



Vantage RuntimeAI

Batch agent check-rides, model comparison, and operational scorecards for autonomous AI workflows

8.0/10

Overall score

Run summary

Scenario: Null Rate Alert

Model: deepseek/deepseek-chat

Roles: Data Engineer (AI agent) <> Engineering counterpart (AI agent)

Duration: 37s

Actual run cost: \$0.001492

Projected execution cost: \$2.46/mo (75/biz day × 22 days)

Report date: July 13, 2026 - 10:53 PM EST

Overall

8.0/10

PASS - trust=high

Pass - score 8.0/10 clears 7.0 with acceptable trust/closure.

CAPABILITY RUBRIC



#	Dimension (matches spoke #)	Score
1	Null spike diagnosis	8/10
2	Uses alert metrics	8/10
3	Stakeholder / deploy awareness	8/10
4	Structured triage summary	8/10
5	Validation SQL / next checks	8/10

Spoke numbers run clockwise from the top; dimension names are listed in full in the table.

Technical criteria checklist

Automated pass/fail signals for this task - use when comparing models on specific requirements.

- Null spike identified: Met
- Validation SQL proposed: Met
- Recent deploy considered: Met
- References alert column/table: Met

TECHNICAL JUDGEMENT

Holistic read on this single-shot deliverable - criteria checklist plus rubric gaps, not any one dimension alone.

8/10

Telemetry

Run metrics

Mode: multi-turn - Agent msgs: 2 - Counterpart msgs: 3 - Output: 4650 chars - Latency: 37s - Criteria: 4/4

Score analysis

Identifies null-rate spike on shipping_country_code.

Score snapshot

Solid: Solid technical task response with a few gaps.

Interpretation

Multi-turn technical task: fixed brief, 2 agent turns with scripted follow-ups and reviewer sign-off. 4 of 4 checklist criteria passed.

BRIGHT SPOTS (RUBRIC)

- + Null spike diagnosis: Correctly identifies the primary technical issue from the artifacts.
- + Uses alert metrics: Uses tables, logs, or schema facts from the task brief.
- + Stakeholder / deploy awareness: Shows appropriate safety, budget, or risk awareness for the scenario.
- + Structured triage summary: Clear structure with readable SQL, diff, or memo formatting.
- + Validation SQL / next checks: Strong rationale and actionable follow-through.

DIMENSION NOTES

- Null spike diagnosis: Correctly identifies the primary technical issue from the artifacts.
- Uses alert metrics: Uses tables, logs, or schema facts from the task brief.
- Stakeholder / deploy awareness: Shows appropriate safety, budget, or risk awareness for the scenario.
- Structured triage summary: Clear structure with readable SQL, diff, or memo formatting.
- Validation SQL / next checks: Strong rationale and actionable follow-through.
- Overall judgement (8/10): Sound judgement on this technical response: checklist criteria and rubric align with a strong deliverable.

Signals

Null spike identified: yes - Validation SQL proposed: yes - Recent deploy considered: yes - References alert column/table: yes

CAVEATS

- Heuristic keyword/signal scoring - use the criteria grid and transcript to validate edge cases.

Transcript

Session

Scenario: Null Rate Alert

Agent response <-> Engineering counterpart

Data Engineer (AI agent) <> Engineering counterpart (AI agent)

Duration: 37s

Engineering counterpart

2:52 AM

Task brief - Null rate alert (simple complexity)

A column null rate spiked overnight. Triage the alert JSON and recommend next steps:

```
```json
{
 "table": "analytics.orders",
 "column": "shipping_country_code",
 "window": "last_7_days",
 "null_rate_pct": {
 "baseline_30d_avg": 0.3,
 "yesterday": 17.8,
 "today_so_far": 19.2
 },
 "row_counts": {
 "total_orders_7d": 842000,
 "null_shipping_country_yesterday": 148900
 },
 "recent_deploys": [
 {"service": "checkout-api", "deploy_at": "2026-07-04T22:10:00Z", "change": "default empty string -> null for missing geo"},
 {"service": "orders-ingest", "deploy_at": "2026-07-03T08:00:00Z", "change": "no schema change"}
],
 "note": "Finance dashboard uses shipping_country_code for regional revenue splits."
}
```
```

Agent response

2:52 AM - Responded in 12s

1. Likely Cause

The spike in null rates (from 0.3% to ~18%) coincides with a deploy to `checkout-api` that changed default empty strings to `NULL` for missing geo data. This is almost certainly the root cause, as:

- The timing aligns (deploy at 2026-07-04T22:10:00Z -> nulls appeared overnight).
- The change explicitly mentions handling of missing values.

2. Validation SQL

```sql

SELECT

```
COUNT() FILTER (WHERE shipping_country_code IS NULL) AS null_count,
COUNT() FILTER (WHERE shipping_country_code = "") AS empty_string_count,
COUNT() AS total_orders
```

FROM analytics.orders

WHERE created\_at >= CURRENT\_DATE - INTERVAL '1 day';

```

Expected Result: High `null\_count` and near-zero `empty\_string\_count` (confirming empty strings were converted to NULLs).

3. Recent Deploy Suspect?

? Yes - The `checkout-api` deploy explicitly changed null handling. The `orders-ingest` deploy is unrelated (no schema change).

4. Stakeholder Comms

"We've traced the shipping_country_code null spike to a recent checkout-api deploy (4-Jul) that converted empty strings to NULLs. Validating fix options now-will update by EOD. Temporary workaround: use COALESCE(shipping_country_code, "") in Finance dashboards."

Next Steps:

- Confirm with validation SQL.
- Coordinate with checkout-api team to revert or adjust the change.
- Update dashboards to handle NULLs if this is intentional.

Engineering counterpart

2:52 AM - Responded in 0s

Thanks - walk me through your reasoning step by step. What assumptions did you make?

Agent response

2:53 AM - Responded in 25s

Here's my step-by-step reasoning, including explicit assumptions and decision points:

1. Observing the Spike

- Baseline: Null rate was stable at 0.3% for 30 days.
- Spike: Jumped to ~18% overnight.
- Assumption: A null rate spike this large is unlikely to be natural (e.g., sudden user behavior change). It's probably systemic (code/schema change).

2. Triaging Possible Causes

I considered (and ruled out) these potential causes:

A. Data Pipeline Breakage

- Check: `orders-ingest` deploy on 3-Jul has "no schema change" in notes.
- Assumption: No schema change -> unlikely to affect null rates.
- Decision: Rule this out.

B. Source System Change

- Evidence: The `checkout-api` deploy on 4-Jul explicitly mentions:
 - > "default empty string -> null for missing geo"
- Assumption: This is a direct schema/behavior change for the `shipping\_country\_code` field.

C. Dashboard Query Logic Change

- Note: The alert mentions Finance dashboard uses this column, but:
 - Dashboard changes wouldn't affect raw table null rates.
 - No deploys for dashboards are listed.
- Decision: Rule this out.

3. Validating the Leading Cause

Hypothesis: The `checkout-api` deploy caused empty strings to become NULLs.

Key Assumptions:

1. Empty strings were previously common (~17% of records had `shipping\_country\_code` = "").
 - Why? If empty strings were rare, converting them to NULLs wouldn't cause an 18% spike.
2. The deploy affected writes (not just API responses).
 - Implied by the alert showing table-level null rates.

Validation SQL Purpose:

- Confirm that:
 - `NULL` counts increased.
 - Empty string counts dropped to near-zero.
 - Total orders are stable (rules out data loss).

4. Ruling Out Alternatives

Could other factors explain the spike?

A. Surge in International Orders?

- Check: Nulls imply missing country codes, not new countries.
- Assumption: User behavior doesn't suddenly make 18% of orders omit country codes.

B. Third-Party API Failure?

- Check: No alerts for geo lookup services.
- Assumption: The explicit deploy note overrides this theory.

5. Why This Comms Draft?

- Stakeholder need: Finance cares about dashboards breaking, not root cause.
- Actionable fix: Suggest `COALESCE` as a temporary workaround.
- Transparency: Name the deploy but avoid blame ("we" not "they").

Explicit Assumptions Summary

1. Empty strings were previously common (~17% of records).
2. The deploy affected writes, not just API responses.
3. No silent failures (e.g., geo API downtime) coincided with the deploy.
4. Dashboard logic didn't change (no evidence in alert).

If Assumptions Are Wrong?

- If empty strings were rare pre-deploy, the spike would need another explanation (e.g., data corruption).
- If the deploy only affected API responses (not writes), the table null rate shouldn't change.

Would you adjust

Engineering counterpart**2:53 AM - Responded in 0s**

Thanks - assumptions and validation path are reasonable. Run the confirmatory null-rate query before paging broader - spike may be ingest timing. Review complete - no further response needed.