



Vantage RuntimeAI

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

8.4/10

Overall score

Run summary

Scenario: Train/Serve Skew

Model: ibm-granite/granite-4.0-h-micro

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

Duration: 2m 5s

Actual run cost: \$0.000388

Projected execution cost: \$0.0427/mo (5/biz day × 22 days)

Report date: July 13, 2026 - 11:45 PM EST

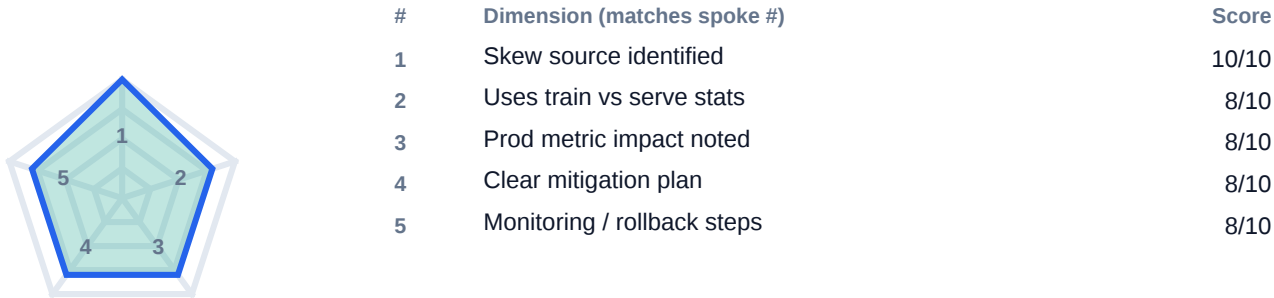
Overall

8.4/10

PASS - trust=high

Pass - score 8.4/10 clears 7.0 with acceptable trust/closure.

CAPABILITY RUBRIC



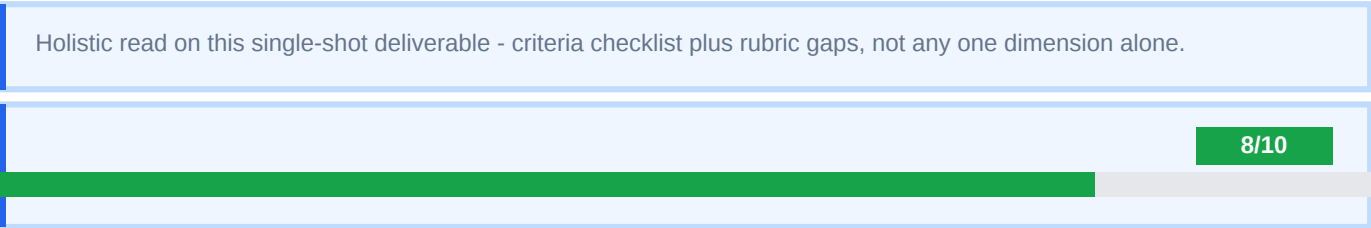
Spoke numbers run clockwise from the top; dimension names are listed in full in the table.

Technical criteria checklist

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

- Encoding mismatch identified: Met
- Distribution shift noted: Met
- Prod metric impact cited: Met
- Mitigation / monitoring steps: Met

TECHNICAL JUDGEMENT



Telemetry

Run metrics

Mode: multi-turn - Agent msgs: 6 - Counterpart msgs: 7 - Output: 20840 chars - Latency: 125s - Criteria: 4/4

Score analysis

Identifies one-hot vs hash encoding mismatch as skew source.

Score snapshot

Solid: Solid technical task response with a few gaps.

Interpretation

Multi-turn technical task: fixed brief, 6 agent turns with scripted follow-ups and reviewer sign-off. 4 of 4 checklist criteria passed.

BRIGHT SPOTS (RUBRIC)

- + Skew source identified: Correctly identifies the primary technical issue from the artifacts.
- + Uses train vs serve stats: Uses tables, logs, or schema facts from the task brief.
- + Prod metric impact noted: Shows appropriate safety, budget, or risk awareness for the scenario.
- + Clear mitigation plan: Clear structure with readable SQL, diff, or memo formatting.
- + Monitoring / rollback steps: Strong rationale and actionable follow-through.

DIMENSION NOTES

- Skew source identified: Correctly identifies the primary technical issue from the artifacts.
- Uses train vs serve stats: Uses tables, logs, or schema facts from the task brief.
- Prod metric impact noted: Shows appropriate safety, budget, or risk awareness for the scenario.
- Clear mitigation plan: Clear structure with readable SQL, diff, or memo formatting.
- Monitoring / rollback steps: 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

Encoding mismatch identified: yes - Distribution shift noted: yes - Prod metric impact cited: yes - Mitigation / monitoring steps: yes

CAVEATS

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

Transcript

Session

Scenario: Train/Serve Skew

Agent response <-> Engineering counterpart

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

Duration: 2m 5s

Engineering counterpart

3:43 AM

Task brief - Train/serve skew (complex complexity)

Production precision dropped after an encoding change. Investigate using the artifact:

```
```json
{
 "model": "churn_classifier_v3",
 "issue": "Precision dropped 8 points in prod after category encoding change",
 "training_distribution": {
 "plan_tier": {"free": 0.42, "pro": 0.38, "enterprise": 0.20},
 "encoding": "one_hot columns plan_tier_free, plan_tier_pro, plan_tier_enterprise"
 },
 "serving_distribution_last_7d": {
 "plan_tier": {"free": 0.61, "pro": 0.29, "enterprise": 0.10},
 "encoding": "single categorical plan_tier hashed to integer bucket (new inference path)"
 },
 "metrics": {
 "offline_auc_holdout": 0.84,
 "online_precision_7d": 0.71,
 "online_precision_baseline_30d": 0.79
 },
 "deploys": [
 {"service": "inference-api", "at": "2026-07-01T18:00:00Z", "change": "switched from one-hot to hash encoding for latency"}
],
 "note": "Enterprise accounts underrepresented in live traffic vs training."
}
```
```

Agent response

3:43 AM - Responded in 24s

Investigation of Train/Serve Skew Leading to Precision Drop

1. Overview of the Situation

The churn classifier model (`churn\_classifier\_v3`) experienced a significant drop in precision from 0.79 (baseline over 30 days) to 0.71 in the last 7 days of production. This drop coincides with a change in the encoding strategy for the `plan\_tier` feature from one-hot encoding to a hash encoding. The discrepancy between the training and serving distributions, coupled with the underrepresentation of enterprise accounts in live traffic, suggests a train/serve skew.

2. Analysis of the Data

2.1 Training vs. Serving Distributions

- Training Distribution:

- `plan\_tier`: free (0.42), pro (0.38), enterprise (0.20)
- Encoding: one-hot for `plan\_tier\_free`, `plan\_tier\_pro`, `plan\_tier\_enterprise`

- Serving Distribution (Last 7 Days):

- `plan\_tier`: free (0.61), pro (0.29), enterprise (0.10)
- Encoding: single categorical feature hashed to an integer bucket

The significant shift in the distribution of `plan\_tier` from 42% free / 38% pro / 20% enterprise in training to 61% free / 29% pro / 10% enterprise in production, along with the change in encoding strategy, is likely the primary cause of the precision drop.

2.2 Impact of Encoding Change

- One-Hot Encoding: Each category (`free`, `pro`, `enterprise`) was represented by a separate binary feature. This allows the model to learn distinct relationships for each category.
- Hash Encoding: The feature is now hashed into a single integer bucket. This reduces dimensionality but can lead to loss of information, especially if the hash function does not distribute the categories evenly or if the hash function introduces collisions.

The switch to hash encoding likely introduced noise or loss of information, particularly for the `enterprise` category, which is underrepresented in live traffic. This could lead to the model being less accurate for enterprise accounts, which are crucial for churn prediction.

2.3 Underrepresentation of Enterprise Accounts

Enterprise accounts constitute only 10% of the serving distribution but are likely more critical for churn prediction due to higher revenue potential. Their underrepresentation in live traffic exacerbates the skew, as the model has less exposure to enterprise data during training and less opportunity to learn from enterprise examples during inference.

3. Potential Causes of Skew

1. Encoding Strategy Change: The switch from one-hot to hash encoding altered the feature representation, potentially disrupting the model's ability to learn from the data.
2. Distribution Mismatch: The disparity between training and serving distributions means the model is optimized for a different data distribution than what it encounters in production.
3. Underrepresentation of Enterprise Accounts: Enterprise accounts are underrepresented in live traffic, leading to a lack of exposure for the model to learn from these critical cases.

4. Mitigation Options

4.1 Revert to One-Hot Encoding

- Pros: Preserves the distinct representation of each category, potentially restoring the model's ability to learn from the data accurately.

- Cons: May increase model complexity and latency due to higher dimensionality.

4.2 Adjust Hash Encoding Strategy

- Pros: Maintains lower dimensionality while potentially improving distribution alignment.
- Cons: Requires careful tuning of the hash function to ensure even distribution and minimize collisions.

4.3 Data Augmentation for Underrepresented Categories

- Pros: Increases the model's exposure to enterprise accounts, potentially improving its predictive power for these cases.
- Cons: Requires additional data collection and preprocessing efforts.

4.4 Collect More Enterprise Data

-

Engineering counterpart
3:43 AM - Responded in 0s

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

Agent response

3:44 AM - Responded in 24s

Certainly! Let's walk through the reasoning step by step, detailing the assumptions made at each stage of the investigation into the train/serve skew causing the precision drop in the churn classifier model.

Step 1: Identify the Symptom

Symptom: Precision dropped from 0.79 (baseline over 30 days) to 0.71 in the last 7 days of production.

Assumption: The drop in precision is directly related to the change in encoding strategy for the `plan\_tier` feature. This assumption is based on the observation that the change in encoding coincides with the timing of the precision drop.

Step 2: Examine Training vs. Serving Distributions

Data Provided:

- Training Distribution:

- `plan\_tier`: free (0.42), pro (0.38), enterprise (0.20)
- Encoding: one-hot for `plan\_tier\_free`, `plan\_tier\_pro`, `plan\_tier\_enterprise`

- Serving Distribution (Last 7 Days):

- `plan\_tier`: free (0.61), pro (0.29), enterprise (0.10)
- Encoding: single categorical feature hashed to an integer bucket

Assumption: The discrepancy between the training and serving distributions is the primary cause of the precision drop. This assumption is based on the principle that models perform best when trained and tested on similar data distributions.

Rationale:

- Distribution Mismatch: The training distribution shows a higher proportion of `pro` and `free` accounts compared to the serving distribution. This mismatch suggests that the model has not been exposed to the same distribution of data during inference as it was during training.
- Enterprise Underrepresentation: The `enterprise` category is significantly underrepresented in the serving distribution (10% vs. 20% in training). This underrepresentation could lead to the model being less accurate for enterprise accounts, which are critical for churn prediction.

Step 3: Analyze the Impact of Encoding Change

Encoding Changes:

- Before Change: One-hot encoding for `plan\_tier`.
- After Change: Hash encoding for `plan\_tier`.

Assumption: The switch from one-hot to hash encoding altered the feature representation, potentially disrupting the model's ability to learn from the data.

Rationale:

- One-Hot Encoding: Each category (`free`, `pro`, `enterprise`) is represented by a separate binary feature. This allows the model to learn distinct relationships for each category, which is crucial for accurate predictions.
- Hash Encoding: The feature is now hashed into a single integer bucket. This reduces dimensionality but can lead to loss of information, especially if the hash function does not distribute the categories evenly or if the hash function introduces collisions. The loss of distinct category representation can degrade model performance, particularly for underrepresented categories like `enterprise`.

Step 4: Investigate the Underrepresentation of Enterprise Accounts

Assumption: Enterprise accounts are underrepresented in live traffic, leading to a lack of exposure for the model to learn from these critical cases.

Rationale:

- Impact on Model Learning: Models learn from the data they are exposed to during training and inference. If enterprise accounts are underrepresented in live traffic, the model has less opportunity to learn from enterprise examples, both during

training (when the training data is imbalanced) and during inference (when the serving distribution is skewed).

- Critical for Churn Prediction: Enterprise accounts are likely more critical for churn prediction due to higher revenue potential. Their underrepresentation exacerbates the skew, as the model may not generalize well to enterprise cases.

Step 5: Formulate Mitigation Options

Based on the analysis, several mitigation options were considered to address the train/serve skew and restore model precision.

4.1 Revert to One-Hot Encoding

- Pros:

Engineering counterpart
3:44 AM - Responded in 0s

What would you verify before shipping this to production?

Agent response

3:44 AM - Responded in 24s

Before shipping the mitigation to production, several critical verifications should be conducted to ensure that the changes will effectively address the train/serve skew and restore the model's precision without introducing new issues. Here's a detailed checklist of verifications:

1. Data Validation

- Verify Distribution Alignment:

- Ensure that the serving distribution now closely matches the training distribution, particularly for the `plan\_tier` feature.

This can be done by comparing the distribution of `plan\_tier` values in both training and serving datasets.

- Verification Method: Use statistical tests (e.g., chi-squared test) to assess the similarity between the distributions.

- Check Enterprise Account Representation:

- Confirm that the serving distribution now has a more balanced representation of `enterprise` accounts, ideally closer to the training distribution (20% enterprise vs. 10% in production).

- Verification Method: Calculate the percentage of `enterprise` accounts in both training and serving datasets and ensure the serving distribution is closer to the training distribution.

2. Model Performance Evaluation

- Offline Evaluation:

- AUC and Precision: Re-evaluate the model's performance using offline metrics such as AUC and precision on the updated serving data. Ensure that the precision has improved and is closer to the baseline (0.79) observed over 30 days.

- Verification Method: Run the model on a validation set that reflects the updated serving distribution and compare the results with the baseline metrics.

- Online Evaluation:

- Real-Time Monitoring: Deploy the updated model in a controlled environment (e.g., shadow mode) to monitor its performance in real-time. Compare the online precision and other relevant metrics against the baseline.

- Verification Method: Use monitoring tools to track key metrics such as precision, recall, and AUC in production. Set up alerts for any significant deviations from expected performance.

3. Model Interpretability and Explainability

- Feature Importance Analysis:

- Verify that the model's feature importance remains consistent or improves after the encoding change. This ensures that the model is still relying on meaningful features for predictions.

- Verification Method: Use techniques like SHAP values or permutation importance to assess feature contributions.

- Explainability Checks:

- Ensure that the model's explanations (e.g., feature contributions) are still valid and meaningful. This helps in understanding why the model is making certain predictions and ensures that the changes have not introduced bias or opacity.

- Verification Method: Conduct manual or automated explanations for a sample of predictions to ensure they align with business logic and expectations.

4. Performance Under Different Scenarios

- Edge Cases and Anomalies:

- Test the model with edge cases and anomalies to ensure it behaves as expected. This includes scenarios where the `plan\_tier` feature might be atypical or missing.

- Verification Method: Create synthetic test cases that cover a range of possible inputs and evaluate the model's predictions.

- Latency and Resource Utilization:

- Ensure that the chosen encoding strategy (whether reverted to one-hot or adjusted hash encoding) does not introduce significant latency or resource overhead. This is crucial for maintaining the model's efficiency in production.

- Verification Method: Measure the inference latency and resource usage (CPU, memory) for both the old and new

encoding strategies. Compare these metrics to ensure they meet performance requirements.

5. Rollback and Monitoring Plan

- Rollback Strategy:

- Develop a clear rollback plan in case the new encoding strategy introduces issues. This includes having the previous encoding strategy readily available and ensuring that the deployment process can revert quickly if needed.

- Verification Method: Document the steps required to revert to the previous encoding strategy and test the rollback process in a staging environment.

- Monitoring and Alerting:

- Implement

Engineering counterpart
3:44 AM - Responded in 0s

What are the top risks, and what's the rollback plan if this doesn't work?

Agent response

3:44 AM - Responded in 25s

Top Risks Associated with the Train/Serve Skew Mitigation

Before proceeding with the mitigation, it's crucial to identify and understand the potential risks involved. Here are the top risks and a corresponding rollback plan if the mitigation does not yield the expected results:

1. Model Performance Degradation

Risk: The mitigation could inadvertently degrade the model's performance rather than improve it.

- Impact: A decrease in precision, recall, or AUC could lead to poorer churn prediction accuracy, resulting in higher customer churn rates and financial losses.
- Mitigation: Conduct thorough offline and online evaluations before and after the change. Ensure that the model's performance metrics meet or exceed baseline levels.

2. Increased Model Complexity and Latency

Risk: Reverting to one-hot encoding or adjusting the hash encoding strategy might increase the model's complexity, leading to higher latency and resource consumption.

- Impact: Increased latency could degrade user experience, especially in real-time applications, and lead to higher operational costs.
- Mitigation: Benchmark the performance of the new encoding strategy in terms of latency and resource utilization. If the increase is significant, consider alternative strategies or optimizations.

3. Underrepresentation of Enterprise Accounts Remains Unresolved

Risk: Even after encoding changes, enterprise accounts might still be underrepresented in live traffic, leading to persistent skew.

- Impact: The model may not generalize well to enterprise accounts, continuing to underperform in critical scenarios.
- Mitigation: Ensure that the serving distribution is balanced, particularly for the `enterprise` category. If necessary, augment the training data with more enterprise examples or implement data augmentation techniques.

4. Data Drift

Risk: The change in encoding strategy could introduce data drift, where the distribution of the input data changes over time, causing the model to become less accurate.

- Impact: Data drift can lead to a gradual decline in model performance over time, requiring continuous retraining or reevaluation of the encoding strategy.
- Mitigation: Monitor the distribution of input features continuously. Set up alerts for significant deviations from the expected distribution and be prepared to retrain or adjust the encoding strategy as needed.

5. Increased Complexity in Model Maintenance

Risk: The new encoding strategy might introduce additional complexity in model maintenance, making it harder to update or debug the model.

- Impact: Increased maintenance complexity could lead to longer development cycles, higher operational costs, and potential errors during updates.
- Mitigation: Document the encoding strategy changes thoroughly and ensure that the deployment and monitoring processes are well-documented. Train the operations team on the new encoding strategy to facilitate smoother maintenance.

6. Operational Disruptions

Risk: Deploying the mitigation could lead to operational disruptions if not executed carefully.

- Impact: Downtime or performance issues during deployment could negatively impact business operations and customer satisfaction.
- Mitigation: Perform a thorough testing phase in a staging environment before deploying to production. Use canary releases or shadow mode deployments to gradually roll out the changes and monitor for any issues.

Rollback Plan if Mitigation Does Not Work

If the mitigation does not resolve the precision drop or introduces new issues, a well-defined rollback plan is essential to quickly revert to the previous state and minimize impact. Here's a step-by-step rollback plan:

Step 1: Immediate Detection of Issues

- Monitoring: Continuously monitor key metrics (precision, latency, resource usage) in real-time. Set up alerts for any deviations from expected performance.
- Alert Response: If any critical metrics deviate significantly from baseline levels, trigger the rollback process immediately.

Step 2: Immediate Rollback to Previous Encoding Strategy

- Deployment Rollback: If the new encoding strategy (either reverted to one-hot or adjusted hash encoding) is causing issues, immediately roll back to the previous encoding strategy

Engineering counterpart
3:44 AM - Responded in 0s

Can you tighten this into concrete next steps with owners and timing?

Agent response

3:45 AM - Responded in 25s

Certainly! Below is a concrete action plan with specific next steps, owners, and timelines to address the train/serve skew issue and mitigate the precision drop in the churn classifier model. This plan is designed to be actionable and time-bound, ensuring that the necessary actions are taken promptly and efficiently.

Action Plan: Addressing Train/Serve Skew and Restoring Model Precision

1. Immediate Data Validation and Distribution Analysis

Objective: Verify that the serving distribution aligns closely with the training distribution, particularly for the `plan\_tier` feature, and that enterprise accounts are adequately represented.

Steps:

1. Owner: Data Engineering Lead

- Action: Compare the distribution of `plan\_tier` in training and serving datasets. Calculate the percentage of `free`, `pro`, and `enterprise` accounts in both datasets.
- Timeline: 1 business day

2. Owner: Data Science Lead

- Action: Analyze the impact of the encoding change on model performance using offline metrics (AUC, precision). Compare these metrics with the baseline (0.79).
- Timeline: 2 business days

Deliverables:

- Distribution comparison report.
- Performance comparison report showing the impact of the encoding change.

2. Model Performance Evaluation and Monitoring

Objective: Ensure that the model's performance improves after the encoding change and monitor its behavior in real-time.

Steps:

1. Owner: Machine Learning Engineer

- Action: Re-evaluate the model's performance using offline metrics (AUC, precision) on the updated serving data. Conduct a shadow deployment to monitor real-time performance.
- Timeline: 3 business days

2. Owner: Data Science Lead

- Action: Set up monitoring tools to track key metrics (precision, recall, AUC) in production. Establish alerts for any significant deviations from expected performance.
- Timeline: 2 business days

Deliverables:

- Performance evaluation report showing improved metrics.
- Monitoring setup documentation and alert configuration.

3. Model Interpretability and Explainability Checks

Objective: Verify that the model's feature importance remains consistent or improves after the encoding change and ensure that explanations are valid and meaningful.

Steps:

1. Owner: Machine Learning Engineer

- Action: Use SHAP values or permutation importance to assess feature contributions. Ensure that the model's explanations are still valid and meaningful.
- Timeline: 2 business days

2. Owner: Data Science Lead

- Action: Conduct manual or automated explanations for a sample of predictions. Validate that explanations align with business logic and expectations.
- Timeline: 1 business day

Deliverables:

- Feature importance analysis report.
- Explanation validation report.

4. Performance Under Different Scenarios

Objective: Test the model with edge cases and anomalies to ensure robustness and generalizability.

Steps:

1. Owner: Machine Learning Engineer

- Action: Create synthetic test cases covering a range of possible inputs, including edge cases and anomalies. Evaluate the model's predictions for these scenarios.
- Timeline: 3 business days

2. Owner: Data Science Lead

- Action: Review the results of the edge case testing. Ensure that the model behaves as expected and does not introduce bias or opacity.
- Timeline: 1 business day

Deliverables:

- Edge case testing report.
- Validation of model robustness and generalizability.

5. Rollback and Monitoring Plan

Objective: Develop a clear rollback strategy and monitoring plan in case the new encoding strategy introduces issues.

Steps:

1. Owner: DevOps Lead

- Action: Document the steps required to revert to the previous encoding strategy

Engineering counterpart

3:45 AM - Responded in 0s

What's the one thing you'd want a second pair of eyes on?

Agent response

3:45 AM - Responded in 3s

The one thing I would want a second pair of eyes on is the monitoring and rollback plan. Ensuring that the monitoring setup is robust, the alerts are appropriately configured, and the rollback process is well-documented and tested is critical. This step directly impacts the safety and reliability of the deployment, and having an independent review can help identify potential gaps or oversights that might otherwise go unnoticed.

Engineering counterpart

3:45 AM - Responded in 0s

Thanks - skew source and mitigation path are clearer now. Try encoding alignment before a full retrain unless offline metrics still diverge. Review complete - no further response needed.