InsightCore Agent – Behavioral Pattern

& Action Dashboard for the AI Assistant

- Snehal Kadam

1. Executive Summary

The **InsightCore Agent** is a lightweight behavioral analytics solution designed to track, analyze, and optimize user task performance. It records user events (completed, skipped, repeated), identifies productivity patterns, and generates actionable insights. By analyzing task data, the system highlights peak productivity windows, completion rates, and recurring bottlenecks.

The prototype is designed to be compact, easily demonstrable, and export-ready, enabling organizations to quickly showcase its value in enhancing productivity and task management.

2. Deliverables

The InsightCore Agent package includes:

- **Code modules**: data_logger.py, insights_generator.py
- **Dataset**: user_behavior_log.csv (sample user activity log)
- **Automated outputs**: insights_report.csv and insights.json
- **Demo video**: A short 2–3 minute walkthrough of the system
- **Documentation**: README file and this corporate-style summary report

3. System Architecture

The architecture is modular and designed for clarity:

Workflow:
User Events → Data Logger → Event Log (CSV) → Insights Generator → Reports (CSV/JSON) → Visualization Dashboard

Components:

1. **Data Logger (data_logger.py)** — Logs user tasks and appends them to the event file.

2. **Event Log (user_behavior_log.csv)** — Stores structured user activity (timestamp, task type, action, duration).

3. **Insight Generator (insights_generator.py)** — Processes the CSV log, computes metrics, and exports results.

4. **Reports & Dashboard** — Outputs (insights_report.csv, insights.json) are available for analysis or visualization in Excel/Power BI.

4. Key Metrics & Rule Logic

- **Completion Rate** = Completed ÷ (Completed + Skipped)
- **Best Time Window** — Day is split into blocks (morning, afternoon, evening, night); the slot with the highest completion rate is identified.
- **Top Task Types** — Ranking of most frequently completed tasks.
- **Reward Scoring** — +1 (on-time), 0 (late), -1 (skipped).
- **Suggestions Engine** (rule-based):
    - Recommend rescheduling tasks into more productive time windows.
    - Suggest splitting tasks into smaller subtasks if skipped repeatedly.
    - Adjust task frequency when persistent drop-offs are detected.

5. How to Run

- Place user_behavior_log.csv into the project folder.
- Run the following command:

  ```
  python insights_generator.py
  --input user_behavior_log.csv --output insights_report.csv --json insights.json
  ```

- The script will display a summary in the terminal and generate CSV/JSON files.
- Outputs can be visualized in Excel or Power BI.

6. Demo Video Storyboard

- **00:00–00:10** — Title slide: “InsightCore Agent — Demo”
- **00:10–00:40** — Present user_behavior_log.csv and explain data fields
- **00:40–01:10** — Run insights_generator.py and show terminal output
- **01:10–01:40** — Display insights_report.csv and highlight results
- **01:40–02:10** — Show visualization charts (completion by time window, top tasks)
- **02:10–02:30** — Conclude with recommendations and next steps

7. Challenges & Solutions

- **Sparse data** → Applied minimum thresholds to avoid misleading insights
- **Ambiguous task naming** → Standardized using mapping and metadata normalization
- **Timezone inconsistencies** → Adopted ISO timestamps with offsets for consistency

8. Visualization Suggestions

- **Line Chart** — Weekly trend of productivity score
- **Table** — Top 3 positive habits vs top 3 drop-off areas

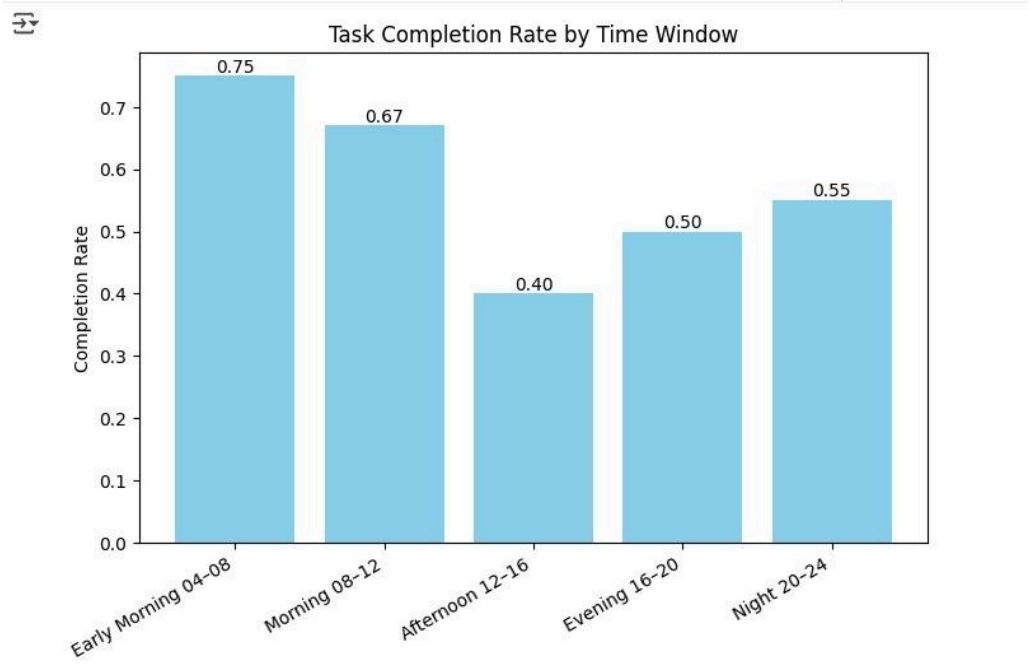
9. Outputs

Metrics (from test dataset):

Metric	Value	Details
Overall Completion Rate	0.67	Completed: 8, Skipped: 4
Best Time Window	Morning (08:00–12:00)	Completion Rate: 78%
Top Tasks	Exercise, Study, Emails	Based on frequency of completions

Example Suggestions:

- "Schedule focus tasks between 08:00–12:00 (highest completion rate)."
- "Break study into smaller sessions to reduce repeated skips."



10. Next Steps

- Expand data ingestion to databases (SQLite) for scalability.
- Implement feedback loop (thumbs up/down) to refine suggestions.
- Add real-time dashboards with automated visualizations.
- Explore reinforcement learning for adaptive task scheduling.