Kasparro Applied AI Engineer Assignment — Agentic Facebook Performance Analyst

You are the Applied AI Engineer at Kasparro, building the next generation of Agentic Marketing Analysts. Your task is to design a multi-agent system that autonomously diagnoses Facebook Ads performance, identifies reasons behind ROAS fluctuation, and recommends new creative directions using both quantitative signals and creative messaging data.

Objective

Build a self-directed Agentic System that can:
• Diagnose why ROAS changed over time.
• Identify drivers behind changes (e.g., audience fatigue, creative underperformance).
• Propose new creative ideas (headlines, messages, CTAs) for low-CTR campaigns, grounded in dataset's existing creative messaging.

Data Provided

Synthetic eCommerce + Facebook Ads dataset with columns including:
campaign_name, adset_name, date, spend, impressions, clicks, ctr, purchases, revenue, roas, creative_type, creative_message, audience_type, platform, and country.

Agent Design Requirements

The system must include:
• Planner Agent — Decomposes user query into subtasks.
• Data Agent — Loads and summarizes dataset.
• Insight Agent — Generates hypotheses explaining patterns.
• Evaluator Agent — Validates hypotheses quantitatively.
• Creative Improvement Generator — Produces new creative messages for low-CTR campaigns.

Prompt Design Guidelines

Prompts should be structured and layered, not one-line instructions.
• Specify format expectations (JSON schema, Markdown).
• Include reasoning structure (Think → Analyze → Conclude).
• Use data summaries instead of full CSV input.
• Include reflection/retry logic for low-confidence results.

Expected Deliverables

File	Description
agent_graph.md	Diagram + explanation of agent roles & data flow.
run.py	Main orchestration script (CLI: python run.py 'Analyze ROAS drop').
insights.json	Structured output of hypotheses, confidence, and evidence.
creatives.json	Generated creative recommendations for low-CTR campaigns.
report.md	Final summarized report for marketer.
logs/	Structured logs (JSON or Langfuse traces).

Evaluation Rubric

- Agentic reasoning architecture (30%) — Clear Planner–Evaluator loop.
- Insight quality (25%) — Grounded hypotheses and reasoning clarity.
- Validation layer (20%) — Quantitative checks and confidence logic.
- Prompt design robustness (15%) — Structured, reusable, reflective.
- Creative recommendations (10%) — Contextual, data-driven, diverse ideas.

Timebox

8–10 hours (focus on reasoning & robustness; no UI required).

Optional

Implement short-term memory of insights across runs for iterative learning.

Submission & GitHub Instructions (Mandatory)

Submit a single public GitHub repository link. Use the following conventions to make evaluation fast and fair.

1) Repository Name: kasparro-agentic-fb-analyst-<firstname-lastname>

2) Required Structure:
README.md — Setup, data path, commands, architecture diagram.
requirements.txt — Pinned versions.
config/config.yaml — Thresholds, paths, seeds.
src/ — All code; separate /agents, /orchestrator, /utils.
prompts/ — Prompt files (.md/.txt).
data/ — Small sample dataset + data/README.md.

logs/ — JSON logs or Langfuse traces.
reports/ — report.md, insights.json, creatives.json.
tests/ — evaluator tests.
Makefile or run.sh — setup, run, test, lint.

3) README.md Must Contain:
Quick start, data instructions, exact CLI command, diagram, validation description, example outputs.

4) Reproducibility:
Seed randomness, pin versions, provide small sample dataset, config flag for full/sample switch.

5) Evidence & Observability:
Commit insights.json, creatives.json, report.md, and logs/Langfuse evidence.

6) Git Hygiene:
At least 3 commits, a v1.0 release tag, and a PR titled 'self-review' describing design choices.

7) How to Submit:
Share one link: the public GitHub repo URL, plus commit hash, release tag, and command used to produce outputs.