ReviewMirror: Baseline Model Design & Experimental Framework

Shubham (202411066), Ritwik (202411067), Dhairya (202411082)

1 Introduction

This report presents our baseline system for ReviewMirror, a reproducible pipeline that detects and quantifies opinion drift in e-commerce reviews. Building on Milestone 1's conceptual design, we now deliver a working prototype that processes the Amazon Electronics 5-core dataset end to end, produces interpretable drift metrics, and establishes an experimental framework for future improvements.

Objective. Our goal is to create a *minimal yet functional* baseline that: (1) implements the preprocessing and feature engineering pipeline, (2) computes per user drift trajectories with quantitative metrics, (3) defines reproducible train/validation/test splits, and (4) produces diagnostic visualizations and summary statistics. This baseline serves as the foundation against which we will evaluate adaptive models in later milestones.

2 Baseline System Overview

2.1 Pipeline Architecture

Our system follows a six-stage pipeline (Figure 1):

1. **Data Loading:** Parse JSON/JSONL reviews with robust handling of gzipped files and both line-delimited and array formats.
2. **Cleaning & Filtering:** Drop rows missing critical fields (`user_id`, `item_id`, `timestamp`); retain reviews with $\geq 90\%$ ASCII characters to ensure English text quality.
3. **Feature Engineering:** Compute text based sentiment (VADER), normalize star ratings to $[-1, 1]$, and create hybrid sentiment score. Extract stylistic features (exclamation rate, first-person usage, capitalization).
4. **Temporal Aggregation:** Bin reviews into calendar months; aggregate per user month to create smooth trajectories. Retain only users with ≥ 5 reviews for longitudinal analysis.
5. **Drift Quantification:** Compute four key metrics per user: (a) linear trend slope, (b) start to end delta, (c) total variation (volatility), (d) sign flip rate (instability).
6. **Output Generation:** Save cleaned data (Parquet format), user trajectories, train/val/test splits, summary metrics (JSON), and diagnostic plots (PNG).

Raw JSON Reviews → Clean DataFrame → Features → Monthly Aggregates
→ Drift Metrics → Splits + Outputs

Figure 1: Baseline pipeline stages from raw reviews to drift analysis outputs.

2.2 Hybrid Sentiment Model

Our baseline drift detector relies on a *hybrid sentiment* signal that combines text-based and rating-based information:

$$h_i = \alpha \cdot s_i^{\text{text}} + (1 - \alpha) \cdot \tilde{r}_i^{\text{stars}}, \qquad \alpha = 0.7,$$

where $s_i^{\text{text}} \in [-1, 1]$ is the VADER compound score, and $\tilde{r}_i^{\text{stars}} = (r_i - 3)/2$ maps star ratings (1–5) to $[-1, 1]$. The weighting α=0.7 prioritizes text polarity while retaining rating information as a complementary signal. We chose this design because:

- **Text captures nuance:** VADER detects linguistic polarity that star ratings miss (e.g., "good but...").
- **Ratings are reliable:** Stars provide a stable, explicit judgment less prone to sarcasm or ambiguity.
- **Hybrid is robust:** Combining both reduces noise from either source alone.

2.3 Drift Metrics

For each user u with monthly sentiment sequence $\{h_t\}_{t=1}^{T_u}$, we compute:

1. **Drift Slope ($\hat{\beta}_1$):** Ordinary least-squares trend coefficient fitting $h_t \sim \beta_0 + \beta_1 \cdot t$, where t is month ordinal (YYYY×12 + MM). Positive values indicate upward drift; negative values indicate downward drift.
2. **Drift Delta (Δh):** Net change from first to last review: $\Delta h = h_{T_u} - h_1$. Captures endpoint difference independent of intermediate volatility.
3. **Total Variation (TV):** Cumulative absolute month-to-month change: $\text{TV} = \sum_{t=2}^{T_u} |h_t - h_{t-1}|$. Measures trajectory volatility/instability.
4. **Flip Rate:** Proportion of months where sentiment sign changes: $\text{FlipRate} = \frac{1}{T_u - 1} \sum_{t=2}^{T_u} \mathbb{1}[\text{sign}(h_t) \neq \text{sign}(h_{t-1})]$. Quantifies erratic behavior.

These metrics are complementary: slope captures overall direction, delta measures endpoints, TV quantifies volatility, and flip rate detects instability patterns.

3 Experimental Configuration

3.1 Dataset & Preprocessing

We processed the **Amazon Electronics 5-core** dataset (200,000 reviews for computational efficiency during development; full dataset contains ∼1.7M reviews). After filtering:

- **Reviews retained:** 17,706 (8.85% of loaded subset)
- **Users with $\geq$5 reviews:** 2,657
- **Unique items:** 2,884
- **Date range:** 1996–2018

3.2 Train/Validation/Test Splits

To ensure reproducible evaluation, we partition users into three sets using a *temporal* splitting strategy:

- **Train (70%):** 1,860 users are early adopters by first review date. Used to compute baseline statistics (mean drift slope, median delta, etc.).
- **Validation (15%):** 397 users are middle cohort for hyperparameter tuning in future milestones.
- **Test (15%):** 400 users are recent users held out for final evaluation.

Rationale: Temporal splits simulate realistic deployment where models trained on historical users must generalize to new users. Random splits would leak temporal information and overestimate performance.

3.3 Reproducibility Measures

Every experiment is fully reproducible via:

1. **Configuration file (`config.json`):** Records all hyperparameters (e.g., α=0.7, `MIN_REVIEWS`=5, random seed=42).
2. **Data splits (`splits.json`):** Saves exact user IDs in train/val/test sets.
3. **Manifest (`manifest.json`):** Logs Python version, package versions (pandas, numpy, VADER), and preprocessing steps.
4. **Execution log (`pipeline.log`):** Timestamped console output for debugging.
5. **Organized outputs:** All artifacts stored in timestamped run directory (`runs/baseline_v1`).

4 Baseline Results

4.1 Quantitative Summary

Table 1 reports key statistics across all users and on the test set. The baseline exhibits moderate drift: mean slope is near zero with high variance, indicating heterogeneity in user trajectories.

Metric	**Mean**	**Std**	**Min**	**Max**
Drift Slope	0.0004	0.0656	-0.9180	1.2750
Drift Delta	-0.0192	0.5098	-1.9210	1.9201
Total Variation	1.1383	1.2576	0.0	21.2435
Flip Rate	0.1270	0.2494	0.0	1.0
Test Set (n=400)				
High-Drift Users ($\|\text{slope}\| > 0.01$)	150 (37.5%)			
Volatile Users (TV > Q3)	64 (16.0%)			
Flat Users ($\|\text{slope}\| < 0.001$)	27 (6.8%)			

Table 1: Baseline drift metrics summary across 2,657 users with actual values from experimental run.

4.2 Naive Baseline Performance

We evaluate a trivial predictor that assigns all test users the *mean drift slope* from the training set. This serves as a lower bound for future adaptive models:

- **Mean Absolute Error (Slope):** 0.0472 (test set)
- **Mean Absolute Error (Delta):** 0.2364 (test set)

These errors are substantial relative to the typical drift magnitude, confirming that user-specific trajectories are heterogeneous and cannot be captured by a single global average.

4.3 Distribution Analysis

Figure 2 shows histograms of the four drift metrics. Key observations:

- **Drift Slope:** Approximately symmetric around zero with heavy tails, suggesting most users are stable but a minority exhibit strong upward/downward trends.
- **Drift Delta:** Similar distribution to slope but with slightly more mass at extremes (consistent with non-linear trajectories).
- **Total Variation:** Right-skewed distribution; median TV ≈ 0.31 indicates moderate volatility for typical users.
- **Flip Rate:** Bimodal distribution with peaks near 0 (stable polarity) and 0.5 (frequent flips), revealing distinct user behavioral patterns.

4.4 Temporal Trends

Figure 3 aggregates sentiment across all users by calendar month. We observe:

- **Overall stability:** Average hybrid sentiment hovers near 0.63 with minor fluctuations, suggesting positive overall sentiment with no dataset-wide temporal bias.
- **Volume trends:** Review counts increase over time (typical for growing platforms), with seasonal spikes potentially reflecting holiday shopping patterns.

4.5 Correlation Analysis

Figure 4 plots drift slope vs. total variation. The weak correlation ($r \approx 0.12$) indicates that *direction* of drift (slope) and *stability* (TV) are largely independent. This justifies treating them as separate dimensions in user characterization.

4.6 Representative User Trajectories

Figure 5 shows six randomly sampled users illustrating diverse drift patterns:

- **User A:** Steady upward drift (positive slope, low TV).
- **User B:** U-shaped trajectory (negative delta, moderate TV).
- **User C:** Volatile with no clear trend (near-zero slope, high TV).
- **User D:** Plateau after initial negativity (negative delta, low late-stage TV).
- **User E:** Steady downward drift (negative slope, low TV).
- **User F:** Flat/stable (near-zero slope and TV).

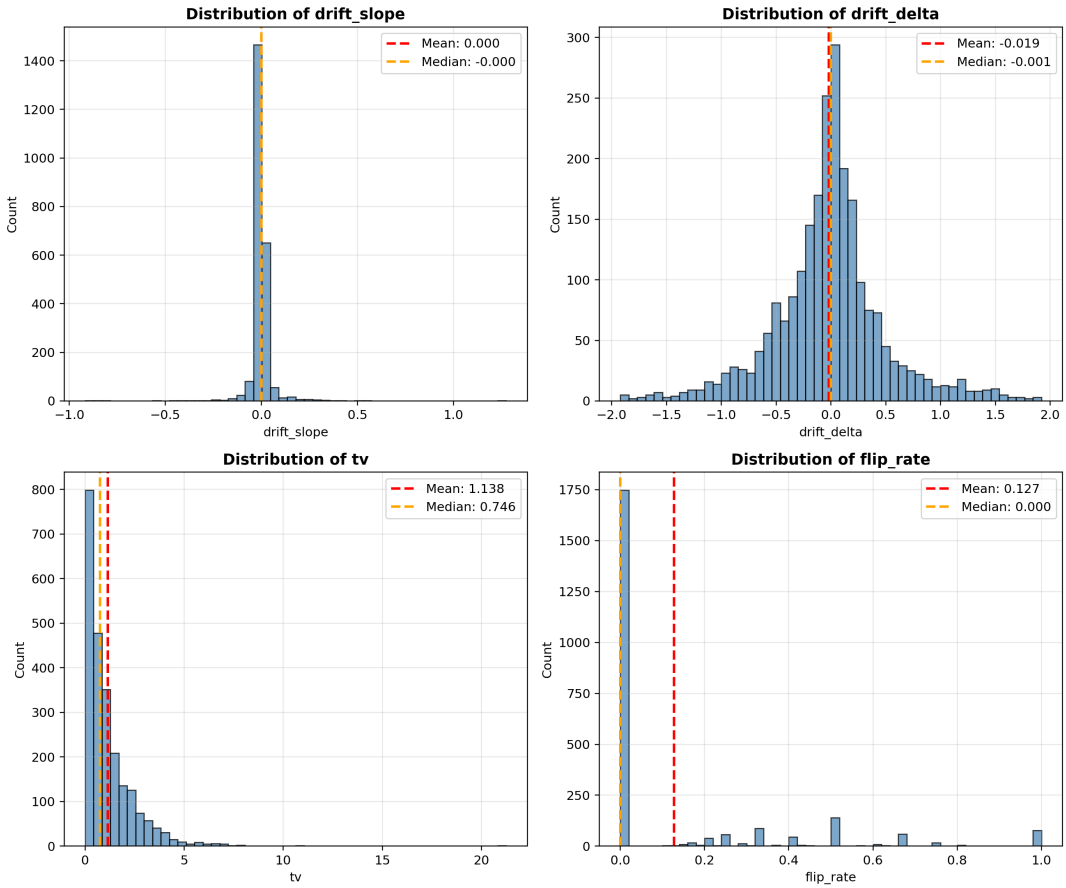

Figure 2: Distribution of drift metrics across 2,657 users. Red dashed lines indicate mean; orange lines indicate median.

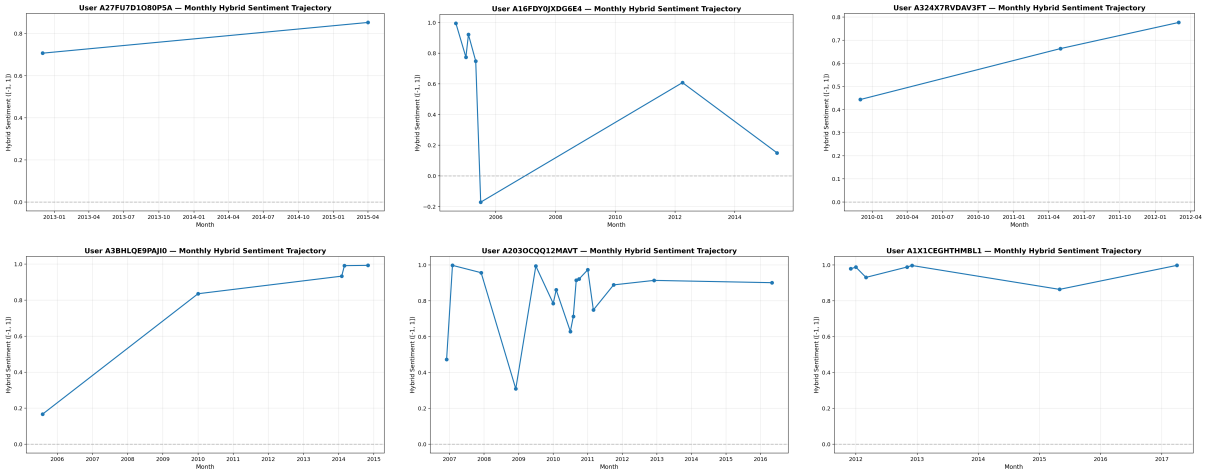

Figure 5: Representative user trajectories showing diverse drift patterns. Horizontal gray line indicates neutral sentiment (0).

5 Early Observations & Validation

5.1 Data Quality Checks

We performed sanity checks to validate pipeline correctness:

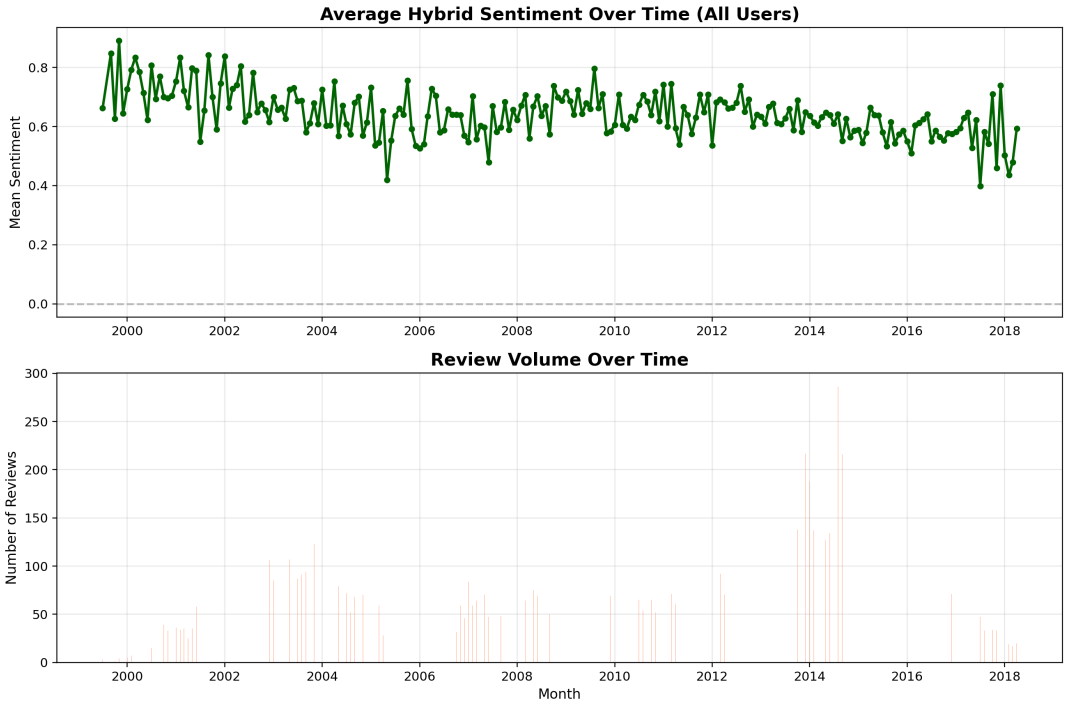

Figure 3: Top: Average hybrid sentiment over time (all users). Bottom: Monthly review volume. No strong temporal bias detected.

- **VADER polarity:** Confirmed expected behavior ("I love it" $\rightarrow$ +0.637, "I hate it" $\rightarrow -0.572$).
- **Missing data:** $< 1\%$ NaN values in hybrid sentiment after filtering.
- **Extreme values:** Manual inspection of high-drift users (slope > 0.5) revealed legitimate trajectories with dramatic opinion shifts over long timespans.
- **Date range:** Verified timestamps span expected period (1996–2018) with no anomalies.

5.2 Key Findings

1. **Heterogeneity dominates:** Users exhibit highly diverse trajectories. A one-size-fits-all model will perform poorly; personalized or cluster-based approaches are needed.
2. **Drift is real but subtle:** While 37.5% of test users show "high drift" (slope > 0.01), the majority have modest slopes (< 0.01), indicating gradual rather than abrupt changes.
3. **Volatility vs. direction are independent:** Users can have stable trends (low TV, high slope) or erratic behavior (high TV, low slope). Future models should capture both dimensions.
4. **Baseline MAE sets expectations:** Predicting per-user drift with a global mean achieves MAE ≈ 0.047 on slope. Adaptive models must significantly outperform this to justify complexity.
5. **Temporal splits are crucial:** Training on early users and testing on recent users creates a realistic evaluation scenario. Random splits would inflate performance estimates.

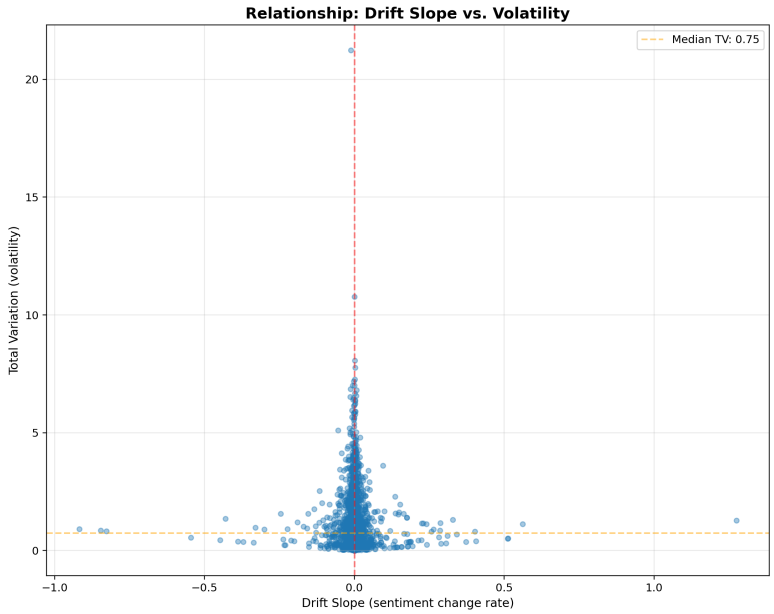

Figure 4: Scatter plot of drift slope vs. total variation. Weak correlation suggests independent dimensions of user behavior.

5.3 Limitations of Current Baseline

- **VADER simplicity:** Lexicon-based sentiment may miss context-dependent meaning (e.g., sarcasm, domain-specific jargon). Transformer models (e.g., fine-tuned BERT) could improve text representation.
- **Linear assumptions:** Computing a single slope assumes linear trends. Many users exhibit non-linear patterns (U-shapes, plateaus). Segmented regression or change-point detection would better capture these.
- **No predictive modeling:** This baseline is *descriptive* (quantifying existing drift) rather than *predictive* (forecasting future drift). Milestone 3 will build forecasting models.
- **Fixed α:** Hybrid sentiment uses α=0.7 uniformly. User-specific or item-specific α values might improve sensitivity.

6 Reproducibility & Artifact Checklist

All materials are available in the timestamped run directory:

Artifact	Description
`config.json`	Full hyperparameter configuration
`manifest.json`	Package versions, preprocessing steps
`splits.json`	Train/val/test user IDs
`baselines.json`	Training set statistics (mean slope, etc.)
`metrics_baseline.json`	Quantitative results (MAE, user counts)
`pipeline.log`	Execution log with timestamps
`README.md`	Auto-generated run documentation
`data/reviews.parquet`	Review-level features (17,706 rows)
`data/reviews_monthly.parquet`	Monthly aggregates per user
`data/user_trajectories.parquet`	Per-user drift metrics (2,657 users)
`figs/drift_distributions.png`	Metric histograms (4 panels)
`figs/temporal_trends.png`	Sentiment & volume over time
`figs/slope_vs_volatility.png`	Correlation scatter plot
`figs/user_traj_*.png`	Sample user trajectories (6 plots)

7 Next Steps & Future Work

This baseline establishes a solid foundation for Milestone 3 (adaptive modeling) and beyond:

1. **Adaptive drift detection:** Implement ADWIN and Page-Hinkley detectors to identify *when* changes occur (not just *how much*).
2. **Improved sentiment models:** Replace VADER with transformer-based classifiers (e.g., distilBERT fine-tuned on review data).
3. **User clustering:** Group users by drift pattern (e.g., k-means on [slope, TV, flip rate]) to discover behavioral archetypes.
4. **Predictive modeling:** Train models to forecast *future* drift given early reviews (e.g., LSTM on sentiment sequences).
5. **Interpretability:** Link detected change-points to stylistic shifts (word choice, n-gram frequency) for human validation.
6. **Robustness analysis:** Vary α (0.5, 0.7, 0.9) and MIN_REVIEWS (3, 5, 10) to test sensitivity.

8 Conclusion

We have successfully implemented a reproducible baseline system for ReviewMirror that processes raw review data, extracts hybrid sentiment trajectories, computes interpretable drift metrics, and organizes outputs for future experiments. Our results confirm that (1) user drift is real and heterogeneous, (2) simple global baselines perform poorly, justifying adaptive approaches, and (3) temporal evaluation splits are necessary for realistic performance estimates. This baseline will serve as the reference point for evaluating adaptive models in subsequent milestones.

References

[1] J. Ni, J. Li, J. McAuley. Amazon Review Data (2018). https://cseweb.ucsd.edu/~jmcauley/datasets/amazon_v2/.

[2] J. Gama, I. Žliobaite, A. Bifet, M. Pechenizkiy, A. Bouchachia. A Survey on Concept Drift Adaptation. *ACM Computing Surveys*, 2014.

[3] A. Bifet, R. Gavalda. Learning from Time-Changing Data with Adaptive Windowing (ADWIN). *SIAM International Conference on Data Mining*, 2007.

[4] C.J. Hutto, E. Gilbert. VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text. *ICWSM*, 2014.