Restaurant Review Insights: Hybrid Sentiment Analytics & Local LLM Summarization

A decision-support tool positioned in the RAMI 4.0 Functional Layer

Master Project Report

Submitted by
Bony Martin

Supervisor:
TBD

Hochschule Emden/Leer
M.Sc. Industrial Informatics

December 30, 2025

Abstract

Online customer reviews contain valuable operational signals, but their unstructured nature makes it hard for restaurant owners to translate feedback into concrete improvements. This project develops a reproducible analytics pipeline that combines (1) classical sentiment analysis (VADER) with (2) controlled large language model (LLM) extraction and summarization running locally via Ollama. The system transforms raw review text into structured insights across three operational areas: *Kitchen*, *Service*, and *Management*. It generates weekly owner-ready emails with evidence quotes, KPI suggestions, and prioritized actions (7-day quick wins and 30-day improvements), plus charts and structured exports (CSV/JSON).

Within RAMI 4.0, the solution is positioned as a **Functional Layer** capability that consumes review data (Information Layer) and supports managerial decision-making (Business Layer). While the case study focuses on a quick-service restaurant dataset, the architecture generalizes to competitor benchmarking and to other product- and service-based companies that rely on customer feedback.

Contents

Chapter 1

Introduction

Customer opinions are increasingly communicated through online review platforms and delivery applications. For quick-service restaurants, these reviews are one of the fastest indicators of operational performance—often faster than internal KPIs. However, review data is largely unstructured, multilingual, and noisy. Managers therefore fall back to coarse indicators (average star rating) that rarely explain *why* customers are unhappy, *where* problems occur, and *what* should be done next.

This project aims to bridge that gap by building a practical tool that turns raw reviews into an owner-oriented performance summary. The system was developed end-to-end: data ingestion, preprocessing, sentiment scoring, LLM-based fact extraction, themed summarization, visualization, and automated email delivery.

1.1 Project goals

The main goals are:

- Create a reproducible pipeline that converts unstructured reviews into structured insights.
- Combine interpretable sentiment scoring (VADER) with **controlled** LLM extraction (JSON outputs + evidence-only rules).
- Deliver a compact weekly owner email with prioritized actions and suggested KPIs.
- Ensure privacy and controllability by running the LLM locally (phi3.5 (Ollama local)).
- Provide a product-like deliverable (README, configuration, modular code, and

report).

1.2 Generalization beyond restaurants

Although the reference dataset contains restaurant reviews, the approach extends to:

- competitor benchmarking (compare strengths/weaknesses across brands)
- retail products (reviews about durability, delivery, pricing)
- SaaS products (feature requests, UX issues, support quality)
- any customer-feedback-heavy domain requiring systematic evidence-based summaries

Chapter 2

Background and Motivation

In highly competitive markets, customer satisfaction is a leading indicator of future revenue. Small service failures—long wait times, incorrect orders, hygiene issues—can quickly surface in reviews and impact reputation. However, organizations often lack tools that can systematically extract operational meaning from large volumes of text.

The motivation for this work is practical: provide a tool that a restaurant owner can run weekly, receive a clear summary, and implement changes. At the same time, the project aims to demonstrate how a hybrid approach (classical NLP + LLM) can be positioned as a *Functional Layer* capability in RAMI 4.0 for decision support.

Chapter 3

State of the Art in Customer Feedback Analytics

Early review analytics relied on manual reading or keyword counting. Automated sentiment analysis later introduced scalable polarity scoring, commonly using:

- lexicon-based models (e.g., VADER)
- machine learning classifiers (requires labeled data)
- aspect-based sentiment (associate sentiment with topics/aspects)

Lexicon methods are fast and interpretable but struggle with context and domain language. Aspect-based approaches provide more detail but often need custom topic extraction logic.

Large language models can extract meaning and summarize without task-specific training, but introduce risks:

- hallucination (invented facts)
- inconsistency across runs
- difficulty integrating outputs into structured workflows

This project uses a hybrid strategy: VADER provides stable quantitative anchors, while the LLM is used only for **controlled extraction** and **structured summarization** with strict JSON constraints and evidence quoting.

Chapter 4

Problem Definition and Use Case

4.1 Problem statement

Restaurant managers need tools that explain not only whether customers are satisfied, but *why*. Star ratings and raw review lists do not provide actionable prioritization, and manual analysis does not scale.

4.2 Use case

The case study uses German-language reviews from a quick-service restaurant dataset. The target user is the restaurant owner of Pizza House. Inputs arrive as a CSV export from review platforms. Outputs are:

- sentiment KPIs and charts
- evidence-based strengths and improvements
- 7-day quick wins and 30-day actions
- a weekly owner email with attachments

4.3 RAMI 4.0 positioning

The solution is positioned in the **Functional Layer**:

- **Information Layer:** review CSV + processed CSV/JSON outputs
- **Functional Layer:** sentiment scoring + extraction + summarization + KPI computation

- **Business Layer:** owner decisions (pricing, staffing, training, quality control)

Functionally, the tool supports condition monitoring of customer perception, diagnosis of root causes through evidence quotes, and improvement planning.

Chapter 5

System Architecture and Design

5.1 Architectural objectives

- **Modularity:** separate pipeline, email builder, and email sender.
- **Reproducibility:** per-run output folders and cached progress.
- **Interpretability:** VADER scores + evidence quotes in summaries.
- **Privacy/Control:** local LLM via Ollama.
- **Extensibility:** easy to add languages, data sources, or BI dashboards.

5.2 Architecture overview

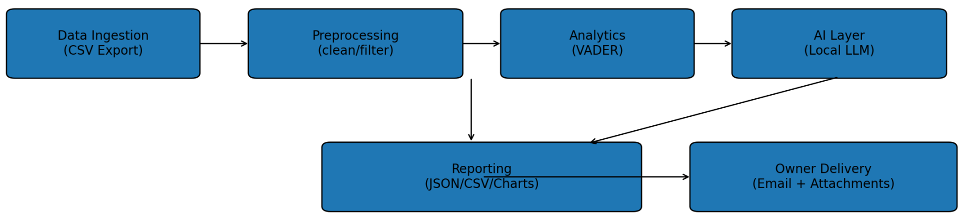

Figure 5.1: Layered architecture: ingestion → preprocessing → analytics (VADER) → AI extraction (LLM) → reporting/email

5.3 Run folder strategy (repeatability)

Each pipeline execution creates a timestamped folder: `runs/YYYY-MM-DD_HHMMSS/`. A pointer file (`runs/last_run.txt`) stores the most recent run folder so helper scripts can reliably find the correct outputs. This design avoided earlier issues where scripts read from different output directories.

5.4 Key modules

- **Main pipeline:** preprocessing, VADER, strict LLM extraction, charts.
- **Owner output builder:** transforms `owner_summary.json` into a compact email.
- **Email sender:** reads email text and sends it with attachments via SMTP.

Chapter 6

Analytics Methodology and Knowledge Discovery Process

This project follows the KDD (Knowledge Discovery in Databases) process.

6.1 KDD mapping

1. **Selection:** choose review text, ratings, timestamps from CSV.
2. **Preprocessing:** cleaning, deduplication, language filtering, date parsing.
3. **Transformation:** compute VADER scores, create evidence lines per review.
4. **Data mining:** extract operational pros/cons (Kitchen/Service/Management) via strict prompts.
5. **Interpretation:** aggregate themes; generate owner summary JSON and email report.
6. **Evaluation:** validate consistency, evidence traceability, and usefulness for decisions.

6.2 Strict prompting strategy

To minimize hallucination:

- the LLM is instructed to **extract only facts present in the review text**
- output format is forced to **JSON only**
- extraction is split by area (Kitchen/Service/Management) to reduce ambiguity
- evidence quotes are attached to each summary item

Chapter 7

Implementation and Technologies

7.1 Technology stack

- Python (pipeline orchestration)
- Pandas (CSV processing)
- NLTK + VADER (sentiment scoring)
- Matplotlib (charts)
- Ollama local LLM API (`/api/generate`) for controlled extraction
- SMTP (Gmail) for automated email delivery

7.2 Code organization

Main scripts:

- `Project_Sentiment_Analysis_22.12.1.py`: end-to-end pipeline
- `owner_outputs.py`: compact owner email builder + flat export
- `send_weekly_report.py`: SMTP sender orchestration
- `email_reporter.py`: low-level email attachment sending helper

7.3 Practical engineering challenges and fixes

During development, several real-world issues appeared:

- **Inconsistent CSV schemas:** multiple possible column names for date/rating. *Fix:* auto-detection of common alternatives and safe fallbacks.

- **Language noise and short reviews:** irrelevant rows degraded results. *Fix:* minimum text length and optional language filter (`LANGUAGE=auto|de|en`).
- **LLM instability and non-JSON replies:** models sometimes return extra text. *Fix:* strict prompts, `format=json`, and robust JSON parsing with fallback extraction.
- **Interrupted runs:** long LLM calls can stop mid-way. *Fix:* caching (`cache_reviews.csv`) and resume logic.
- **Duplicate output files:** earlier versions produced multiple legacy copies. *Fix:* canonical filenames only (single set per run folder) + consistent attachment list.

7.4 Configuration via `.env`

The application is configured using a simple `.env` file:

```
LLM_URL=http://localhost:11434/api/generate
MODEL_NAME=phi3.5
RESTAURANT_NAME=Pizza House
INPUT_CSV=sample_data/Burger King Data.csv

SMTP_HOST=smtp.gmail.com
SMTP_PORT=587
SMTP_USER=...
SMTP_PASS=... # Gmail app password
OWNER_EMAIL=...
```

Chapter 8

Evaluation and Results

8.1 Evaluation approach

Evaluation focuses on decision-support value:

- interpretability: are insights traceable to evidence quotes?
- consistency: do repeated runs on the same data produce stable outputs?
- usefulness: are suggested actions and KPIs meaningful for an owner?

8.2 Observed outputs

The pipeline produces:

- sentiment distributions (bucket counts and star distribution)
- a trend chart (monthly VADER average when dates are available)
- structured owner summary with strengths and prioritized improvements

8.3 Example insight categories

Across runs, the system consistently organized feedback into:

- **Kitchen:** quality consistency, temperature, freshness.
- **Service:** speed, friendliness, order accuracy, cleanliness.
- **Management:** pricing transparency, ambience, staffing levels.

Chapter 9

Discussion

9.1 What worked well

- Hybrid design anchors qualitative summaries in quantitative sentiment KPIs.
- Evidence quotes increase trust and reduce misinterpretation.
- Local LLM deployment improves privacy and operational controllability.

9.2 Limitations

- VADER can miss sarcasm and domain-specific nuances.
- LLM quality depends on the selected model and available compute.
- Batch design is ideal for weekly reporting but not real-time monitoring.

9.3 Extension opportunities

- competitor benchmarking dashboard (multi-restaurant comparison)
- multilingual support (DE/EN auto summaries)
- integration with BI tools (Grafana/Power BI) using the flat CSV output

Chapter 10

Conclusion and Future Work

This project delivered a product-like analytics tool that turns raw customer reviews into owner-ready insights. By combining VADER sentiment scoring with controlled local LLM extraction and strict evidence rules, the solution achieves interpretability and practical decision support. The system aligns with RAMI 4.0 as a Functional Layer analytics capability and can generalize beyond restaurants to other customer-feedback-driven domains.

Future work should focus on multi-entity benchmarking, real-time ingestion, and systematic user evaluation with restaurant managers.

Appendix A

Appendix

A.1 End-to-end workflow

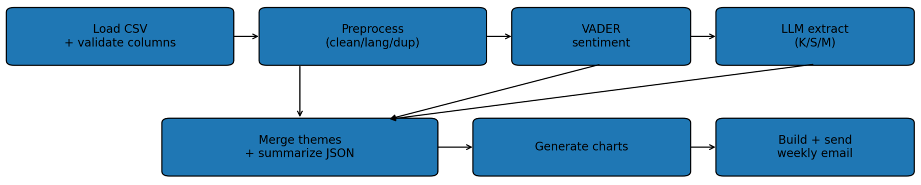

Figure A.1: Pipeline flow: load data → preprocess → VADER → LLM extraction → summary → charts → email

A.2 Suggested repository structure

```
restaurant-review-insights/
  Project_Sentiment_Analysis_22.12.1.py
  owner_outputs.py
  send_weekly_report.py
  email_reporter.py
  README.md
  requirements.txt
  .env.example
  sample_data/
    Burger King Data.csv
```

```
runs/
  (generated per execution)
report/
  report.tex
  architecture.png
  flowchart.png
```

Bibliography

[1] Hutto, C. J., & Gilbert, E. (2014). VADER: A Parsimonious Rule-Based Model for Sentiment Analysis of Social Media Text.

[2] Jurafsky, D., & Martin, J. H. Speech and Language Processing.

[3] Brown, T. et al. (2020). Language Models are Few-Shot Learners.