**Holiday Planner App: Metrics Analysis**

**Metrics Being Logged**

**1. Telemetry Metrics (via telemetry.py)**

**Source**: app.py logs telemetry data for all three variants (A, B, C)

**Fields Logged**:

* ts - Timestamp (ISO format)
* session\_id - Unique session identifier
* variant - App variant (A, B, or C)
* task - Task type (same as variant code: A, B, or C)
* latency\_ms - Response time in milliseconds
* tokens\_user - Input tokens from user
* tokens\_total - Total tokens (input + output)
* cost\_usd - Calculated cost in USD

**Cost Formula**:

# Claude Haiku 3.5 pricing

input\_cost = (input\_tokens / 1\_000\_000) \* 0.80

output\_cost = (output\_tokens / 1\_000\_000) \* 4.00

total\_cost = input\_cost + output\_cost

**2. Success Metrics (via success.py)**

**Source**: app.py calls log\_success\_metrics() when users provide feedback

**Fields Logged**:

* ts - Timestamp
* session\_id - Session identifier
* variant - App variant
* task - Task identifier
* user\_satisfied - Boolean user satisfaction (True/False/None)
* auto\_eval\_score - Automated evaluation score (0-1)
* response\_text - Full LLM response
* keyphrases - Expected keyphrases as JSON

**Auto-Evaluation Formula**:

# Keyphrase coverage scoring

hits = sum(1 for phrase in expected\_keyphrases if phrase.lower() in response\_text.lower())

score = hits / max(1, len(expected\_keyphrases))

# Enhanced evaluation (if metrics.py available):

# - Keyphrase coverage (0-1)

# - Response length (word count)

# - Has structure (Day 1, Morning, etc.)

# - Has specifics (prices, times, durations)

# - Overall score = (coverage + 0.1\*structure + 0.1\*specifics) / 1.2

**3. Evaluation Metrics (via run\_evals.py)**

**Source**: Standalone evaluation script

**Fields Logged**:

* id - Task identifier
* latency\_ms - Processing time
* output - Mock model output

**Dashboard Metrics Display**

**Overview Metrics (Top Row)**

1. **Total Requests** - COUNT(\*) FROM telemetry
2. **Average Latency** - AVG(latency\_ms) in milliseconds
3. **Total Tokens** - SUM(tokens\_total)
4. **Total Cost** - SUM(cost\_usd) in USD
5. **User Satisfaction** - (satisfied\_count / total\_feedback) \* 100%

**Performance Tab**

* **Latency Distribution** - Histogram of response times
* **Token Usage Over Time** - Daily aggregated token consumption
* **Daily Cost Trends** - Cost accumulation over time

**Variant Comparison Tab**

* **Average Latency by Variant** - Performance comparison across A/B/C
* **Usage Distribution** - Pie chart of variant popularity
* **Variant Statistics Table**:
  + Average latency ± standard deviation
  + Average tokens per request
  + Average cost per request
  + Usage count

**Task Analysis Tab**

* **Task Performance Metrics** - Latency, tokens, cost by task type
* **Task Usage Distribution** - Which tasks are most popular
* **Cross-Analysis** - Variant × Task performance matrix

**Success Metrics Tab**

* **Total Feedback Count** - Responses with user feedback
* **Satisfaction Rate** - Percentage of positive feedback
* **Auto-Evaluation Score** - Average automated quality score
* **Satisfaction by Variant** - Which variants perform better
* **Auto-Eval by Task** - Quality scores across different tasks
* **Recent Feedback Table** - Latest user satisfaction data

**Key Insights Available**

**Performance Insights**

* **Latency Patterns**: Identify slow variants/tasks for optimization
* **Cost Efficiency**: Which variants provide best value (quality/cost ratio)
* **Usage Trends**: Popular features and user preferences
* **Scalability**: Token consumption growth patterns

**Quality Insights**

* **User Satisfaction**: Real user feedback on helpfulness
* **Content Quality**: Automated scoring based on expected elements
* **Variant Effectiveness**: A/B/C testing results
* **Task-Specific Performance**: Which tasks work best with which variants

**Business Insights**

* **Feature Adoption**: Usage distribution across variants
* **Cost Management**: Spending patterns and optimization opportunities
* **User Experience**: Satisfaction trends and problem areas
* **Product Development**: Data-driven feature improvement priorities

**Metric Formulas Summary**

| **Metric** | **Formula** | **Purpose** |
| --- | --- | --- |
| Satisfaction Rate | (positive\_feedback / total\_feedback) × 100 | User experience quality |
| Auto-Eval Score | matched\_keyphrases / total\_expected\_keyphrases | Content quality assessment |
| Average Latency | SUM(latency\_ms) / COUNT(\*) | Performance benchmarking |
| Total Cost | SUM((input\_tokens × $0.80 + output\_tokens × $4.00) / 1M) | Financial tracking |
| Token Efficiency | tokens\_total / satisfaction\_rate | Cost-effectiveness |
| Variant Performance | AVG(metric) GROUP BY variant | A/B testing analysis |