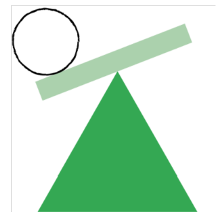
## Errors + Grace Failure



# Chapter worksheet

## Instructions

Block out time to get as many cross-functional leads as possible together in a room to work through these exercises & checklists.

## Exercises

### 1. Error audit [~1 hour]

Collect canonical error examples to define existing and potential errors and solutions.

### 2. Quality assurance [~30 minutes]

Prioritize how you'll test and monitor errors and reporting so you can hear from your users early and often.

## 1. Error audit

As a team, brainstorm what kinds of errors users could encounter. If your team has a working prototype of your feature, try to add current examples.

Use the template below to start collecting error examples so your team has a shared understanding about the different error types and solutions your model could produce.

| **Error**<br><br>Add screenshots, pictures, or logs to show what the user sees when encountering the error |
|---|
| 1. Error: Empty or missing recommendations<br>Users: New or low-activity users<br>User Stakes: High<br>Error Type: System limitation<br>Why It Happens: Cold-start problem: user/book lacks interaction data<br><br>2. **Error:** Repetitive or generic recommendations<br>**Users:** Returning users<br>**User Stakes:** Medium<br>**Error Type:** Model bias<br>**Why It Happens:** Model overfits to popular books and fails to diversify results<br><br>3. **Error:** Irrelevant recommendations<br>**Users:** All users<br>**User Stakes:** High<br>**Error Type:** Context drift<br>**Why It Happens:** Model not retrained frequently enough to reflect new user behavior<br><br>4. **Error:** Failed data pipeline<br>**Users:** Developers / All users<br>**User Stakes:** High<br>**Error Type:** Background system error<br>**Why It Happens:** Airflow or BigQuery job failure due to schema mismatch or resource limits<br><br>5. **Error:** API timeout / slow response<br>**Users:** End users<br>**User Stakes:** Medium<br>**Error Type:** System hierarchy error<br>**Why It Happens:** Query joins or large result sets exceed BigQuery timeout limit |

## Error sources

**1. Error: Empty or Missing Recommendations**

**Error Sources**

Take each error identified above through these questions to determine the source of the error:

**Input Error Signals**

**Did the user anticipate the auto-correction of their input into an AI system?**
No. The user expects the system to show recommendations automatically without having to provide prior ratings or reviews.

**Was the user's habituation interrupted?**
Yes. The empty results screen breaks the normal expectation of seeing book suggestions, causing confusion.

**Did the model improperly weigh a user action or other signal?**
Yes. The model gives insufficient weight to cold-start users because they lack prior interactions.

**Relevance Error Signals**

**Is the model lacking available data or requirements for prediction accuracy?**
Yes. New users and books lack historical data for accurate recommendations.

**Is the model receiving unstable or noisy data?**
No. The issue arises from lack of data rather than data instability.

**Is the system output presented to users in a way that isn't relevant to the user's needs?**
Yes. The user receives an empty or blank recommendation area with no explanation or fallback.

**System Hierarchy Error**

**Is your user connecting your product to another system, and it isn't clear which system is in charge?**
No. This is an internal model data sparsity issue.

**Are there multiple systems monitoring a single (or similar) output and an event causes simultaneous alerts?**
No. This error is localized within the recommendation service.

**Failure State**

**Is your feature unusable as the result of multiple errors?**
Yes. Users cannot engage with the recommendation feature at all.

**2. Error: Repetitive or Generic Recommendations**

**Error Sources**

Take each error identified above through these questions to determine the source of the error:

**Input Error Signals**

**Did the user anticipate the auto-correction of their input into an AI system?**
No. Users expect recommendations to improve over time based on feedback, which doesn't occur here.

**Was the user's habituation interrupted?**
Partially. Users see repetitive books and lose interest gradually.

**Did the model improperly weigh a user action or other signal?**
Yes. The model overemphasizes popular items and ignores user-specific behaviors like dismissing suggestions.

**Relevance Error Signals**

**Is the model lacking available data or requirements for prediction accuracy?**
No. The data is sufficient but the weighting mechanism prioritizes popularity over personalization.

**Is the model receiving unstable or noisy data?**
No. Data is stable but biased toward overrepresented genres or authors.

**Is the system output presented to users in a way that isn't relevant to the user's needs?**
Yes. Repetitive or overly generic recommendations do not align with user preferences.

**System Hierarchy Error**

**Is your user connecting your product to another system, and it isn't clear which system is in charge?**
No. This issue originates within the model's ranking logic.

**Are there multiple systems monitoring a single (or similar) output and an event causes simultaneous alerts?**
No. There's no overlap in system monitoring for this error.

**Failure State**

**Is your feature unusable as the result of multiple errors?**
No, but user satisfaction and trust decrease over time, lowering engagement.

**3. Error: Irrelevant Recommendations**

**Error Sources**

Take each error identified above through these questions to determine the source of the error:

**Input Error Signals**

**Did the user anticipate the auto-correction of their input into an AI system?**
Yes. Users expect recommendations to quickly reflect their changing interests.

**Was the user's habituation interrupted?**
Yes. Seeing irrelevant books disrupts the personalized experience.

**Did the model improperly weigh a user action or other signal?**
Yes. The model gives too much weight to older ratings or outdated preferences.

**Relevance Error Signals**

**Is the model lacking available data or requirements for prediction accuracy?**
No. Data exists but retraining frequency is too low.

**Is the model receiving unstable or noisy data?**
Partially. Some signals become stale or outdated, reducing prediction quality.

**Is the system output presented to users in a way that isn't relevant to the user's needs?**
Yes. Book recommendations no longer match user preferences or interests.

**System Hierarchy Error**

**Is your user connecting your product to another system, and it isn't clear which system is in charge?**
No. This is purely a model maintenance issue.

**Are there multiple systems monitoring a single (or similar) output and an event causes simultaneous alerts?**
No. Only the recommendation model governs output here.

**Failure State**

**Is your feature unusable as the result of multiple errors?**
Partially. Users can still see results, but they become irrelevant and unhelpful.

**4. Error: Failed Data Pipeline**

**Error Sources**

Take each error identified above through these questions to determine the source of the error:

**Input Error Signals**

**Did the user anticipate the auto-correction of their input into an AI system?**
No. Users are unaware of the backend processes.

**Was the user's habituation interrupted?**
Yes. The application fails to refresh or update results due to backend job failure.

**Did the model improperly weigh a user action or other signal?**
No. The issue stems from data unavailability, not signal misinterpretation.

**Relevance Error Signals**

**Is the model lacking available data or requirements for prediction accuracy?**
Yes. Pipeline failure interrupts the flow of new data for training and inference.

**Is the model receiving unstable or noisy data?**
Yes. Partial or truncated data can lead to inconsistent results.

**Is the system output presented to users in a way that isn't relevant to the user's needs?**
Yes. Users see stale or missing recommendations due to broken data flow.

**System Hierarchy Error**

**Is your user connecting your product to another system, and it isn't clear which system is in charge?**
Yes. Failures may occur across Airflow, BigQuery, or the model hosting environment, making it unclear which subsystem failed.

**Are there multiple systems monitoring a single (or similar) output and an event causes simultaneous alerts?**
Yes. Multiple monitoring tools (Airflow logs, GCP alerts) may trigger simultaneously.

**Failure State**

**Is your feature unusable as the result of multiple errors?**
Yes. Users cannot access any recommendations until the pipeline is restored.

**5. Error: API Timeout / Slow Response**

**Error Sources**

Take each error identified above through these questions to determine the source of the error:

**Input Error Signals**

**Did the user anticipate the auto-correction of their input into an AI system?**
No. The user expects a fast, seamless experience like other online apps.

**Was the user's habituation interrupted?**
Yes. Slow responses break task flow and user expectations.

**Did the model improperly weigh a user action or other signal?**
No. The issue occurs after inference, during result delivery.

**Relevance Error Signals**

**Is the model lacking available data or requirements for prediction accuracy?**
No. The prediction itself is correct; latency affects accessibility.

**Is the model receiving unstable or noisy data?**
No. Data processing is stable, but query response time is high.

**Is the system output presented to users in a way that isn't relevant to the user's needs?**
Yes. Delayed or incomplete loading makes the experience frustrating and seemingly broken.

**System Hierarchy Error**

**Is your user connecting your product to another system, and it isn't clear which system is in charge?**
Yes. The frontend depends on BigQuery APIs, making the source of failure unclear.

**Are there multiple systems monitoring a single (or similar) output and an event causes simultaneous alerts?**
Yes. Both frontend and backend may flag timeouts, causing alert confusion.

**Failure State**

**Is your feature unusable as the result of multiple errors?**
Yes. The delay effectively renders the recommendation engine unresponsive.

## Error resolution

Once you have identified the source or sources of the error, complete the sections below for each of the errors in the template with your team's plan for improving / reducing the identified error: Create as many copies as you need to cover all your identified errors.

| 1. Error: Empty or Missing Recommendations | Solution type |
|---|---|
| **Error rationale:** Users perceive a blank recommendation area as a system failure rather than a lack of data. They expect some kind of fallback content and don't understand that the model has insufficient data to make predictions. | ✅ Feedback<br>✅ User control<br>✅ Other: System fallback logic |
| **Error resolution**<br><br>User path: When no personalized data is available, the system displays a "Recommended for new readers" section with popular or trending books. The user can rate or like/dislike a few to jumpstart personalization.<br><br>Opportunity for model improvement: User-provided quick ratings from cold-start scenarios are logged and used to seed the collaborative filtering matrix or fine-tune embeddings for new profiles. | |

| 2. Error: Repetitive or Generic Recommendations<br><br>**Error rationale**<br><br>Users feel the system is “stuck” — it keeps suggesting the same popular books regardless of their changing preferences, reducing trust in personalization. | **Solution type**<br><br>✅Feedback<br>✅Other: Algorithmic diversity enhancement |
|---|---|
| **Error resolution**<br><br>User path:<br><br>User marks a recommendation as “Not interested” or “Show more like this.” System immediately updates the next batch of recommendations with greater genre diversity.<br><br>Opportunity for model improvement:<br><br>Feedback data helps reweight diversity terms in ranking (e.g., penalizing over-represented authors or genres). A diversity regularization component can be added to the model’s objective function. | |

| 3. Error: Irrelevant Recommendations<br><br>**Error rationale**<br><br>Why the user thinks this is an error:<br>Users see books that no longer align with their interests or recent reading patterns. They assume the AI has “forgotten” their behavior or that it’s outdated | **Solution type**<br><br>✅Feedback<br>✅Other: Frequent model retraining with time-decay weighting |
|---|---|

### Error resolution

User path:

User gives explicit feedback (“Not relevant”) or interacts with newer content. The system temporarily filters results based on more recent signals while queuing retraining.

Opportunity for model improvement:

Incorporate temporal weighting for interactions — recent ratings and reads should have more influence. Feedback logs can trigger scheduled retraining or fine-tuning jobs in Airflow.

**4. Error: Failed Data Pipeline**

### Error rationale

Users perceive it as a total application failure since no recommendations or updates appear. Developers also lose trust due to inconsistent data refresh

### Solution type

✅ Other: Automated monitoring, data recovery, and alerting

### Error resolution

User path:

End users see a fallback message (“Recommendations temporarily unavailable”) while the system self-heals. Backend automatically retries pipeline jobs and notifies the monitoring dashboard

Opportunity for model improvement:

Add Airflow sensors and BigQuery schema validation to prevent incomplete runs. Implement automated Slack/email alerts and retry mechanisms for transient failures.

| 5. Error: API Timeout / Slow Response<br><br>**Error rationale**<br><br>Users assume the site is broken or “frozen.” Long loading times feel like errors even if the model’s prediction is correct. | **Solution type**<br><br>✅ User Control<br>✅Other: Performance optimization, caching |
|---|---|
| **Error resolution**<br><br>User path:<br><br>If a timeout occurs, the user sees a progress animation with an option to “View cached recommendations” or retry. They maintain a sense of control instead of abandoning the task<br><br>Opportunity for model improvement:<br><br>Implement result caching and async prefetching in the backend. Track query latency metrics to optimize slow joins or API bottlenecks. Use logs to identify slow queries for model-serving optimization | |

# 2. Quality assurance

Getting your feature into users’ hands is essential for identifying errors that your team, as expert users, may never encounter. Meet as a team to prioritize how you want to monitor errors reported by users so that your model is being tested and criticized by your users early and often.

As you have this discussion, consider all potential sources of error reporting:

- Reports sent to customer service
- Comments and reports sent through social media channels
- In-product metrics
- In-product surveys
- User research (out-of-product surveys, deep dive interviews, diary studies, etc.)

# QA template

<table>
<tr>
<td>

**Goal**

Continuously monitor, detect, and reduce errors in the recommendation system to ensure high-quality, relevant, and timely book suggestions for users.

</td>
<td rowspan="3">

**Review frequency**

Daily

✅ Weekly

Monthly

Other:

</td>
</tr>
<tr>
<td>

**Method**

1. **In-product metrics:** Track key indicators like click-through rate, dwell time, diversity score, and latency from BigQuery dashboards.
2. **User feedback loop:** Integrate a “Was this recommendation helpful?” rating and track user selections (like/dislike).
3. **User research:** Conduct periodic user interviews or surveys to identify perception of personalization quality.
4. **System monitoring:** Use Airflow logs and GCP alerts to monitor pipeline or API failures.
5. **Social feedback review:** Scan app store or forum feedback for mentions of recommendation relevance.

</td>
</tr>
<tr>
<td>

Start date: October 2025

Review / End date: Ongoing – initial assessment every week with quarterly retraining summary reviews

</td>
</tr>
</table>