Reflection on the NordTech ETL Pipeline Project 👌

Project Overview ✨

This project involved designing and implementing an end-to-end ETL pipeline for NordTech order data, covering data extraction, exploratory analysis, transformation, sentiment enrichment, and loading into a relational database.
The primary objective was to transform raw, inconsistent transactional data into a reliable, analysis-ready dataset that supports KPI reporting and business insight generation.

The project was approached with a strong emphasis on modularity, reproducibility, and robustness to future data variations.

Architectural Decisions 🏗️

Modular ETL Design

The pipeline was intentionally split into separate modules (`extract`, `transform`, `load`, and `sentiment`) to enforce separation of concerns. This structure improves maintainability, simplifies debugging, and allows individual components to be reused or extended without affecting the entire system.
Such modularity reflects common industry practices in production ETL systems.

Configuration values such as file paths and database locations were centralized in a configuration module to ensure reproducibility and reduce hard-coded dependencies.

Data Quality Challenges and Solutions 🔍

Handling Real-World Messy Data

The raw dataset contained numerous real-world data quality issues, including:

- Mixed and inconsistent date formats, including Swedish month names
- Textual price values with currency symbols and formatting variations
- Inconsistent categorical labels caused by casing, abbreviations, and synonyms
- Duplicate records
- Invalid logical relationships (e.g. delivery dates preceding order dates)

Addressing these issues required designing robust, defensive cleaning functions that could tolerate malformed input while preserving as much valid information as possible.

Date Normalization 🗓️

Dates were parsed using flexible parsing logic to handle multiple formats and languages.
Additional validation was applied to detect and correct reversed order and delivery dates, ensuring temporal consistency across the dataset.

Quantity (`antal`) Imputation 🔢

The quantity column contained numeric values embedded in text, unit annotations, and Swedish number words.
After normalization, missing quantities were imputed with a value of **1**, based on the business assumption that an order line without an explicit quantity typically represents a single purchased unit.
Using 0 would incorrectly remove revenue contribution, while higher values would inflate sales metrics.

Customer Ratings (`betyg`) Handling ⭐

Ratings were validated to ensure they fell within the valid range of 1–5.
Missing values were imputed using the **median** rather than the mean, as ratings are ordinal rather than continuous.
The median preserves the central tendency without introducing unrealistic decimal values or being skewed by extreme ratings.

Missing Value Strategy 🧩

Missing values were handled based on semantic meaning rather than a single global rule:

- Geographic fields (e.g. region) were kept as null to avoid creating artificial categories.
- Transactional categorical fields (e.g. payment method) used an explicit `"unknown"` category to preserve row completeness for aggregation.
- Textual placeholders such as `"nan"` in review text were converted to true missing values to prevent contamination of NLP analysis.

Grain and Aggregation 🧠

The grain of the cleaned dataset is **one row per order line**, not one row per order.
This distinction is critical because a single order may contain multiple products, quantities, and prices.

As a result, all KPI calculations required careful aggregation logic.
Revenue-based KPIs were calculated using quantity-adjusted line prices, while order-level metrics required grouping by `order_id`.
Understanding the dataset grain was also essential when handling duplicates, as full-row duplicates at the order-line level would otherwise inflate revenue and volume metrics.

This awareness directly influenced how KPIs were designed and ensured that insights were based on correct business logic.

Sentiment Analysis Integration 🤖

A multilingual BERT model was selected for sentiment analysis due to its contextual understanding and support for both Swedish and English text.
Compared to lexicon-based approaches, BERT provides more reliable sentiment classification for short, informal customer reviews.

BERT was chosen over larger LLMs due to its ability to run locally, predictable inference behavior, and lower operational complexity.
However, this choice comes with trade-offs, including higher computational cost compared to simpler models and slower inference speed when applied row-by-row.

Sentiment analysis was intentionally kept separate from the core ETL pipeline to preserve pipeline purity and allow sentiment enrichment to be recomputed or replaced without reprocessing the entire dataset.

KPI Insights and Business Value 📈

After cleaning and standardization, the dataset enabled reliable KPI computation, including delivery performance, return rates, revenue per region, and customer satisfaction metrics.

Several actionable insights emerged:

- Certain regions showed higher return rates, indicating potential logistics or fulfillment issues.
- Delivery delays were more common for specific delivery statuses and payment methods.
- In some cases, numerical ratings were high while sentiment was neutral or negative, suggesting that ratings alone do not fully capture customer experience.

These insights allow NordTech to identify operational bottlenecks, improve delivery processes, and prioritize customer experience improvements based on data rather than assumptions.

Customer Insights from Reviews 💬

Sentiment analysis of customer reviews revealed patterns that are not visible through numerical ratings alone.
Negative sentiment frequently related to delivery delays, damaged products, or unmet expectations, while positive sentiment often emphasized fast delivery and product quality.

The presence of neutral or negative sentiment alongside high ratings suggests that customers may still report issues even when assigning acceptable scores.
By monitoring sentiment trends, NordTech can proactively address recurring complaints and improve communication, delivery reliability, and post-purchase support to increase customer satisfaction.

Database Integration and Reproducibility 🗄️

The cleaned dataset was loaded into a local SQLite database, enabling downstream analysis to be performed directly on validated, standardized data rather than raw CSV files.
Validation data was appended rather than overwritten to simulate incremental data ingestion and ensure the pipeline remained robust when exposed to new data.

Reading from SQL in the KPI notebook reinforced the separation between data preparation and analysis layers.

Limitations and Future Improvements 🚀

With additional time, the pipeline could be further improved by:

- Implementing batch sentiment inference for performance optimization
- Adding structured logging instead of print-based debugging
- Writing unit tests for individual transformation functions
- Introducing a workflow orchestrator such as Prefect or Airflow

These enhancements would move the pipeline closer to production-grade standards.

Conclusion 🎯

This project reinforced the importance of thoughtful data preparation as the foundation of reliable analytics.
Designing a robust ETL pipeline required not only technical implementation but also careful consideration of business assumptions, dataset grain, and downstream use cases.

Overall, the project significantly strengthened my ability to design maintainable data pipelines and translate messy real-world data into meaningful business insights.