Formatted Income Statements from Generated Synthetic Data- Approach Document

Document Purpose: To outline the technical strategy, architectural decisions, and roadmap for Creating 5 formats of income statements for Companies with different industries and income brackets with Synthetic Data-Gen.

Document Metadata

1. **Author:** Dhruvv Raghu
2. **Reviewers:** Poornachandra
3. **Date:** Monday, 29/12/2025
4. **Version:** 1.0

Current Baseline (State of the Union)

1. **Technology Stack:** Python, Torch, Unsloth, Transformers (All the supporting libraries for its usage such as BitsAndBytes, trl and so on)
2. **Current Baseline:** Using Llama 3.3 8B for its proficiency with JSON Formatting and responses in Inference.

3. Your approach

Approach 1: Adopt Automation for the procedure.

Approach 2: LLM generates synthetic data, and upon this data (JSON Formatted Response) we build plots and representations of data visually.

[Agent Name 1]: LLama3.3 8B Model

1. **[Agent Name 2]:** (e.g., Prompt Generator) – Logic for dynamic instruction building.
2. **[Agent Name 3]:** (e.g., Analysis/OCR) – Orchestration of external tools and data extraction.

5. Strategic Roadmap (Phased Approach)

Phase 1: Data Unification & Deterministic Logic

- **Goal:** Establish a "Single Source of Truth" and generate baseline financial structures.

- **Intuition:** AI requires high-quality, structured training data. Raw financial data is currently siloed across multiple datasets (Market Data, P&L, Balance Sheets), preventing holistic analysis. We must merge these and apply industry-specific logic (e.g., "SaaS" vs. "Retail") to create a reliable ground truth for training.
- **Key Tasks:**
    - **Data Ingestion:** Merge disparate CSVs (Price, Metrics, Balance Sheet, P&L) into a unified Master Financial Record using common keys (BSE Code).
    - **Logic Mapping:** Implement deterministic Python functions to map generic accounting columns to industry-specific formats (e.g., deriving "Prime Cost" for Hospitality).
    - **Baseline Generation:** Batch-process the Master Record to output standardized CSVs and structured text reports for 5 distinct industries.

Phase 2: Synthetic Intelligence (Unsloth Fine-Tuning)

- **Goal:** Transition from rigid rule-based logic to probabilistic generative modeling.
- **Intuition:** Deterministic scripts cannot simulate complex, non-linear financial scenarios or "hallucinate" realistic future projections. By fine-tuning a Large Language Model (Llama-3) on the Phase 1 data, we create a system that understands the *relationships* between financial line items (e.g., Revenue $\leftrightarrow$ COGS) rather than just following hardcoded instructions.
- **Key Tasks:**
    - **Dataset Prep:** Convert Phase 1 CSV outputs into an Alpaca-style JSONL format (`Instruction` $\rightarrow$ `JSON Output`).
    - **Model Training:** Fine-tune Llama-3-8B using Unsloth (QLoRA) to learn the specific JSON schemas and number distributions of the 5 industry formats.
    - **Validation:** Verify the model's ability to output valid JSON structures without input context (zero-shot generation).

Phase 3: Inference, Visualization & Robustness

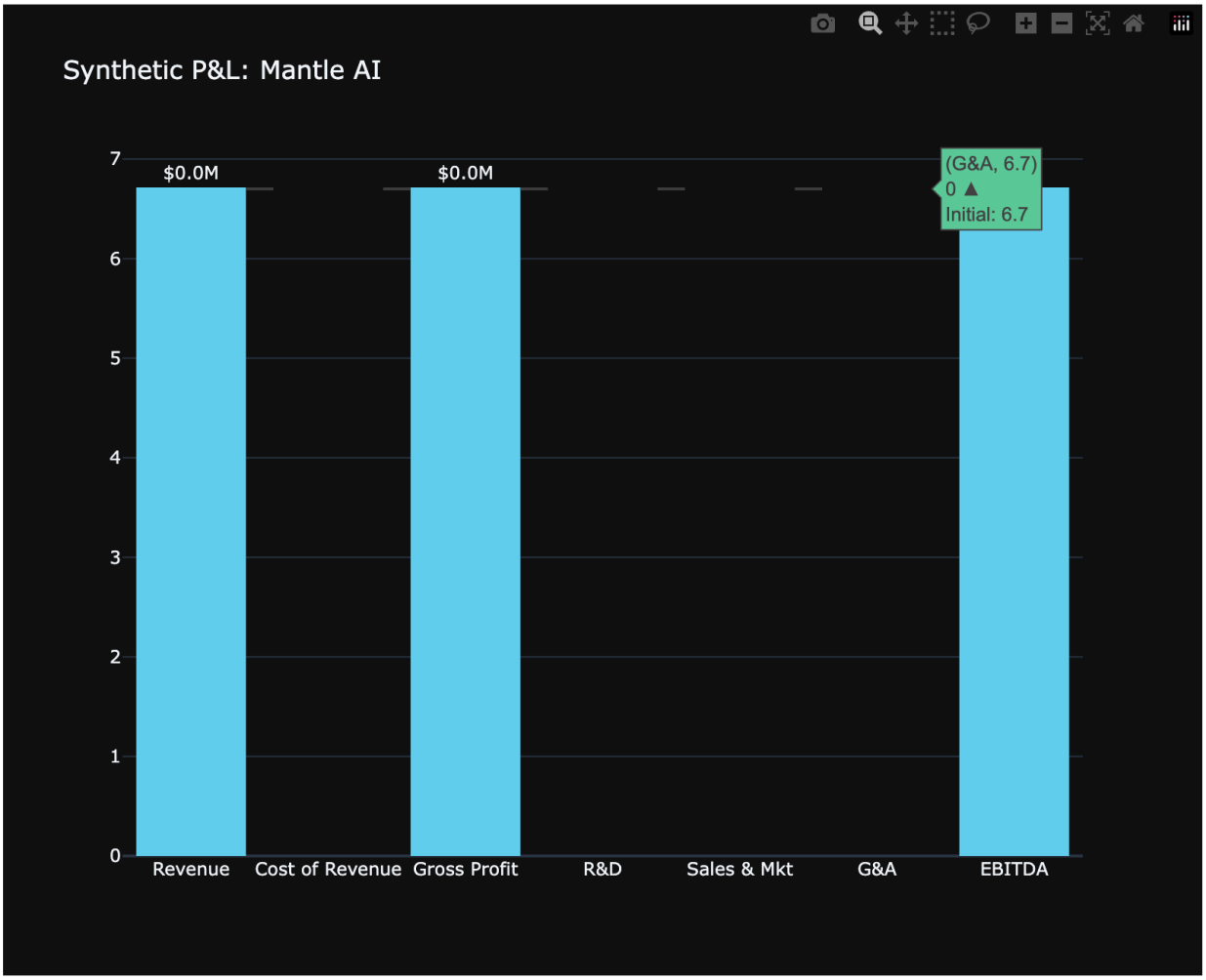
- **Goal:** Deploy a resilient Analysis Engine for dynamic reporting and visualization.
- **Intuition:** Generative models are powerful but prone to variance (e.g., aliasing keys like `Sales` vs `Revenue`). A raw JSON output is insufficient for end-users. We need a robust "Translation Layer" that parses imperfect LLM outputs and dynamically renders them into professional visual insights.
- **Key Tasks:**
    - **Robust Inference:** Implement a "Soft-Parsing" engine with Regex fallbacks to handle JSON errors or key variations from the LLM.
    - **Dynamic Visualization:** Build adaptive Plotly Waterfall charts that automatically detect available line items (e.g., "Detailed OpEx" vs. "Total OpEx") and render the correct flow.

6. Success Criteria & Metrics

1. **Performance:** Training Loss, ability of model to generalise to rows of data.
2. **Quality:** LLM generates data without key errors, is able to generate plots. Make sure that even if LLM generates remotely the same words to describe key, regex soft parsing covers the plotting

Some of the data I generated during this process:

```
================================================
INFOSYS (MOCK SAAS)
SaaS Income Statement
Period Ending: 2025-01-01
================================================
Line Item                      |       Amount
------------------------------------------------
Revenue                         | $15,000,000.00
Cost of Revenue (Support)          | ($3,000,000.00)
------------------------------------------------
Gross Profit                    | $12,000,000.00
Operating Expenses (R&D Proxy)      | ($8,000,000.00)
------------------------------------------------
EBITDA                          | $4,000,000.00
Net Income                       | $3,500,000.00
================================================
```



🤖 Generating Synthetic Financials...

✅ Successfully Parsed JSON:


```
{
  "Company": "Mantle AI",
  "Recurring_Revenue": 6.7,
  "Contribution_Margin": 3.7,
  "Corporate_Overheads": 3.3,
  "Net_Income": 0.6,
  "Weighted_Average_Shares_Outstanding": 8.1,
  "Revenue_Growth": 1.3,
  "Contribution_Margin_Growth": 1.3,
  "Net_Income_Growth": 1.3,
  "Free_Cash_Flow": 1.2,
  "Adjusted_Free_Cash_Flow": 1.2,
  "Gross_Margin": 5.4
}
```