TARPON AUTOMATION EXCHANGE WITH UNIV OF UTAH

AI INTEGRATION WITH ANALYTICS

Jared Fluhman, Jed Nigro, Ken Briercheck, Martin Smith
Analytics and IT, Advanced Analytics and AI, DevOps

HEALTH
UNIVERSITY OF UTAH

MONTH 00, YEAR

TEAM HIGHLIGHTS

DEVOPS, DATA SCIENCE, ADVANCED ANALYTICS/AI

- Highly skilled, lean team: Four employees with Ken and Jed as working leaders.
- Self-taught: Udemy, YouTube, AI and other resources.
- Open Source: postgres, dagster, redis, django, podman, etc.
- Built a foundation on the web, supporting all our applications. Allows for rapid development and beta testing.

PROBLEM AND PROPOSED SOLUTION

Problem:

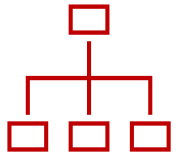
Our Business Intelligence Analysts spend hours preparing updates for monthly meetings with their assigned departments. They reference many dashboards and reports reviewing specific KPIs and metrics. After this review, they prepare a summary of the department's current financial status. This is a time-consuming affair that seems to be a great use-case for AI/LLM.

Solution:

Leverage AI/LLMs and other coding to create a single view summarizing all metrics for the specific department or division needing review.

KEY POINTS

ADDITIONAL PROBLEM CONTEXT



Data often needs to be filtered by:

Provider Group

Specialty Place of Service

(and other organizational dimensions)

Data isolated and stored disparately

Additionally, traditional dashboards require users to interpret trends themselves.

GOAL: Generate objective executive summaries directly from the data

OVERALL PROCESSING PIPELINE

FILTER TO CACHING PROCESS

1. USER SELECTS FILTERS

User selects Group, Division, Provider, Place of Service, etc.

2. RETRIEVE FILTERED DATASETS

Data is retrieved based on user selected filters.

3. TRANSFORM & ANALYZE DATA

Time series analysis functions are performed on data, including forecasting

4. SERIALIZE TO JSON

Output of data transformation is serialized for ease of passing to LLM

5. DASHBOARD GENERATION

Plotly leveraged to generate salient dashboards per metric

6. REVENUE STREAM LLM SUMMARIES

A composable, extensible LLM application consumes the JSON and streams a summary

7. EXECUTIVE SUMMARY

All summaries are compiled and reprocessed via LLM to generate an executive summary

8. REDIS CACHE

Redis cache allows for rapid retrieval of previously executed processes

WHY WE PROCESSED THE DATA

DETERMINISTIC DATA BEFORE AI

Instead of sending raw tables to the LLM...

- **We first computed:**
  - Time series analysis
  - Forecasting
  - Variance calculations
  - Trend detection
  - Business Metrics
- Then serialized everything into structured JSON.

- **Benefits**
  - Smaller prompts
  - Lower token usage
  - Faster Execution
  - Eliminated LLM mathematical errors
  - Consistent Inputs

MULTI-STAGE AI SUMMARIZATION

DIAGRAM OF SUMMARY FLOW TO EXECUTIVE SUMMARY

MULTI-STAGE AI SUMMARIZATION

REASONING BEHIND MULTI-STAGE AI SUMMARIZATION

- **Instead of asking one LLM to summarize all data:**
  - Each Revenue stream is summarized independently
  - Domain summaries are combined
  - Final executive summary is generated from those summaries

- **Benefits**
  - Better Context
  - More consistent output
  - Smaller Prompts
  - Modular architecture

ENGINEERING THE PROMPTS

PROMPT ENGINEERING WAS THE LARGEST INVESTMENT

- **RELIABILITY**
  - Neutral Language
  - No unsupported conclusions
  - Consistent formatting
- **ACCURACY**
  - Structured JSON
  - Fixed context
  - Deterministic calculations
- **PERFORMANCE**
  - Compact Prompts
  - Reduced tokens
  - Faster responses

```
"Payments lag charges by about 30 days, so a strong charge month typically shows up in payments the following mo
"Write for operational and executive leadership using concise business language.\n\n"

"Output rules:\n"
"- Use Markdown only.\n"
"- Never output HTML tags.\n"
"- Do not use Markdown headings (#, ##, ###), tables, code fences, blockquotes, or strikethrough.\n"
"- Start with one short plain-text title on its own line.\n"
"- Insert one blank line after the title before the first section header.\n"
"- Then use these section headers in this exact order: **Summary**, **Charges Summary**, **Payments Summary**, *
"- Put each section header on its own line.\n"
"- Under each section, write 2-5 bullets.\n"
"- Every bullet must begin with '- ' and be on its own line.\n"
"- Do not collapse bullets into paragraph text.\n"
"- Do not use em dashes or en dashes; use commas, periods, or plain hyphens instead.\n"
"- Keep the full summary concise and decision-oriented.\n\n"

"Content guidance:\n"
"- In **Summary**, give a short cross-metric read-through across charges, payments, WRVUs, and procedure quantit
"- In **Summary**, note the payment lag relationship when it helps explain the current month.\n"
"- In each metric section, describe the latest month, how it compares with the 12-month average, year over year
"- Mention maximum and minimum months when that adds useful context.\n"
"- If a value is missing, skip it rather than inventing content.\n\n"

"Example format:\n"
"Financial Performance Snapshot\n"
"\n"
"**Summary**\n"
"- Summarize the overall operating read-through across production, payments, productivity, and volume using the
"- Note the payment lag relationship when it helps explain the current month.\n"
"\n"
"**Charges Summary**\n"
"- Latest month charges were [value] and were [above/below/near] the 12-month average.\n"
```

ARCHITECTURE

OPEN-SOURCE STACK (OPEN AI AS AN EXCEPTION)

Django
Open-source Python based web framework used to host the application.

PostgreSQL
Open-source database used to store data.

REDIS
Open-source, in-memory NoSQL data structure store used to cache LLM summaries and HTML pages.

PLOTLY
Open-source graphing library used to present visualizations.

OLLAMA (beta):
Open-source LLM hosting service used to stream data summaries.

OpenAI (production):
Cloud-based, LLM hosting service used to stream data summaries in production.

Brief Demonstration of Application

TRADEOFFS & LESSONS LEARNED

Decision	Benefit	Cost
Dynamic filtering	Unlimited flexibility	Higher latency
No cache pre-warming	Any filter combination	Cache generated on demand
JSON preprocessing	Better accuracy	More backend processing
Summary-of-summaries	Better executive output	Multiple LLM calls

RATIONALE:
We consistently favored correctness and flexibility over raw response time.

THANK YOU!

jared.fluhman@hsc.utah.edu
jed.nigro@hsc.utah.edu
martin.smith@hsc.utah.edu