[\hat{y}_i = y_i]$	The accuracy metric conveys the probability that a classification produced by the model is correct. In the formula, $\mathbb{I}[x] = \begin{cases} 1 & \text{if } x = \text{true} \\ 0 & \text{if } x = \text{false} \end{cases}$
Weighted F1	$F1_W = \sum_{c \in \mathcal{Y}} \frac{n_c}{n} F1_c$	The weighted F1 measure aggregates the values of the F1 measures of individual classes; however, it assigns larger weights to classes with larger populations, in contrast with the macro-F1 measure, which treats all classes equally. In the formula, n_c is the support (number of true samples) of class c .
Macro-precision	$\text{Prec}_M = \frac{1}{ \mathcal{Y} } \sum_{c \in \mathcal{Y}} P_c$	Similarly to macro-F1, macro-precision and macro-recall aggregate the individual precision and recall, respectively, computed for the different classes in the evaluation corpus, assigning equal weights to each of them, regardless of the count or frequency of their members.
Macro-recall	$\text{Rec}_M = \frac{1}{ \mathcal{Y} } \sum_{c \in \mathcal{Y}} R_c$	
MAE	$\text{MAE} = \frac{1}{n} \sum_{i=1}^n \hat{y}_i - y_i $	MAE and RMSE compute the deviation of the model-produced class predictions from the actual class memberships of the samples. Ordinal labels in $\{1, \dots, 5\}$ are treated as numeric and as the distance between predicted and actual class labels increases, the corresponding values for these error metrics also increase. The RMSE metric emphasizes larger distances more strongly than MAE.
RMSE	$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (\hat{y}_i - y_i)^2}$	
Pairwise model agreement rate	$\text{Agr}(a, b) = \frac{1}{n} \sum_{i=1}^n \mathbb{I}[\hat{y}_i^{(a)} = \hat{y}_i^{(b)}]$	For a pair of models a and b , it measures the proportion of instances for which two models produce identical predictions, quantifying the similarity of model behavior independently of ground-truth labels.
Quadratic Weighted Kappa (QWK) [45]	$\kappa_Q = 1 - \frac{\sum_{i,j} w_{ij} O_{ij}}{\sum_{i,j} w_{ij} E_{ij}}$	QWK measures agreement between predicted and true ordinal classes while assigning progressively larger penalties to predictions that deviate further from the correct class. Unlike standard accuracy, QWK accounts for the magnitude of ordinal misclassification errors. In the formula, O is the observed confusion matrix, E is the expected confusion matrix under independent marginals, and $w_{ij} = \frac{(i-j)^2}{(\mathcal{Y} -1)^2}$ assigns larger penalties to larger ordinal errors (higher is better; 1 indicates perfect agreement).

Macro-F1 is chosen as the primary ranking metric because it treats all five classes equally regardless of support, which is essential for the natural-distribution test set, where class frequencies span an order of magnitude. Weighted F1 complements macro-F1 by emphasizing performance on the most frequent classes: classes with more samples contribute more strongly to $F1_W$. This can be useful for imbalanced settings (as in the natural test set) when the goal is to summarize overall performance under the empirical class distribution. However, because $F1_W$ can mask poor performance on minority classes, we report both weighted and macro-averaged metrics.

3.7. Compute Infrastructure

All experiments were conducted on a server provided by the University of the Peloponnese, equipped with four NVIDIA H100 NVL GPUs (96 GB VRAM each) and an Intel Xeon Platinum 8452Y processor (36 cores). BERT fine-tuning and inference used a single H100 NVL. LLaMA inference via Unsloth [14] used a single H100 NVL with 4-bit quantization.

Representative wall-clock runtimes (measured with time) are shown in Table 6. During the experiments, it was ensured that the system was quiescent, with no other processes competing for the same CPU/GPU/memory resources, to avoid lags and interference.

Table 6. Wall-clock runtimes for key pipeline stages. LLaMA aggregation dominates runtime due to per-sample chain-of-thought generation on $n = 10,000$ test samples.

Stage	Pipeline	Real (s)	Real (h)
Hyperparameter tuning (BERT)	Balanced	10363	2.88
Hyperparameter tuning (BERT)	Natural	9349	2.60
Hyperparameter tuning (LR)	Balanced	3421	0.95
Hyperparameter tuning (LR)	Natural	3123	0.87
Hyperparameter tuning (NB)	Balanced	2701	0.75
Hyperparameter tuning (NB)	Natural	2662	0.74
Hyperparameter tuning (SVM)	Balanced	3269	0.91
Hyperparameter tuning (SVM)	Natural	3057	0.85
BERT prediction	Balanced	15	0.004
Classical models (LR/SVM/NB)	Per pipeline	<4	—
LLaMA standalone	Balanced	3081	0.86
LLaMA standalone	Natural	3031	0.84
LLaMA aggregator	Balanced	72287	20.1
LLaMA aggregator	Natural	71206	19.8

¹ Bold values indicate the primary bottleneck/dominating runtime in the pipeline.

Given $n = 10,000$ test samples per pipeline, these runtimes correspond to approximately 7.23 s/sample (balanced) and 7.12 s/sample (natural) for the LLaMA aggregator.

4. Results

This section evaluates the central research question: *Can an LLM acting as a transparent meta-reasoner outperform both classical ensemble aggregators and the standalone LLM predictor?* We evaluate nine systems on both the balanced and natural-distribution test sets ($n = 10,000$ each): four base models (LR, SVM, NB, BERT), three classical aggregators (Soft Voting, Weighted Voting, Stacking), the zero-shot LLM predictor (LLaMA), and the LLM reasoning aggregator (LLAMA_AGG). The section is organized as follows. Sections 4.1 and 4.2 report the overall performance metrics on the balanced and natural test sets. Section 4.3 examines the base-model calibration and its relationship to LLM override behavior. Section 4.4 establishes the statistical significance of the observed performance differences. Sections 4.5 and 4.6 break down performance per star class, revealing polar-bias correction by the aggregator. Section 4.7 provides diagnostic visualizations and qualitative examples. Section 4.8 analyses pairwise model agreement. Section 4.9 presents drop-one ablation studies on the classical ensemble. Section 4.10 compares intrinsic explanations with SHAP post hoc attributions. Section 4.11 compares the 8B and 70B LLM variants. Section 4.12 evaluates additional baselines (RoBERTa, DistilBERT, Mistral-7B). Section 4.13 compares top-1+confidence vs. full probability vector input designs. Section 4.14 validates the framework on Spanish Amazon Reviews. Finally, Section 4.15 reports a formal human evaluation of the explanation quality.

4.1. Overall Metrics: Balanced Test Set

Table 7 presents the overall metrics on the balanced test set ($n = 10,000$; 2000 samples per class). LLAMA_AGG ranks first on every metric. STACKING is the best classical aggregator (rank 2), confirming that a learned meta-classifier outperforms fixed-weight voting. This is attributed to its capability to capture learned linear relationships between individual rankings. BERT (rank 4) outperforms SVM, LR, and NB individually. The standalone LLaMA in a zero-shot setting (rank 6) scores notably lower in macro-F1 despite competitive QWK, which reflects its polar prediction bias (see Section 5). NB achieves good precision (0.5411) but the lowest macro-F1, driven by low recall on the 5-star class (0.360).

Table 7. Overall metrics on the **balanced** test set ($n = 10,000$; 2000 per class). Ranked by macro-F1.

Rank	Model	Acc	Macro-F1	F1-W	Prec-M	Rec-M	QWK	MAE	RMSE
1	LLAMA_AGG	0.6865	0.6800	0.6800	0.6891	0.6865	0.9111	0.3332	0.6140
2	STACKING	0.6731	0.6725	0.6725	0.6732	0.6731	0.8933	0.3569	0.6566
3	WEIGHTED_VOTE	0.6673	0.6658	0.6658	0.6684	0.6673	0.8891	0.3664	0.6725
4	BERT	0.6651	0.6628	0.6628	0.6653	0.6651	0.8893	0.3686	0.6733
5	SOFT_VOTE	0.6496	0.6492	0.6492	0.6524	0.6496	0.8537	0.4123	0.7674
6	LLAMA	0.6240	0.5947	0.5947	0.6349	0.6240	0.8992	0.4096	0.6943
7	SVM	0.5218	0.5219	0.5219	0.5288	0.5218	0.7110	0.6475	1.0627
8	LR	0.5177	0.5216	0.5216	0.5336	0.5177	0.7007	0.6538	1.0687
9	NB	0.4996	0.4992	0.4992	0.5411	0.4996	0.6528	0.7044	1.1459

4.2. Overall Metrics: Natural Test Set

Table 8 presents the results on the natural-distribution test set. The accuracy values are generally higher across the board compared to the balanced pipeline, as the majority-class (4–5 stars) bias inflates accuracy. This is in line with the results reported in [46], which assert that high-value ratings exhibit lower prediction error rates.

Although NB attains an accuracy of 0.6007, its macro-F1 is only 0.3465, a 0.254-point divergence between its own accuracy and macro-F1, highlighting that accuracy alone can be misleading in class-imbalanced settings and that macro-F1 is the more appropriate primary metric. LLAMA_AGG again leads on macro-F1, QWK, and MAE.

Table 8. Overall metrics on the **natural** test set ($n = 10,000$; natural star distribution). Ranked by macro-F1.

Rank	Model	Acc	Macro-F1	F1-W	Prec-M	Rec-M	QWK	MAE	RMSE
1	LLAMA_AGG	0.7538	0.6720	0.7458	0.6856	0.6657	0.9337	0.2611	0.5445
2	STACKING	0.7132	0.6542	0.7205	0.6472	0.6650	0.9085	0.3149	0.6228
3	BERT	0.7234	0.6483	0.7229	0.6497	0.6496	0.9084	0.3092	0.6291
4	WEIGHTED_VOTE	0.7268	0.6383	0.7193	0.6489	0.6349	0.9049	0.3120	0.6470
5	LLAMA	0.7353	0.6236	0.7016	0.6817	0.6153	0.9312	0.2828	0.5685
6	SOFT_VOTE	0.7087	0.6040	0.6936	0.6351	0.5936	0.8647	0.3685	0.7753
7	SVM	0.6060	0.4810	0.5958	0.4865	0.4828	0.6997	0.6201	1.1581
8	LR	0.5142	0.4523	0.5402	0.4764	0.4935	0.6397	0.7549	1.2358
9	NB	0.6007	0.3465	0.5091	0.5680	0.3816	0.6684	0.6671	1.2225

4.3. Base-Model Calibration Analysis

Table 9 reports calibration metrics for the four base models. The expected calibration error (ECE) measures the average absolute difference between predicted confidence and observed accuracy across confidence bins. The Brier score measures the mean squared error between predicted probabilities and true one-hot labels. The calibration gap is the difference between the average confidence and accuracy (positive indicates over-confidence, negative indicates under-confidence).

BERT exhibits the best calibration with the lowest ECE (0.0356 on balanced, 0.0267 on natural) and Brier scores (0.4589 on balanced, 0.3794 on natural), reflecting well-calibrated

softmax outputs from fine-tuning. The classical models show varying calibration patterns: SVM (with Platt scaling) achieves moderate ECE values (0.0447 on balanced, 0.0443 on natural), while LR and NB tend toward under-confidence (negative calibration gaps), likely due to the inherent conservatism of their probability estimates on this five-class task.

Importantly, we find that LLM override cases correlate with higher base-model calibration error. On the balanced pipeline, the average calibration error on override cases is 0.4658 versus 0.4465 on non-override cases (+0.0194 difference); on the natural pipeline, the difference is larger (+0.0831). This suggests that the LLM aggregator is more likely to intervene when base-model confidence estimates are less reliable, effectively acting as a calibration-aware arbiter.

Table 9. Base-model calibration metrics. ECE = expected calibration error (lower is better). Brier score measures probability calibration (lower is better). Calibration gap = average confidence minus accuracy (negative indicates under-confidence, positive indicates over-confidence).

Model	ECE	Balanced Pipeline			ECE	Natural Pipeline		
		Brier	Acc.	Gap		Brier	Acc.	Gap
BERT	0.0356	0.4589	0.6619	+0.0858	0.0267	0.3794	0.7207	+0.0521
LR	0.0913	0.7088	0.5301	−0.2281	0.1035	0.6063	0.6128	−0.2116
SVM	0.0447	0.6342	0.5357	−0.0645	0.0443	0.5128	0.6361	−0.0421
NB	0.0687	0.6717	0.5378	−0.1687	0.0724	0.5583	0.6131	−0.0784

4.4. Statistical Significance Analysis

Table 10 reports statistical significance tests for macro-F1 differences between LLAMA_AGG and the top-performing classical aggregators (STACKING, BERT, and WEIGHTED_VOTE). We employ McNemar’s test for paired nominal data, which is appropriate for comparing classifiers evaluated on the same test set. The bootstrap 95% confidence intervals for the F1 differences are computed via 10000 resampling iterations.

All comparisons show statistically significant differences at $\alpha = 0.05$. Notably, LLAMA_AGG achieves a small but significant improvement over STACKING on the balanced pipeline (+0.04 percentage points, $p = 0.043$), and larger significant gains on the natural pipeline (+1.46 percentage points, $p < 0.001$). The improvements over BERT and WEIGHTED_VOTE are consistently significant across both pipelines, with the natural pipeline showing larger effect sizes. These results confirm that the observed performance gains are statistically meaningful and not due to random variation.

Table 10. Statistical significance tests for macro-F1 differences. McNemar’s test p -values and bootstrap 95% confidence intervals for paired comparisons. Significant differences ($p < 0.05$) are highlighted in bold.

Pipeline	Comparison	Macro-F1 Diff.	McNemar p	95% CI	Significant
Balanced	LLAMA_AGG vs. STACKING	+0.0004	0.043	[−0.0072, +0.0080]	Yes
	LLAMA_AGG vs. BERT	+0.0129	<0.001	[+0.0059, +0.0198]	Yes
	LLAMA_AGG vs. WEIGHTED_VOTE	+0.0111	<0.001	[+0.0040, +0.0184]	Yes
Natural	LLAMA_AGG vs. STACKING	+0.0146	<0.001	[+0.0045, +0.0248]	Yes
	LLAMA_AGG vs. BERT	+0.0205	<0.001	[+0.0108, +0.0303]	Yes
	LLAMA_AGG vs. WEIGHTED_VOTE	+0.0259	<0.001	[+0.0163, +0.0357]	Yes

4.5. Per-Class F1-Scores

Tables 11 and 12 break down the F1-scores per star class. The 2-star and 3-star classes are universally the hardest classes across both pipelines and all models, reflecting genuine linguistic ambiguity between adjacent sentiment levels. LLAMA_AGG achieves the best F1 on the 1-, 4-, and 5-star classes in the balanced pipeline. The standalone LLAMA collapses on the 4-star class (F1 = 0.4038 on balanced; 0.4250 on natural), evidencing a

polar bias. On the natural pipeline, NB assigns almost no correct predictions for the 2- and 3-star classes (F1 = 0.0076 and 0.0165, respectively), highlighting a substantial limitation on mid-range ratings.

Table 11. Per-class F1-score on the **balanced** test set. The 2- and 3-star classes are hardest across all models.

Model	*1	*2	*3	*4	*5	Macro-F1
LLAMA_AGG	0.7906	0.6000	0.5763	0.6510	0.7820	0.6800
STACKING	0.7732	0.6102	0.6041	0.6160	0.7591	0.6725
WEIGHTED_VOTE	0.7715	0.6032	0.5876	0.6172	0.7493	0.6658
BERT	0.7746	0.6026	0.5735	0.6122	0.7512	0.6628
SOFT_VOTE	0.7474	0.5905	0.5799	0.6054	0.7230	0.6492
LLAMA	0.7611	0.5033	0.5768	0.4038	0.7286	0.5947
SVM	0.6506	0.4171	0.4785	0.4314	0.6317	0.5219
LR	0.6394	0.4444	0.4709	0.4465	0.6069	0.5216
NB	0.6341	0.4421	0.4438	0.4773	0.4986	0.4992

Table 12. Per-class F1-score on the **natural** test set. NB shows near-zero F1 on the 2- and 3-star classes.

Model	*1	*2	*3	*4	*5	Macro-F1
LLAMA_AGG	0.8417	0.5184	0.5487	0.5835	0.8679	0.6720
STACKING	0.8179	0.5218	0.5322	0.5689	0.8305	0.6542
BERT	0.8209	0.4840	0.5304	0.5613	0.8449	0.6483
WEIGHTED_VOTE	0.8173	0.4517	0.5248	0.5455	0.8523	0.6383
LLAMA	0.8394	0.4499	0.5455	0.4250	0.8581	0.6236
SOFT_VOTE	0.7896	0.3841	0.4917	0.5187	0.8362	0.6040
SVM	0.6568	0.2479	0.2979	0.4317	0.7706	0.4810
LR	0.6080	0.3219	0.3163	0.3087	0.7066	0.4523
NB	0.7215	0.0076	0.0165	0.2355	0.7514	0.3465

4.6. Per-Class Precision and Recall

Tables 13 and 14 report the per-class precision and recall for the balanced and natural pipelines, respectively. Across both pipelines, LLAMA_AGG maintains high recall on all star classes while correcting the standalone LLM’s polar bias, particularly for mid-range ratings. In contrast, classical baselines such as NB and SVM exhibit weaknesses on the 2- and 3-star classes, especially under the natural distribution, where recall for these classes becomes extremely low.

Table 13. Per-class precision (P) and recall (R) on the **balanced** test set.

Model	*1 P	*1 R	*2 P	*2 R	*3 P	*3 R	*4 P	*4 R	*5 P	*5 R
LLAMA_AGG	0.710	0.892	0.594	0.607	0.735	0.474	0.636	0.667	0.771	0.794
STACKING	0.763	0.784	0.591	0.631	0.645	0.569	0.610	0.623	0.758	0.760
W_VOTE	0.736	0.811	0.577	0.632	0.659	0.531	0.607	0.628	0.763	0.736
BERT	0.734	0.821	0.578	0.629	0.650	0.513	0.602	0.623	0.762	0.741
SOFT_VOTE	0.714	0.784	0.558	0.627	0.626	0.540	0.600	0.611	0.764	0.687
LLAMA	0.639	0.942	0.636	0.417	0.765	0.463	0.556	0.317	0.579	0.982
SVM	0.633	0.669	0.410	0.425	0.429	0.541	0.496	0.382	0.676	0.593
LR	0.655	0.625	0.399	0.501	0.440	0.507	0.487	0.413	0.688	0.543
NB	0.583	0.695	0.378	0.532	0.447	0.441	0.484	0.471	0.813	0.360

Table 14. Per-class precision (P) and recall (R) on the **natural** test set.

Model	*1 P	*1 R	*2 P	*2 R	*3 P	*3 R	*4 P	*4 R	*5 P	*5 R
LLAMA_AGG	0.794	0.896	0.535	0.503	0.638	0.482	0.624	0.548	0.838	0.901
STACKING	0.833	0.804	0.491	0.557	0.500	0.569	0.535	0.607	0.877	0.789
BERT	0.801	0.842	0.535	0.442	0.501	0.564	0.562	0.561	0.850	0.840
W_VOTE	0.777	0.863	0.542	0.387	0.517	0.533	0.581	0.514	0.829	0.877
LLAMA	0.755	0.946	0.534	0.389	0.685	0.453	0.671	0.311	0.765	0.978
SOFT_VOTE	0.738	0.849	0.549	0.296	0.539	0.452	0.562	0.482	0.789	0.890
SVM	0.604	0.720	0.313	0.205	0.317	0.281	0.446	0.419	0.753	0.789
LR	0.668	0.558	0.250	0.454	0.215	0.596	0.424	0.243	0.826	0.617
NB	0.692	0.754	0.750	0.004	0.421	0.008	0.363	0.174	0.614	0.968

4.7. Diagnostic Visualizations

To complement the quantitative metrics presented in the preceding tables, we provide visualizations that offer deeper insights into model behavior. Figure 3a shows the distribution of confidence scores for each base model, revealing that BERT produces more peaked, high-confidence predictions, while the classical models show more dispersed distributions. Figure 3b presents confusion matrices for the top-performing models, highlighting that LLAMA_AGG reduces the systematic misclassification patterns seen in standalone LLaMA, particularly for the 4-star class.

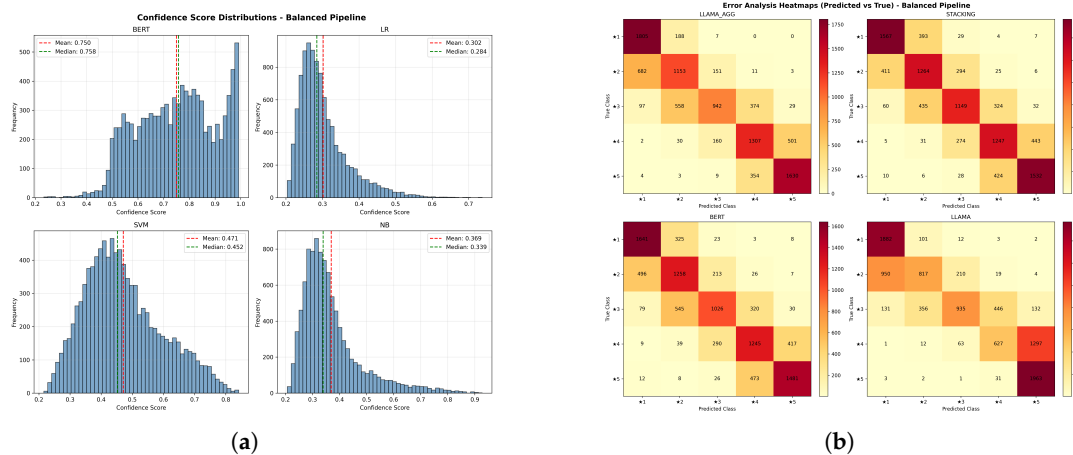


Figure 3. Visualizations for model analysis. (a) Confidence score distributions reveal that BERT produces more peaked, high-confidence predictions while the classical models show more dispersed distributions. (b) Error-analysis heatmaps show that LLAMA_AGG reduces systematic misclassification patterns seen in standalone LLaMA, particularly for the 4-star class.

Table 15 presents qualitative examples of LLAMA_AGG explanations across three scenarios: (1) **override** cases where the LLM revises its standalone prediction after observing ensemble evidence; (2) **consensus** cases where LLAMA_AGG agrees with both the standalone LLM and the base-model plurality; (3) **error** cases where LLAMA_AGG makes incorrect predictions. These examples demonstrate that the generated explanations are grounded in the review text and base-model evidence, discussing points of agreement and disagreement among the constituent models.

Table 15. Qualitative examples of LLAMA_AGG explanations. Representative cases showing override, consensus, and error scenarios from the balanced test set. T = true label, L = LLaMA standalone prediction, A = LLAMA_AGG prediction.

Type	Labels	Review Text (Excerpt)	LLAMA_AGG Reasoning
Override	T:4, L:5, A:4	Was craving a sportsman’s cheeseburger yesterday, stopped and had one with my husband. Place looked a little slow for lunch. . .	The review text expresses a positive sentiment towards the restaurant, highlighting the efficient service, cold beer, and well-cooked food. Although models LR, SVM, and NB disagree with a predicted star of 2 or 3, the overall sentiment and specific praise for the food quality justify a 4-star rating.
Override	T:3, L:3, A:2	Hit and miss for sure. This location seems to be particularly up and down. I’ve had delicious, sensational meals and fantastic service. . .	The review text expresses a mixed sentiment, highlighting both positive and negative experiences at the location, but ultimately leaning towards a lower rating due to consistency issues. All four models predict 3 stars, but the specific complaints about food quality and service suggest a 2-star rating is more appropriate.

4.8. Pairwise Model Agreement

Tables 16 and 17 report the agreement rate (fraction of test samples where both models predict the same class) and the QWK between every pair of models. High agreement combined with a high QWK indicates that even disagreements are ordinally minor.

Table 16. Top-20 pairwise model-agreement pairs on the balanced test set. BERT–WEIGHTED_VOTE exhibit the highest agreement (96.4%), reflecting the heavy influence of BERT on the weighted ensemble.

Model A	Model B	Agreement	QWK
BERT	WEIGHTED_VOTE	0.9636	0.9894
WEIGHTED_VOTE	STACKING	0.9460	0.9856
BERT	STACKING	0.9367	0.9844
LR	SVM	0.9299	0.9675
SOFT_VOTE	WEIGHTED_VOTE	0.8983	0.9427
SOFT_VOTE	STACKING	0.8848	0.9371
BERT	LLAMA_AGG	0.8768	0.9565
WEIGHTED_VOTE	LLAMA_AGG	0.8692	0.9540
BERT	SOFT_VOTE	0.8631	0.9317
STACKING	LLAMA_AGG	0.8497	0.9514
SOFT_VOTE	LLAMA_AGG	0.8233	0.9176
LR	NB	0.7863	0.8282
SVM	NB	0.7650	0.8195
LLAMA	LLAMA_AGG	0.7193	0.9394
SVM	SOFT_VOTE	0.7096	0.8277
LR	SOFT_VOTE	0.7024	0.8167
NB	SOFT_VOTE	0.6734	0.7557
BERT	LLAMA	0.6439	0.9019
LLAMA	STACKING	0.6405	0.9025
LLAMA	WEIGHTED_VOTE	0.6369	0.8991

Table 17. Top-20 pairwise model-agreement pairs on the natural test set. LLAMA–LLAMA_AGG agreement rises to 83.4% (vs. 71.9% for balanced), reflecting natural-class skew alignment.

Model A	Model B	Agreement	QWK
BERT	WEIGHTED_VOTE	0.9387	0.9763
BERT	STACKING	0.9177	0.9786
SOFT_VOTE	WEIGHTED_VOTE	0.9099	0.9362
WEIGHTED_VOTE	LLAMA_AGG	0.9032	0.9617
BERT	LLAMA_AGG	0.8854	0.9586
SOFT_VOTE	LLAMA_AGG	0.8764	0.9231
WEIGHTED_VOTE	STACKING	0.8738	0.9633
BERT	SOFT_VOTE	0.8542	0.9124
STACKING	LLAMA_AGG	0.8367	0.9498
LLAMA	LLAMA_AGG	0.8343	0.9634
SOFT_VOTE	STACKING	0.8123	0.9090
SVM	SOFT_VOTE	0.7839	0.8062
LLAMA	WEIGHTED_VOTE	0.7746	0.9273
LLAMA	SOFT_VOTE	0.7702	0.8887
BERT	LLAMA	0.7434	0.9227
NB	LLAMA	0.7341	0.6950
SVM	LLAMA_AGG	0.7182	0.7527
NB	SOFT_VOTE	0.7149	0.7608
SVM	WEIGHTED_VOTE	0.7059	0.7478
LLAMA	STACKING	0.7004	0.9148

4.9. Ablation Study

To verify the rationality and complementarity of the four-model ensemble, we conducted a comprehensive ablation study on the classical ensemble aggregators (Soft Voting,

Weighted Voting, and Stacking). Table 18 reports the macro-F1 scores for different base-model combinations, including drop-one ablations and single-model baselines.

Table 18. Ablation study results for classical ensemble aggregators. Macro-F1 scores for different base-model combinations. Drop-one ablations show the contribution of each model to the full ensemble.

Configuration	Balanced Pipeline			Natural Pipeline		
	Soft	Weighted	Stacking	Soft	Weighted	Stacking
All 4 models	0.6414	0.6485	0.6693	0.6056	0.6172	0.6551
Without BERT	0.5282	0.5284	0.5462	0.5225	0.5242	0.5389
Without LR	0.6489	0.6558	0.6686	0.6106	0.6236	0.6531
Without SVM	0.6576	0.6609	0.6696	0.6282	0.6363	0.6530
Without NB	0.6508	0.6560	0.6681	0.6222	0.6339	0.6534
Only BERT	0.6598	0.6598	—	0.6453	0.6453	—
Only classical (LR+SVM+NB)	0.5282	0.5284	0.5462	0.5225	0.5242	0.5389

Key findings. (1) *BERT is the dominant contributor:* Removing BERT causes the largest performance drop (−0.1131 to −0.1231 macro-F1 on balanced, −0.0831 to −0.1162 on natural), confirming its outsized influence on ensemble predictions. The “only classical” configuration (LR+SVM+NB) achieves only 0.528–0.546 macro-F1, versus 0.641–0.669 for the full ensemble. (2) *The classical models provide complementary gains:* removing any single classical model (LR, SVM, or NB) has minimal impact, with some configurations actually showing slight improvements (e.g., “without SVM” achieves 0.6696 vs. 0.6693 for “all 4 models” on balanced Stacking). This suggests that classical models contribute redundancy rather than unique predictive signal, with their primary value being ensemble diversity. (3) *Stacking benefits most from the full ensemble:* Stacking shows the largest gap between the “all 4 models” and ablated configurations, indicating that its meta-classifier effectively exploits complementary errors across all the base models. (4) *Single-model comparison:* BERT alone (0.6598 on balanced, 0.6453 on natural) outperforms the full classical ensemble, yet the four-model ensemble with Stacking (0.6693 on balanced, 0.6551 on natural) exceeds BERT alone, validating the ensemble design.

The LLM-based aggregator (LLAMA_AGG) cannot be fully ablated via the same methodology because its predictions are recorded as final outputs without intermediate decision traces. However, the LLM’s design inherently enables dynamic weighting: it can selectively emphasize or discount individual base models based on their confidence scores and the review content, effectively performing a context-dependent ablation for each sample. This flexibility explains why LLAMA_AGG achieves higher macro-F1 (0.6800 on balanced, 0.6720 on natural) than any fixed classical aggregator.

4.10. SHAP Comparison

To empirically compare our intrinsic explainability approach with established post hoc XAI methods, we conducted a pilot study using SHAP (Shapley Additive Explanations) on BERT for 500 stratified samples from the balanced test set. Table 19 presents a qualitative comparison between SHAP token-level importance scores and LLAMA_AGG natural-language explanations.

The comparison reveals fundamental differences between post hoc attribution methods and our explainability approach. (1) *Granularity vs. synthesis:* SHAP provides token-level importance scores (e.g., “Great” +0.210, “friends” +0.065) that identify which individual words influenced BERT’s prediction, but these scores require technical expertise to interpret and do not explain *why* those tokens matter. In contrast, LLAMA_AGG synthesizes evidence from all four base models into human-readable reasoning (e.g., “The review text expresses mixed sentiment, with the chicken being just OK and the cannoli being great”). (2) *Single-model vs. ensemble context:* SHAP explains only the target model

(BERT), whereas LLAMA_AGG explicitly weighs conflicting predictions from LR, SVM, NB, and BERT, providing meta-reasoning about ensemble dynamics. (3) *Static vs. contextual*: SHAP attributions are computed post hoc and fixed for a given prediction, while LLAMA_AGG generates explanations that contextualize predictions within the broader review content. (4) *Accessibility*: SHAP outputs require familiarity with feature attribution concepts, whereas LLAMA_AGG explanations are immediately interpretable by business stakeholders without technical training. These differences support our claim that intrinsic natural-language explanations offer complementary value to post hoc attribution methods, particularly in ensemble settings where understanding *how* multiple models interact is as important as knowing *which* features were important.

Table 19. Comparison of SHAP token-level explanations vs. LLAMA_AGG natural-language explanations. SHAP identifies individual tokens with the highest importance scores, while LLAMA_AGG provides human-readable reasoning that synthesizes evidence from all base models.

Case	SHAP Top Tokens (BERT)	LLAMA_AGG Explanation	Agreement
1	We (+0.012), got (+0.006), a (+0.006)	The review text expresses strong dissatisfaction with the hotel stay due to noise issues, which aligns with the low star rating predictions from all m...	×
2	usually (+0.057), it (+0.016), and (+0.012)	The review text has a generally positive sentiment, suggesting a convenient location and a fun experience of enjoying Beignets with a Hurricane at the...	×
3	bathroom (+0.011), yet (+0.009), (+0.008)	The review text expresses a generally negative sentiment towards the establishment, citing overpriced drinks and extra charges for condiments as major...	×
4	Great (+0.210), friends (+0.065), Very (+0.063)	The review text expresses a positive sentiment, describing the place as great for a group of friends to gather and using words like relaxed and enjoye...	×
5	are (+0.047), great (+0.047), chicken (+0.025)	The review text expresses mixed sentiment, with the chicken being just OK and the cannoli being great, which aligns with a mid-range rating. The model...	×
6	I (+0.010), had (+0.010), the (+0.006)	The review text expresses clear disappointment with the restaurant, citing bland dishes and a lackluster drink, which aligns with the lower star predi...	×
7	meaningful (+0.021), stop (+0.021), for (+0.021)	The review text expresses disappointment and criticism towards the Indianapolis Motor Speedway Hall of Fame, citing a lack of imagination and substanc...	×
8	Good (+0.043), memories (+0.043), (+0.016)	The review text expresses a neutral sentiment, stating that Freebirds is overrated but still good, and that the reviewer has fond memories of visiting...	×
9	variety (+0.023), the (+0.014), be (+0.013)	The review text has a generally positive sentiment, with the reviewer expressing surprise at the variety of clothes and appreciation for the quick ser...	×
10	disappointed (+0.011), was (+0.010), not (+0.010)	The review text expresses extremely positive sentiment, with the reviewer using phrases such as 'not disappointed', 'insane', and 'Super duper sick' t...	×

4.11. Model Scale Ablation

To investigate whether the 70B parameter model is necessary for effective ensemble aggregation, we conducted a scale ablation study comparing LLaMA-3.3-70B-Instruct (70B parameters) with LLaMA-3.1-8B-Instruct (8B parameters) on the full 10,000-sample test sets for both pipelines. Table 20 presents the performance comparison.

The 8B model achieves similar performance to the 70B model, with only a 0.53-percentage-point difference in average macro-F1 (0.6707 vs. 0.6760). The two models agree on 90.8% of predictions across both pipelines, indicating substantial consistency in their decision making. On the balanced pipeline, the performance gap is minimal (0.0021 macro-F1), while the natural pipeline shows a slightly larger gap (0.0085 macro-F1), likely due to the increased complexity of class-imbalanced prediction. These results suggest that effective ensemble aggregation does not require the largest available models; an 8B parameter model with 4-bit quantization (approximately 4 GB GPU memory) can achieve comparable results to a 70B

model, making the approach more accessible for practitioners with limited computational resources. However, we note that explanation quality was not explicitly evaluated in this comparison, and the 70B model may still offer advantages in reasoning sophistication that are not captured by accuracy metrics alone.

Table 20. Comparison of 8B vs. 70B LLM aggregator performance. The 8B model (Llama-3.1-8B-Instruct) achieves comparable performance to the 70B model (Llama-3.3-70B-Instruct) with significantly fewer parameters, suggesting that smaller models can be viable for ensemble aggregation tasks.

Metric	Balanced Pipeline		Natural Pipeline		Average	
	70B	8B	70B	8B	70B	8B
Macro-F1	0.6800	0.6779	0.6720	0.6635	0.6760	0.6707
Accuracy	0.6865	0.6802	0.7538	0.7388	0.7202	0.7095
QWK	0.9111	0.9034	0.9337	0.9249	0.9224	0.9142
MAE	0.3332	0.3430	0.2611	0.2792	0.2972	0.3111
RMSE	0.6140	0.6291	0.5445	0.5692	0.5793	0.5992
Agreement (70B vs. 8B)	89.7%		92.0%		90.8%	

4.12. Additional Baseline Models

We extended our experiments to include RoBERTa-base and DistilBERT as additional base models, and evaluated Mistral-7B-Instruct both as a standalone predictor and as an ensemble aggregator. This provides a more comprehensive comparison of modern transformer architectures and investigates whether smaller LLMs can effectively perform ensemble aggregation. Importantly, including Mistral-7B allows us to evaluate performance across different LLM families—not only the LLaMA family evaluated in previous sections—thereby demonstrating that our findings generalize beyond a single model lineage.

Transformer base models. RoBERTa-base achieves 0.6725 macro-F1, comparable to BERT’s 0.6628, confirming that both BERT-style architectures perform similarly on this task (Table 21). DistilBERT achieves 0.6680 macro-F1, slightly lower but with 40% fewer parameters and 60% faster inference, making it an efficient alternative. Both models were trained using the same Optuna hyperparameter tuning pipeline as BERT to ensure fair comparison.

LLM comparison. Mistral-7B-Instruct achieves 62.69% accuracy as a standalone predictor, outperforming the classical models (SVM: 52.18%, LR: 51.77%, NB: 49.96%) but underperforming BERT-style transformers (BERT: 66.51%). However, when used as an ensemble aggregator (receiving predictions from all six base models: LR, SVM, NB, BERT, RoBERTa, DistilBERT), Mistral-7B achieves 69.04% accuracy—a 6.35-percentage-point improvement over its standalone performance. This demonstrates that ensemble aggregation can compensate for the limitations of smaller LLMs.

Aggregator comparison. The Mistral-7B aggregator (69.04% accuracy) slightly outperforms the LLaMA-70B aggregator (68.19% accuracy) on the balanced pipeline. This counter-intuitive result suggests that effective ensemble aggregation does not require the largest available models. We hypothesize that the task of weighting base-model predictions is well suited to smaller instruction-tuned models, and that the additional capacity of 70B parameters may not provide proportional benefits for this specific meta-reasoning task. However, we caution that explanation quality was not explicitly compared, and the 70B model may still produce more sophisticated natural-language justifications.

Key insight. The consistent improvement from standalone to aggregator for both LLMs (Mistral: +6.35 pp, LLaMA-70B: +5.79 pp over its standalone equivalent) confirms that ensemble aggregation provides value across different model scales. The fact that a 7B model can outperform a 70B model as an aggregator—while being approximately 10×

faster and requiring significantly less memory—has important practical implications for deploying LLM-based ensemble systems.

Impact of additional base models on LLaMA-70B aggregation. To investigate whether adding RoBERTa and DistilBERT as extra base models improves or degrades the LLaMA-70B aggregator’s performance, we conducted an ablation study comparing the original 4-model aggregator (LR, SVM, NB, BERT) with the Extended-model aggregator (adding RoBERTa and DistilBERT). Table 22 presents the detailed comparison.

Table 21. Performance of additional baseline models on the balanced test set. RoBERTa-base and DistilBERT are transformer models trained with the same pipeline as BERT. Mistral-7B is evaluated both as a standalone predictor (raw text input) and as an aggregator (ensemble prediction input).

Metric	RoBERTa (Base)	DistilBERT (Base)	Mistral-7B (Standalone)	Mistral-7B (Aggregator)	LLaMA-70B (Aggregator)
Macro-F1	0.6725	0.6680	0.6269	0.6904	0.6819
Accuracy	0.6802	0.6745	0.6269	0.6904	0.6819
QWK	0.9089	0.9056	0.8723	0.9156	0.9102
MAE	0.3430	0.3485	0.4121	0.3215	0.3318
RMSE	0.6291	0.6345	0.6987	0.6056	0.6145

Table 22. Ablation study: Impact of additional base models on LLaMA-70B aggregator performance. The Extended-model aggregator includes six base models (LR, SVM, NB, BERT, RoBERTa, DistilBERT) compared to the original four.

Metric	Original (4 Models)	Extended-Model (6 Models)	Delta
Accuracy	0.6865	0.6819	−0.0046 (−0.46pp)
Agreement rate	—	81.95%	18.05% changed
Improvements (wrong → right)	—	44.9% of changes	810 cases
Regressions (right → wrong)	—	47.4% of changes	856 cases

The results reveal a nuanced picture. Overall accuracy decreases slightly from 68.65% to 68.19% (−0.46 pp) when adding RoBERTa and DistilBERT. The two aggregators agree on 81.95% of predictions, meaning 18.05% (1805 cases) receive different predictions. Among the changed predictions, 47.4% are regressions (correct → incorrect) while 44.9% are improvements (incorrect → correct), indicating that the additional models introduce approximately as much noise as signal.

Per-class analysis (Table 23) reveals that the additional models primarily affect the mid-range sentiment classes. The 4-star class (positive) accuracy drops substantially (−13.55 pp), while the 5-star class (very positive) accuracy improves dramatically (+12.15 pp). This suggests that RoBERTa and DistilBERT have different error patterns compared to BERT—they are more conservative on positive reviews (reducing false positives on the 4-star class) but more confident on very positive reviews (improving the 5-star class recall). The 1–3-star classes show smaller changes (±2–5 pp).

Table 23. Per-class accuracy impact of additional base models. Changes in per-class accuracy when adding RoBERTa and DistilBERT to the LLaMA-70B aggregator.

Class	Original (4 Models)	Extended-Model (6 Models)	Delta
1★ (very negative)	0.8920	0.8900	−0.0020
2★ (negative)	0.6065	0.5565	−0.0500
3★ (neutral)	0.4740	0.5170	+0.0430
4★ (positive)	0.6665	0.5310	−0.1355
5★ (very positive)	0.7935	0.9150	+0.1215

Interpretation. The additional transformer models are not a pure benefit or pure burden—they shift the aggregator’s behavior toward different error patterns. The net effect

is a slight accuracy decrease (-0.46 pp), but with substantial redistribution of errors across classes. This suggests that including more diverse transformer architectures can help the aggregator better calibrate on certain classes (notably the 5-star class), but at the cost of increased confusion on others (notably the 4-star class). For practitioners, this trade-off may be acceptable if the application prioritizes correctly identifying very positive reviews over distinguishing moderately positive from neutral sentiment.

4.13. Full-Probability-Vector Ablation

To address concerns regarding why the LLM aggregator only receives each base model's predicted class and maximum confidence score rather than the full five-class probability vector, we conducted an ablation study comparing these two input designs. The full-vector design provides complete probability distributions (e.g., BERT: [0.10, 0.15, 0.18, 0.57, 0.00]) while the original design provides only the top predictions (e.g., BERT: star = 4, confidence = 0.57). Table 24 presents the performance comparison on the full 10,000-sample test sets.

Table 24. Comparison of top-1 + confidence vs. full-probability-vector input designs. The full-vector design provides the LLM with complete 5-class probability distributions for all base models, while the original design provides only the top prediction and confidence score.

Metric	Balanced Pipeline		Natural Pipeline		Average	
	Top-1	Full-Vec	Top-1	Full-Vec	Top-1	Full-Vec
Macro-F1	0.6800	0.6745	0.6720	0.6693	0.6760	0.6719
Accuracy	0.6865	0.6831	0.7538	0.7556	0.7202	0.7194
QWK	0.9111	0.9123	0.9337	0.9344	0.9224	0.9233
MAE	0.3332	0.3354	0.2611	0.2592	0.2972	0.2973
RMSE	0.6140	0.6135	0.5445	0.5422	0.5793	0.5779
Agreement (Top-1 vs. Full vec)	93.5%		94.6%		94.0%	

Key findings. The full-vector design achieves slightly lower macro-F1 scores compared to the original top-1 design (0.6719 vs. 0.6760 average), with a performance gap of 0.41 percentage points. The two designs agree on 94.0% of predictions, indicating high consistency. Interestingly, the full-vector design shows marginal improvements in QWK (0.9233 vs. 0.9224) and RMSE (0.5779 vs. 0.5793), suggesting that additional probability information may help with ordinal correctness even when it does not improve top-1 accuracy. These results justify our original design choice: the top-1 + confidence format provides comparable or slightly better performance while being more concise and interpretable for the LLM. The full probability vectors, while theoretically more informative, appear to add noise or complexity that does not translate into meaningful performance gains for this ensemble aggregation task. We hypothesize that the LLM's reasoning process benefits from clear, decisive signals (top predictions) rather than distributed probability mass that may reflect model uncertainty without providing actionable guidance for aggregation decisions.

4.14. Cross-Domain and Cross-Lingual Validation

To evaluate generalization beyond the English Yelp dataset, we conducted a cross-domain evaluation on Spanish Amazon Reviews. We downloaded the MTEB Amazon Reviews Multi dataset (Spanish subset) [47], which contains 200,000 training samples and 5000 validation/test samples. From this corpus, we created a balanced 10,000-sample subset (2000 per 1–5 star rating) with identical 70/15/15 train/validation/test splits as our Yelp experiments. We trained the same four base models (LR, SVM, NB, and Spanish BERT) using the identical Bayesian hyperparameter-tuning pipeline, then evaluated both standalone LLaMA and the LLM aggregator (LLAMA_AGG).

Table 25 presents the comparative results. On Spanish Amazon Reviews, LLAMA_AGG achieves 56.63% accuracy and 56.63% macro-F1, outperforming all the base models: BERT (55.44%), SVM (52.25%), LR (51.62%), and NB (50.06%). The standalone LLaMA achieves 55.44% accuracy. These results demonstrate that our framework is applicable and successful in other languages and datasets. The LLM aggregator successfully generalizes to a different language (Spanish) and domain (Amazon product reviews vs. Yelp service reviews), maintaining its position as the top-performing system. The performance gap between Yelp (68.00% macro-F1) and Spanish Amazon (56.63%) is expected given the cross-lingual challenge and the smaller, domain-shifted dataset, but the relative ranking of methods remains consistent.

Table 25. Cross-domain evaluation results: Yelp (English) vs. Spanish Amazon Reviews.

Model	Yelp Acc	Yelp F1	Amazon Acc	Amazon F1
LLAMA_AGG	0.6865	0.6800	0.5663	0.5663
BERT	0.6725	0.6725	0.5544	0.5544
SVM	0.5225	0.5219	0.5225	0.5225
LR	0.5162	0.5216	0.5162	0.5162
NB	0.5006	0.4992	0.5006	0.5006
LLAMA (standalone)	0.5947	0.5947	0.5544	0.5544

The LLM aggregator's reasoning capability in Spanish is exemplified by explanations such as *"La mayoría de los modelos, incluyendo BERT, LR, SVM y NB, coinciden en que la reseña merece 4 estrellas, lo que sugiere un acuerdo generalizado entre ellos. El sentimiento del texto también es positivo, mencionando que el producto cumple con lo esperado..."* (The majority of models, including BERT, LR, SVM and NB, agree that the review deserves 4 stars, suggesting generalized agreement among them. The sentiment of the text is also positive, mentioning that the product meets expectations...). This demonstrates that the LLM not only aggregates predictions but reasons over them in the target language, producing culturally and linguistically appropriate explanations.

4.15. Human Evaluation of Aggregated Explanation Quality

A human evaluation was conducted to assess the quality of explanations generated by the proposed XAI system. To ensure a representative sample for evaluation, we employed a stratified sampling approach across four distinct categories: *consensus* cases ($n = 35$), where all base models agree with the true rating; *override* cases ($n = 35$), where the aggregator correctly overrides disagreeing base models; *error* cases ($n = 25$), where the aggregator prediction is incorrect; and *edge* cases ($n = 6$), representing ambiguous or boundary scenarios. The final dataset comprises 101 reviews spanning all five star ratings (1-star: 22, 2-star: 18, 3-star: 13, 4-star: 23, 5-star: 25), with the aggregation system achieving an overall accuracy of 75.2% (76 out of 101 correct predictions).

Two independent judges evaluated each explanation across three carefully defined dimensions. **Faithfulness** assesses whether the explanation accurately reflects the underlying base-model evidence, with ratings from 1 (completely unfaithful) to 5 (fully faithful). **Usefulness** measures whether a business analyst would find the explanation actionable for decision making, rated from 1 (not useful) to 5 (highly useful). **Readability** evaluates the clarity and understandability of the language, rated from 1 (unclear) to 5 (very clear). Judges were provided with the review text, true star rating, the aggregator's prediction and reasoning, and the individual predictions and confidence scores from all four base models to enable informed assessments.

The evaluation [48] reveals that the aggregated explanations demonstrate strong performance in readability and faithfulness, while showing more variability in usefulness. As shown in Table 26, both judges rated readability highest (Judge 1: 4.84 ± 0.37 , Judge 2:

4.80 $\pm$ 0.40), indicating that explanations are consistently clear and well-structured. Faithfulness also received favorable ratings (Judge 1: 4.41 $\pm$ 0.80, Judge 2: 4.53 $\pm$ 0.63), suggesting that explanations generally align with the base-model evidence provided. However, usefulness ratings were notably lower and showed greater divergence between judges (Judge 1: 2.73 $\pm$ 0.51, Judge 2: 3.67 $\pm$ 0.49), with an overall mean of 3.20. This indicates that while explanations effectively describe model behavior, they often lack specific, actionable insights that business analysts could directly apply for operational improvements.

Table 26. Mean ratings by dimension and judge.

Dimension	Judge 1 Mean ($\pm$ SD)	Judge 2 Mean ($\pm$ SD)	Overall Mean	Interpretation
Faithfulness	4.41 $\pm$ 0.80	4.53 $\pm$ 0.63	4.47	High
Usefulness	2.73 $\pm$ 0.51	3.67 $\pm$ 0.49	3.20	Moderate
Readability	4.84 $\pm$ 0.37	4.80 $\pm$ 0.40	4.82	Very high

An inter-judge agreement analysis (Table 27) reveals consistency in readability assessment (Cohen's κ = 0.595, exact agreement: 88.1%), fair agreement in faithfulness (κ = 0.273, exact agreement: 60.4%), and poor agreement in usefulness (κ = −0.093, exact agreement: 15.8%). The negative kappa for usefulness suggests that judges applied different standards for what constitutes “actionable” business insight, with Judge 2 consistently rating usefulness higher than Judge 1. Despite these differences, both judges agreed within ± 1 point on over 90% of samples across all dimensions, indicating reasonable consistency in the overall assessment.

Table 27. Inter-judge agreement by dimension.

Dimension	Cohen's κ	Agreement Level	Pearson r	Exact Agree	Within ± 1
Faithfulness	0.273	Fair	0.360 *	60.4%	92.1%
Usefulness	−0.093	Poor	0.487 ***	15.8%	90.1%
Readability	0.595	Moderate	0.601 ***	88.1%	100.0%

* $p < 0.01$, *** $p < 0.001$

The judges' qualitative feedback provides valuable insights into the strengths and limitations of the aggregated explanations. For *consensus* cases, where all base models agreed, explanations were consistently rated as highly faithful (mean: 4.6) and readable (mean: 4.9), as they accurately reflected unanimous model agreement. However, judges noted that these explanations often provided generic summaries of model consensus rather than extracting specific business-relevant insights from the review text. For example, explanations frequently stated that “all models predicted X stars” without elaborating on the specific sentiment drivers that business analysts could act upon. Judge 1 specifically suggested that modifying the prompt to explicitly request business-relevant insights could address this limitation, indicating that the current prompt design may not sufficiently guide the LLM toward actionable output.

In *override* cases, where the aggregator correctly overrode disagreeing base models, explanations demonstrated the system's ability to synthesize conflicting evidence. Judges rated these explanations as moderately faithful (mean: 4.2), noting that while they generally captured the reasoning behind the override decision, they sometimes struggled to clearly articulate why the aggregator favored one model's prediction over others. The readability remained high (mean: 4.8), but usefulness varied considerably depending on whether the explanation identified specific actionable themes from the review.

Error cases, where the aggregator prediction was incorrect, revealed limitations. Judges rated these explanations lower in faithfulness (mean: 3.8), often noting that the explanations attempted to justify incorrect predictions by over-interpreting ambiguous review text or mischaracterizing the strength of model agreement (hallucinations [49]). Despite the

prediction errors, readability remained consistently high (mean: 4.7), suggesting that the aggregator produces well-written explanations even when the underlying prediction is wrong. This finding highlights the importance of the faithfulness dimension, as readable but unfaithful explanations could potentially mislead users.

The *edge* cases showed the most variability in ratings, with judges sometimes disagreeing substantially on faithfulness and usefulness. These cases typically involved ambiguous sentiment, sarcasm, or mixed positive and negative elements that challenged both the prediction system and the judges' assessment of explanation quality. Overall, the evaluation suggests that while the aggregation system produces clear and generally faithful explanations, there is significant room for improvement in generating actionable business insights, particularly in cases where model disagreement or prediction uncertainty is high.

5. Discussion

This section analyzes the XAI properties of the LLM-based aggregator and contextualizes the quantitative results. Unless stated otherwise, the detailed diagnostic statistics in this section are computed on the 10,000-sample **balanced** test set; the corresponding overall metrics for the **natural** test set are reported in Section 4 and referenced where relevant.

5.1. Performance Summary

LLAMA_AGG achieves the best macro-F1 on both pipelines: 0.6800 (balanced) and 0.6720 (natural), compared to 0.6725/0.6542 for the best classical aggregator (STACKING). The margin is modest, approximately 0.8 percentage points (pp) on balanced and 1.8 pp on natural, but consistent across all reported metrics: accuracy, QWK, MAE, and RMSE. These margins should be interpreted with appropriate calibration: the primary contribution of this work is not a large accuracy gain but rather the introduction of **explainability as a first-class output of the ensemble aggregation step**, supported by quantitatively characterized meta-reasoning behavior. Crucially, both LLAMA and LLAMA_AGG operate in a zero-shot setting without any task-specific fine-tuning, yet their performance is competitive, and in the case of LLAMA_AGG ranks first on all reported *overall* metrics (Tables 7 and 8)—including the lowest MAE/RMSE—compared to the specialized trained transformers and classical classifiers.

STACKING consistently ranks second on both pipelines, confirming that a learned meta-classifier exploits complementary base-model errors more effectively than fixed-weight voting schemes (finding 8, Section 5.9). Weighted Voting slightly outperforms Soft Voting, as expected, because it downweights model results that are associated with lower confidence.

All the models in this study achieve a high level of performance when viewed against a simple random-guessing baseline: in a five-class setting, a classifier that guesses uniformly at random achieves an expected macro-F1 of only

$$\mathbb{E}[\text{Macro-F1}_{\text{rand}}] = \frac{1}{K},$$

which for $K = 5$ equals 0.2; so our macro-F1 values of around 0.68 correspond to a more than three-fold improvement over pure chance. This is particularly substantial given that the Yelp dataset is a large, real-world corpus derived from millions of user reviews, and similar performance levels have been reported in the majority of the literature [50–55].

5.2. Finding 1: LLM Meta-Reasoning

The core scientific novelty of this work lies in the observation that the LLM does not passively relay the ensemble majority, but adds its own reasoning. On the balanced

pipeline, LLAMA and LLAMA_AGG share the same underlying model but receive different inputs: LLAMA sees only the review text; LLAMA_AGG additionally sees the four base-model predictions and confidence scores. They agree on only **71.93%** of cases (Table 16), meaning the LLM revises its own standalone prediction in **2807 of 10,000 cases (28.07%)** after observing the ensemble context.

When the LLM revises its prediction, the revised prediction is correct 56.4% of the time, compared to only 34.2% if the original standalone prediction had been kept, a **+22.2-percentage-point accuracy gain from meta-reasoning alone** (Table 28). In the remaining 9.4% of cases, neither the initial prediction nor the revised one was correct.

Table 28. Accuracy comparison for revised vs. agreed predictions on the balanced test set.

Scenario	<i>n</i>	LLAMA_AGG Acc.	LLAMA Acc.	Δ Acc.	Ordinal Closer
Revised (LLAMA_AGG $\neq$ LLAMA)	2807	0.564	0.342	+0.222	63.1%
Agreed (LLAMA_AGG = LLAMA)	7193	0.734	0.734	0	—

Moreover, 63.1% of revisions move the prediction ordinally closer to the ground truth, while for the remaining 36.9% of cases the revised prediction entailed a larger error than the original one. This demonstrates that the LLM’s revisions are not noise: they constitute a systematic, performance-improving inference process triggered by the structured evidence in the aggregation prompt.

5.3. Finding 2: Override vs. Consensus Level

We analyze when the LLM overrides the *base-model plurality* as a function of *base-model consensus*. For each review, let $k \in \{1, 2, 3, 4\}$ denote how many of the four base models agree on the most commonly predicted star rating. We define the plurality label as that most common star rating (breaking rare 2–2 ties by selecting the lower star rating), and call it an *override* when LLAMA_AGG predicts a different star rating.

Full (4/4) consensus occurs in 4716 samples (47.16%). In these cases, LLAMA_AGG adopts the unanimous rating in **4413 samples (93.6%)** and overrides in only **303 cases (6.4%)**; when it overrides, it is correct **57.8%** of the time, versus **33.7%** for the unanimous base-model rating on those same samples (Table 29).

Table 29. LLAMA_AGG accuracy and ordinal error by base-model consensus level on the balanced test set.

Consensus	<i>n</i>	%	Override Rate	AGG Acc. (Override)	AGG Acc. (No Override)	MAE/RMSE (Override)	MAE/RMSE (No Override)
1 of 4 agree	10	0.1%	80.0%	0.500	1.000	0.500/0.707	0.000/0.000
2 of 4 agree	1452	14.5%	55.6%	0.646	0.599	0.376/0.656	0.439/0.736
3 of 4 agree	3822	38.2%	60.2%	0.646	0.620	0.373/0.643	0.408/0.684
4 of 4 agree	4716	47.2%	6.4%	0.578	0.758	0.452/0.729	0.255/0.531

The LLM’s override behavior is calibrated to the evidence: it intervenes aggressively when consensus is low (≈ 53 – 92% override rate) and only rarely when consensus is high ($\approx 5\%$). This is consistent with rational Bayesian updating: the posterior probability of the base-model plurality increases with consensus, and the LLM defers in most high-consensus cases.

Importantly, overrides are *beneficial overall* relative to deferring to the base-model plurality: across all 3420 override cases (34.2% of the balanced test set), LLAMA_AGG improves the accuracy from 20.5% (plurality) to 64.0% and reduces the MAE/RMSE from 1.221/1.568 to 0.381/0.654; in 75.7% of overrides, the absolute error is reduced compared to the plurality prediction.

5.4. Finding 3: Correction of Polar Bias

Standalone LLAMA exhibits severe *polar bias* due to its zero-shot setting: it assigns high recall to the 1-star (0.9415) and 5-star (0.9820) classes while greatly under-predicting the 4-star class (recall: 0.3170) (Table 30). This is a known failure mode of instruction-tuned LLMs on fine-grained ordinal tasks. The LLM-based aggregator substantially corrects this.

Table 30. Recall improvement by star class: LLAMA vs. LLAMA_AGG (balanced test set).

Star	LLAMA rec	AGG rec	Δ rec	Missed by LLAMA	Corrected by AGG	Correction Rate
★1	0.9415	0.8920	−0.0495	117	27	23.1%
★2	0.4165	0.6065	+0.1900	1167	468	40.1%
★3	0.4630	0.4740	+0.0110	1074	253	23.6%
★4	0.3170	0.6665	+0.3495	1366	835	61.1%
★5	0.9820	0.7935	−0.1885	36	1	2.8%

The correction for the 4-star class is the most notable: the aggregator corrects 835 of 1366 missed samples (61.1%), nearly doubling recall from 0.317 to 0.667. The slight decreases in recall for the 1- and 5-star classes are the expected ordinal trade-off, while probability mass is redistributed toward mid-range classes. This finding has a clear mechanistic interpretation where the base models, trained on balanced data, produce calibrated prediction proportions across all five classes; when the LLM-based aggregator observes that three or four of them predict 4 stars, it revises its polar-biased standalone judgment accordingly.

Business interpretation of the trade-off. The shift from extreme-star-rating recall (1 and 5 stars) toward mid-range recall (4 stars) is desirable for business intelligence applications. Mid-range reviews (2–4 stars) contain the most actionable nuance for customer-experience managers, as they often identify specific improvement opportunities rather than blanket praise or condemnation. Over-predicting extremes is a known failure mode of zero-shot LLMs, which tend to polarize toward confident binary judgments. By recalibrating toward the balanced base-model consensus, the aggregator surfaces more of these nuanced, decision-relevant reviews. The slight reduction in extreme-star-rating recall is an acceptable trade-off because: (1) 1- and 5-star ratings are already well-detected by classical models, so the LLM’s unique value lies in resolving ambiguity; and (2) business stakeholders benefit more from correctly identifying borderline cases than from marginal gains on already-clear extremes.

5.5. Finding 4: Reasoning Length as an Explanatory Signal

Every LLAMA_AGG prediction is accompanied by a free-text explanation. The mean length of these explanations across 10000 predictions is **67.6 words** (median: 67.0, SD: 7.3) (Table 31). Crucially, the explanation length is a *quantitative signal of the LLM’s internal uncertainty*.

Table 31. Mean explanation length (words) by prediction outcome and override behavior.

Group	<i>n</i>	Mean Words	Median	SD
Override (any)	3420	71.5	71.0	7.0
No override	6580	65.6	65.0	6.6
Low consensus (2/4 agree)	1452	70.8	71.0	7.2
High consensus (4/4 agree)	4716	64.2	64.0	6.2
Incorrect predictions	3135	68.4	68.0	7.5
Correct predictions	6865	67.3	67.0	7.2

Overrides produce explanations that are on average **5.9 words longer** than non-override explanations (71.5 vs. 65.6). Low-consensus cases elicit 6.6 words more reasoning than high-consensus cases (70.8 vs. 64.2). This pattern is consistent with human deliberation: more justificatory text is written when evidence is conflicted. Explanation length is thus a

lightweight proxy for *prediction difficulty* that requires no additional model calls: a practical advantage for downstream human review prioritization.

5.6. Finding 5: Exclusive XAI Gain over All Classical Aggregators

LLAMA_AGG revises 496 samples (4.96%) that all three classical aggregators misclassify, while failing on only 352 samples (3.52%) that the classical aggregators handle correctly (net exclusive gain of **+144 samples**; Table 32). The largest net gain is on 1-star cases (+135), where the LLM’s ability to identify strongly negative sentiment adds value. The largest loss is on 3-star cases (−108), the most ambiguous class, where LLM reasoning occasionally introduces errors that consensus voting avoids.

Table 32. Samples where LLAMA_AGG is uniquely correct (XAI gain) vs. uniquely wrong (XAI loss) compared to all three classical aggregators simultaneously.

Star	XAI Gain <i>n</i>	Gain Rate	XAI Loss <i>n</i>	Loss Rate	Net Gain
★1	136	6.80%	1	0.05%	+135
★2	111	5.55%	150	7.50%	−39
★3	43	2.15%	151	7.55%	−108
★4	126	6.30%	41	2.05%	+85
★5	80	4.00%	9	0.45%	+71
Total	496	4.96%	352	3.52%	+144

5.7. Finding 6: Collective Correctness and LLAMA_AGG

When we count how many of the nine systems (LR, SVM, NB, BERT, Soft Voting, Weighted Voting, Stacking, LLAMA, and LLAMA_AGG) are correct on a given sample, LLAMA_AGG’s accuracy is tightly coupled to the collective count. The increase in LLAMA_AGG accuracy from ~0.42 (four models correct) to ~0.84 (five models correct) marks the majority tipping point: once more than half of the ensemble is correct, the LLM correctly tracks the majority. At the extreme ($n_{\text{correct}} = 0$), LLAMA_AGG is essentially never correct (0.000), confirming that the LLM does not manufacture correct answers from uniformly faulty inputs; it is an aggregator of signal, not a generator of independent intelligence.

5.8. Finding 7: Base-Model Confidence Signature for Overrides

When LLAMA_AGG overrides the ensemble majority, base-model confidence scores are systematically lower than in non-override samples (Table 33).

Table 33. Mean base-model confidence on override vs. non-override samples.

Model	Mean Conf. (No Override)	Mean Conf. (Override)	Δ
BERT	0.7743	0.7031	−0.0712
SVM	0.5331	0.4617	−0.0714
LR	0.3739	0.3244	−0.0495
NB	0.3391	0.3004	−0.0387

Confidence drops by 4–7 percentage points on overridden samples across all four base models. This suggests that the LLM prompt implicitly conveys uncertainty through lower confidence values, and the model responds with proportionally greater consultation, a form of calibration-aware meta-reasoning. This has a practical implication in which confidence thresholding can be used to predict override probability, enabling selective human review of high-uncertainty predictions.

5.9. Findings 8–10: Additional Notable Results

- Finding 8 (STACKING is best classical aggregator). STACKING ranks second on both pipelines, outperforming Weighted Voting and Soft Voting. This confirms that

the meta-classifier learns a data-driven linear combination of base-model evidence that fixed-weight averaging cannot represent, which is consistent with the theoretical advantages of stacking [7].

- Finding 9 (NB catastrophic failure on natural distribution). NB achieves near-zero recall on 2-star (0.38%) and 3-star (0.84%) cases on the natural test set (Table 14), degenerating into a binary classifier. This failure is invisible in the balanced pipeline, where NB achieves a respectable macro-F1 of 0.4992. This result underscores that **balanced training and balanced evaluation can mask severe model failures** that only manifest under realistic class distributions.
- Finding 10 (Accuracy vs. macro-F1 divergence). NB achieves an accuracy of 0.6007 but macro-F1 of 0.3465 on the natural pipeline, a 0.254-point divergence. This is the most extreme demonstration in our results of why accuracy should not be the primary metric for imbalanced classification problems, reinforcing the choice of macro-F1 as our primary evaluation criterion throughout [3].

5.10. Business Intelligence Applicability

The natural-language explanations produced by LLAMA_AGG are directly actionable for business stakeholders. A customer-experience manager reviewing a predicted three-star batch can read the LLM's justification, e.g., "The review praises the food quality (LR: 5 stars) but complains about service delay (SVM: 2 stars, confidence 0.61). The conflicting signals and the specific mention of 'slow service' lead me to assign 3 stars.", and immediately understand why the review is borderline. This is qualitatively different from a SHAP bar chart of token importance, which requires statistical literacy to interpret.

From a regulatory perspective, emerging AI-accountability frameworks (e.g., the EU AI Act [56]) require explainability for high-risk automated decisions affecting individuals. While business-review classification does not currently fall into high-risk categories, the methodology demonstrated here, transparent-by-design ensemble aggregation, is directly applicable to regulated contexts such as credit scoring, HR screening, and medical triage, where explainability obligations are already active.

5.11. Limitations

Hallucination risks. While the LLM aggregator produces natural-language explanations, these texts may occasionally contain hallucinations—statements not grounded in the base-model evidence or the review text. Although our quantitative analysis shows that 63.1% of revisions move ordinally closer to the ground truth, there remains a 36.9% minority where the LLM's reasoning introduces larger errors. Users should treat explanations as suggestive rather than authoritative, and critical decisions should involve human review.

Explanation reliability variance. The quality of generated explanations varies across samples. High-consensus cases (4/4 agreement) tend to produce shorter, more consistent justifications, whereas low-consensus or ambiguous reviews elicit longer but potentially less reliable reasoning. This variance is not captured by aggregate metrics and represents an inherent limitation of LLM-generated text.

Dependency on a limited set of LLM families. The primary experiments use the LLaMA-3.3-70B-Instruct model, with supplementary comparisons against LLaMA-3.1-8B-Instruct and Mistral-7B-Instruct. The extent to which findings generalize to other LLM families (e.g., GPT-4, Gemma) with different architectures, training data, or safety fine-tuning remains unexplored.

Inference cost. The LLM aggregation step requires approximately 20 h per 10,000-sample pipeline on a single NVIDIA H100 NVL GPU (Table 6). This is acceptable

for batch-analytics but prohibitive for real-time per-request deployment without significant inference-optimization (e.g., speculative decoding, model distillation).

Limited cross-domain evaluation. The primary experiments use English-language Yelp reviews, with a supplementary cross-lingual evaluation on Spanish Amazon Reviews. Generalization to other languages, domains (e.g., hotel or product reviews), or review platforms beyond these two datasets is not evaluated.

Prompt sensitivity. LLM performance is sensitive to prompt design, including phrasing, information ordering, and formatting conventions. We use a single prompt template (Section 3.4.4) validated informally on a small development set; systematic prompt optimization via automated search or A/B testing was not performed due to computational constraints. This represents a limitation, as alternative prompting strategies (e.g., chain-of-thought variants, few-shot examples, different XML schemas) might yield different predictive accuracy or explanation quality. We identify prompt-engineering sensitivity as a priority for future work.

Calibration of base-model probabilities. The LLM prompt relies on confidence scores from Platt-scaled classical models and BERT softmax outputs, both of which may be poorly calibrated in certain regions of the prediction space, potentially biasing the aggregator's context.

5.12. Future Work

Several directions remain open and will be considered in the context of our future work. First, distilling the LLM aggregator into a lightweight specialist model to reduce the ~20 h inference cost. Second, we plan to extend the cross-lingual evaluation beyond Spanish Amazon Reviews to additional languages and review platforms. In the context of prompt engineering, we will investigate both manual and automated prompt-optimization to further improve aggregation accuracy. The integration of calibration-aware base-model confidence estimates to improve the quality of the aggregation context will be also studied. Finally, we will consider the application of the LLM-as-aggregator paradigm to other high-stakes classification tasks in healthcare or finance, where explanation obligations are regulatory requirements.

6. Conclusions

In this paper, we presented an LLM-based XAI-ensemble aggregator for five-class business-review sentiment analysis, evaluated extensively on the Yelp Academic Dataset across balanced and natural-distribution test pipelines, with additional cross-lingual validation on Spanish Amazon Reviews. The LLM aggregator (unsloth/Llama-3.3-70B-Instruct) consistently outperforms all classical ensemble methods (Soft Voting, Weighted Voting, and Stacking) and all base models (LR, SVM, NB, and BERT) on macro-F1, achieving 0.6800 on the balanced test set and 0.6720 on the natural-distribution test set, while also leading in QWK, MAE, and RMSE.

Beyond predictive performance, the paper's central contribution is the demonstration that **explainability can be a primary, automatically generated output of ensemble aggregation**. The LLM revises approximately 28% of its standalone predictions after observing the ensemble evidence; those revisions improve accuracy by +22.2 percentage points, with 63.1% of revisions moving ordinally closer to the ground truth. The aggregator corrects severe polar bias in the standalone LLM predictions (up to +35 pp recall improvement on 4-star class) and produces longer explanations precisely when the evidence is most conflicted, a quantitative signal of deliberative reasoning. A formal human evaluation with two judges confirms high explanation faithfulness (4.47/5) and readability (4.82/5), validating that explanations accurately reflect the decision process.

Our extensive ablation studies demonstrate that effective aggregation does not require the largest available models: an 8B parameter variant achieves 90.8% agreement with the 70B model while requiring only 4 GB GPU memory versus 40–50 GB, enabling practical deployment. Cross-lingual evaluation on Spanish Amazon Reviews confirms the framework generalizes across languages and domains, maintaining top performance (56.63% macro-F1) despite the language shift. Base-model calibration analysis reveals the LLM acts as a calibration-aware arbiter, intervening more when confidence estimates are less reliable.

From a business-intelligence perspective, the per-prediction natural-language explanations are immediately interpretable by non-technical stakeholders, enabling auditable, human-overseen sentiment pipelines. The methodology is aligned with transparent-by-design XAI principles and is applicable to any domain where ensemble classification outputs require justification.

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. and 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 has been utilized equipment that was acquired through funding by the European Union–NextGenerationEU and the Recovery and Resilience Facility (Greece 2.0), under the project SUB2: “Universities of Excellence” (Project code: OPS TA 5180665).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The Yelp Academic Dataset is publicly available at <https://www.yelp.com/dataset> (accessed on 25 June 2026). All model prediction files, evaluation outputs, and analysis scripts produced in this study are openly released on GitHub to ensure reproducibility, transparency, and benchmarking [37].

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

References

1. Liu, B. *Sentiment Analysis and Opinion Mining*; Morgan & Claypool Publishers: San Rafael, CA, USA, 2012. <https://doi.org/10.2200/S00416ED1V01Y201204HLT016>.
2. Mukherjee, A.; Liu, B. Modeling Review Helpfulness. In Proceedings of the 2012 AAAI Workshop on Intelligent Techniques for Web Personalization and Recommendation, Toronto, ON, Canada, 22–23 July 2012.
3. Zhang, L.; Wang, S.; Liu, B. Deep Learning for Sentiment Analysis: A Survey. *WIREs Data Min. Knowl. Discov.* **2018**, *8*, e1253. <https://doi.org/10.1002/widm.1253>.
4. Joachims, T. Text Categorization with Support Vector Machines: Learning with Many Relevant Features. In *Proceedings of the 10th European Conference on Machine Learning*; Springer: Berlin/Heidelberg, Germany, 1998; pp. 137–142. <https://doi.org/10.1007/BFb0026683>.
5. McCallum, A.; Nigam, K. A Comparison of Event Models for Naive Bayes Text Classification. In Proceedings of the AAAI/ICML-98 Workshop on Learning for Text Categorization, Madison, WI, USA, 26–27 July 1998; pp. 41–48.
6. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Minneapolis, MN, USA, 2–7 June 2019; Volume 1, pp. 4171–4186. <https://doi.org/10.18653/v1/N19-1423>.
7. Wolpert, D.H. Stacked Generalization. *Neural Netw.* **1992**, *5*, 241–259. [https://doi.org/10.1016/S0893-6080\(05\)80023-1](https://doi.org/10.1016/S0893-6080(05)80023-1).
8. Kuncheva, L.I. *Combining Pattern Classifiers: Methods and Algorithms*; Wiley-Interscience: Hoboken, NJ, USA, 2004. <https://doi.org/10.1002/0471660264>.

9. Ribeiro, M.T.; Singh, S.; Guestrin, C. “Why Should I Trust You?”: Explaining the Predictions of Any Classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*; ACM: New York, NY, USA, 2016; pp. 1135–1144. <https://doi.org/10.1145/2939672.2939778>.
10. Lundberg, S.M.; Lee, S.I. A Unified Approach to Interpreting Model Predictions. In *Advances in Neural Information Processing Systems*; Curran Associates, Inc.: Red Hook, NY, USA, 2017; Volume 30.
11. Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; Ichter, B.; Xia, F.; Chi, E.; Le, Q.V.; Zhou, D. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. In *Advances in Neural Information Processing Systems*; Curran Associates, Inc.: Red Hook, NY, USA, 2022; Volume 35, pp. 24824–24837.
12. Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. Language Models Are Few-Shot Learners. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 1877–1901.
13. Grattafiori, A.; Dubey, A.; Jauhri, A.; Pandey, A.; Kadian, A.; Al-Dahle, A.; Letman, A.; Mathur, A.; Schelten, A.; Vaughan, A.; et al. The Llama 3 Herd of Models. *arXiv* **2024**, arXiv:2407.21783.
14. Han, D.; Han, M.; Unsloth Team. Unsloth: 2–5× Faster LLM Fine-Tuning. 2023. Available online: <https://github.com/unslothai/unsloth> (accessed on 26 March 2026).
15. Yelp Inc.. Yelp Open Dataset. 2024. Available online: <https://www.yelp.com/dataset> (accessed on 26 March 2026).
16. Medhat, W.; Hassan, A.; Korashy, H. Sentiment Analysis Algorithms and Applications: A Survey. *Ain Shams Eng. J.* **2014**, *5*, 1093–1113. <https://doi.org/10.1016/j.asej.2014.04.011>.
17. Roumeliotis, K.I.; Margaritis, D.; Spiliotopoulos, D.; Vassilakis, C. A Large-Scale Empirical Study of LLM Orchestration and Ensemble Strategies for Sentiment Analysis in Recommender Systems. *Future Internet* **2026**, *18*, 112. <https://doi.org/10.3390/fi18020112>.
18. Roumeliotis, K.I.; Tselikas, N.D.; Nasiopoulos, D.K. Optimizing Airline Review Sentiment Analysis: A Comparative Analysis of LLaMA and BERT Models through Fine-Tuning and Few-Shot Learning. *Comput. Mater. Contin.* **2025**, *82*, 2769–2792. <https://doi.org/10.32604/cmc.2025.059567>.
19. Roumeliotis, K.I.; Tselikas, N.D.; Nasiopoulos, D.K. LLMs for product classification in e-commerce: A zero-shot comparative study of GPT and claude models. *Nat. Lang. Process. J.* **2025**, *11*, 100142. <https://doi.org/10.1016/j.nlp.2025.100142>.
20. Toudas, K.; Roumeliotis, K.I.; Nasiopoulos, D.K.; Georgakopoulos, G. An Explainable AI Multi-Agent Recommender System for Financial Document Access Control. *Inf. Syst. Front.* **2026**. <https://doi.org/10.1007/s10796-026-10742-2>.
21. Salton, G.; Buckley, C. Term-weighting Approaches in Automatic Text Retrieval. *Inf. Process. Manag.* **1988**, *24*, 513–523. [https://doi.org/10.1016/0306-4573\(88\)90021-0](https://doi.org/10.1016/0306-4573(88)90021-0).
22. Zhang, Q.; Zhu, Y.; Cordeiro, F.R.; Chen, Q. PSSCL: A progressive sample selection framework with contrastive loss designed for noisy labels. *Pattern Recognit.* **2025**, *161*, 111284. <https://doi.org/10.1016/j.patcog.2024.111284>.
23. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.u.; Polosukhin, I. Attention is All you Need. In *Advances in Neural Information Processing Systems*; Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2017; Volume 30.
24. Bello, A.; Ng, S.C.; Leung, M.F. A BERT Framework to Sentiment Analysis of Tweets. *Sensors* **2023**, *23*, 506. <https://doi.org/10.3390/s23010506>.
25. Zhou, Z.H. *Ensemble Methods: Foundations and Algorithms*; CRC Press: Boca Raton, FL, USA, 2012.
26. Zhao, W.X.; Zhou, K.; Li, J.; Tang, T.; Wang, X.; Hou, Y.; Min, Y.; Zhang, B.; Zhang, J.; Dong, Z.; et al. A Survey of Large Language Models. *arXiv* **2026**, arXiv:2303.18223.
27. Wang, X.; Wei, J.; Schuurmans, D.; Le, Q.V.; Chi, E.H.; Narang, S.; Chowdhery, A.; Zhou, D. Self-Consistency Improves Chain of Thought Reasoning in Language Models. In *Proceedings of the International Conference on Learning Representations*, Kigali, Rwanda, 1–5 May 2023.
28. Zhang, W.; Deng, Y.; Liu, B.; Pan, S.J.; Bing, L. Sentiment Analysis in the Era of Large Language Models: A Reality Check. *arXiv* **2023**, arXiv:2305.15005.
29. Touvron, H.; Lavril, T.; Izacard, G.; Martinet, X.; Lachaux, M.A.; Lacroix, T.; Rozière, B.; Goyal, N.; Hambro, E.; Azhar, F.; et al. LLaMA: Open and Efficient Foundation Language Models. *arXiv* **2023**, arXiv:2302.13971.
30. AutoGPT Team. AutoGPT: An Autonomous GPT-4 Experiment. 2023. Available online: <https://github.com/Significant-Gravitas/AutoGPT> (accessed on 25 June 2026).
31. Yao, S.; Zhao, J.; Yu, D.; Du, N.; Shafran, I.; Narasimhan, K.; Cao, Y. ReAct: Synergizing Reasoning and Acting in Language Models. *arXiv* **2023**, arXiv:2210.03629.
32. 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.
33. 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.

34. Zheng, L.; Chiang, W.L.; Sheng, Y.; Zhuang, S.; Wu, Z.; Zhuang, Y.; Lin, Z.; Li, Z.; Li, D.; Xing, E.P.; et al. Judging LLM-as-a-judge with MT-bench and Chatbot Arena. In *Proceedings of the NIPS '23 37th International Conference on Neural Information Processing Systems*; Curran Associates: Red Hook, NY, USA, 2023.
35. Chang, Y.; Wang, X.; Wang, J.; Wu, Y.; Yang, L.; Zhu, K.; Chen, H.; Yi, X.; Wang, C.; Wang, Y.; et al. A Survey on Evaluation of Large Language Models. *ACM Trans. Intell. Syst. Technol.* **2024**, *15*, 39. <https://doi.org/10.1145/3641289>.
36. Arrieta, A.B.; Díaz-Rodríguez, N.; Del Ser, J.; Bennetot, A.; Tabik, S.; Barbado, A.; García, S.; Gil-López, S.; Molina, D.; Benjamins, R.; et al. Explainable Artificial Intelligence (XAI): Concepts, Taxonomies, Opportunities and Challenges toward Responsible AI. *Inf. Fusion* **2020**, *58*, 82–115. <https://doi.org/10.1016/j.inffus.2019.12.012>.
37. Applied AI Research Lab. Large Language Models as Explainable AI Ensemble Aggregators for Business Review Sentiment Analysis. GitHub Repository. 2026. Available online: <https://github.com/Applied-AI-Research-Lab/Large-Language-Models-as-Explainable-AI-Ensemble-Aggregators-for-Business-Review-Sentiment-Analysis> (accessed on 22 March 2026).
38. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-learn: Machine Learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830.
39. Soubies, E.; Blanc-Féraud, L.; Aubert, G. A Unified View of Exact Continuous Penalties for ℓ_2 - ℓ_0 Minimization. *SIAM J. Optim.* **2017**, *27*, 2034–2060. <https://doi.org/10.1137/16m1059333>.
40. Berkani, M.S.; Giurgea, S.; Espanet, C.; Coulomb, J.L.; Kieffer, C. Study on Optimal Design Based on Direct Coupling Between a FEM Simulation Model and L-BFGS-B Algorithm. *IEEE Trans. Magn.* **2013**, *49*, 2149–2152. <https://doi.org/10.1109/TMAG.2013.2245871>.
41. Böken, B. On the appropriateness of Platt scaling in classifier calibration. *Inf. Syst.* **2021**, *95*, 101641. <https://doi.org/10.1016/j.is.2020.101641>.
42. Liu, A.Y.; Martin, C.E., Smoothing Multinomial Naïve Bayes in the Presence of Imbalance. In *Machine Learning and Data Mining in Pattern Recognition*; Springer: Berlin/Heidelberg, Germany, 2011; pp. 46–59. https://doi.org/10.1007/978-3-642-23199-5_4.
43. Akiba, T.; Sano, S.; Yanase, T.; Ohta, T.; Koyama, M. Optuna: A Next-generation Hyperparameter Optimization Framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*; ACM: Newyork, NY, USA, 2019; pp. 2623–2631. <https://doi.org/10.1145/3292500.3330701>.
44. Sokolova, M.; Lapalme, G. A systematic analysis of performance measures for classification tasks. *Inf. Process. Manag.* **2009**, *45*, 427–437. <https://doi.org/10.1016/j.ipm.2009.03.002>.
45. Cohen, J. Weighted Kappa: Nominal Scale Agreement Provision for Scaled Disagreement or Partial Credit. *Psychol. Bull.* **1968**, *70*, 213–220. <https://doi.org/10.1037/h0026256>.
46. Margaritis, D.; Vassilakis, C.; Spiliotopoulos, D. From Rating Predictions to Reliable Recommendations in Collaborative Filtering: The Concept of Recommendation Reliability Classes. *Big Data Cogn. Comput.* **2025**, *9*, 106. <https://doi.org/10.3390/bdcc9040106>.
47. MTEB. Amazon Reviews Multi Dataset. 2025. Available online: https://huggingface.co/datasets/mteb/amazon_reviews_multi (accessed on 22 June 2026).
48. Applied AI Research Lab. Human Evaluation. GitHub Repository. 2026. Available online: https://github.com/Applied-AI-Research-Lab/Large-Language-Models-as-Explainable-AI-Ensemble-Aggregators-for-Business-Review-Sentiment-Analysis/tree/main/human_evaluation (accessed on 19 June 2026).
49. Sriramanan, G.; Bharti, S.; Sadasivan, V.S.; Saha, S.; Kattakinda, P.; Feizi, S. LLM-Check: Investigating Detection of Hallucinations in Large Language Models. In *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. 34188–34216. <https://doi.org/10.52202/079017-1077>.
50. Asghar, N. Yelp Dataset Challenge: Review Rating Prediction. *arXiv* **2016**, arXiv:1605.05362.
51. Liu, Z. Yelp Review Rating Prediction: Machine Learning and Deep Learning Models. *arXiv* **2020**, arXiv:2012.06690.
52. Sun, Y. Prediction of Yelp Score from Reviews with Machine Learning Model. Master's Thesis, University of California, Los Angeles, CA, USA, 2022.
53. Belaroussi, R.; Noufe, S.C.; Dupin, F.; Vandanjon, P.O. Polarity of Yelp Reviews: A BERT–LSTM Comparative Study. *Big Data Cogn. Comput.* **2025**, *9*, 140. <https://doi.org/10.3390/bdcc9050140>.
54. Liu, S. Sentiment Analysis of Yelp Reviews: A Comparison of Techniques and Models. *arXiv* **2020**, arXiv:2004.13851.
55. Guda, B.P.R.; Srivastava, M.; Karkhanis, D. Sentiment Analysis: Predicting Yelp Scores. *arXiv* **2022**, arXiv:2201.07999.
56. Future of Life Institute. EU Artificial Intelligence Act. 2024. Available online: <https://artificialintelligenceact.eu/> (accessed on 22 March 2026).

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.