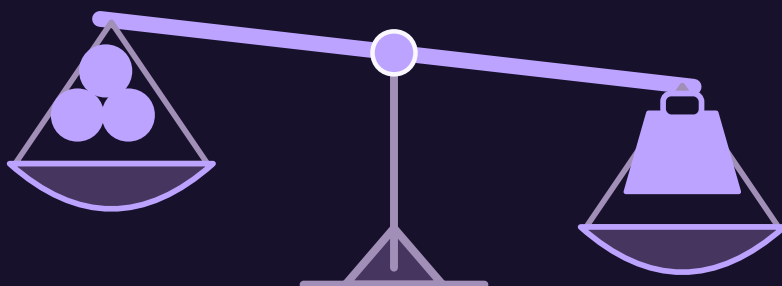




PRACTICAL AI

Build AI Evaluation Scorecards That Explain Tradeoffs

Combine paired evidence, release gates, and readable decision records



Sanjaay Gowda T S

FREE SAMPLE / INTRODUCTION + CHAPTER 1 / EDITION 1.0.0

sanjaaygowdats.ai

About this free sample

Build AI Evaluation Scorecards That Explain Tradeoffs

Sanjaay Gowda T S

First edition 1.0.0, 2026-10-13

Copyright 2026 Sanjaay Gowda T S. This is a labelled free sample, not the complete ebook. You may share this sample unchanged. It contains the cover, sample permissions, contents, introduction, Chapter 1 and a closing product link. The remaining chapters, worked solutions and companion files are not included. The full ebook text, diagrams and artwork have separate single-reader terms; this sample permission does not authorize sharing the full purchased PDF or EPUB.

The companion source code, tests and fictional data are separately MIT licensed. You may adapt or share that project with its included licence notice. MIT does not apply to the ebook text or book artwork. The books and companion projects have no technical activation requirement or buyer-specific watermark.

This first edition was prepared with AI assistance. The case studies are fictional teaching examples. Any measured outputs come from the supplied examples, not production customer data. The teaching scope, assumptions and limitations are explained in the chapters. Independent human technical, editorial and assistive-technology reader review is pending. Automated tests and accessibility checks do not establish certification or suitability for every use.

Run examples in a separate working folder using the supplied synthetic data. Review inputs and calculations before adapting the techniques to real operations. Model software, service access, subscriptions, ongoing updates and consulting are not included.

Author website: sanjaaygowdats.ai

Ebook terms: sanjaaygowdats.ai/ebooks/license

Inside this free sample

Start here.....	4
1. Begin with a release decision	5
Continue with the full guide	6

This free sample contains the introduction and Chapter 1 only. The closing link opens the complete guide. The companion project and worked solutions are supplied with the complete ebook.

Start here

A scorecard should help a reviewer decide what to release and explain why. It should not hide the decision inside an unexplained rank. An AI candidate can be cheaper and faster while making a critical mistake. Another can improve difficult cases while slowing every routine request. Both facts should remain visible after the summary is calculated.

This book builds a deterministic scorecard for three fictional candidates named A, B, and C. Each has supplied quality, latency, cost, and critical-failure values for the same two synthetic cases. The program validates paired coverage, calculates means and slice results, applies mandatory gates, and marks eligible candidates that retain a Pareto tradeoff. No model produced these values. They are teaching fixtures chosen to make the arithmetic and decision policy inspectable.

Candidate A averages quality 0.80, latency 500 milliseconds, and cost two synthetic units. B averages 0.95, 800 milliseconds, and three units. C appears better on every average but has a critical failure, so it is ineligible. The example demonstrates why mandatory requirements must remain separate from preferences.

Run `python main.py --input example-input.json` in project, then `python -m unittest discover -s tests -v`. You need Python 3.10 or later; the standard library is sufficient. The exact output is recorded in `expected-output.json`, and eleven tests cover boundary behavior. Two cases per candidate are enough to teach the program, not to establish statistical superiority or recommend a real product release.

You will leave with rubric anchors, pairing checks, gate definitions, a tradeoff table, and a decision-record template. The working question is simple: which conclusion follows from this evidence, and which preference or uncertainty still needs a human decision?

1. Begin with a release decision

State what the scorecard will decide

Begin with a sentence such as: "Choose an eligible configuration for a limited fictional support-assistant trial under the specified quality and latency requirements." That sentence identifies a decision, scope, and constraints. It is more useful than "find the best model," which leaves the workload, failure costs, and release conditions undefined.

Name the decision owner and the people who own mandatory requirements. A product owner may accept a slower experience for better difficult-case answers. An operations owner may require a bounded latency indicator. A policy owner may prohibit a specific class of action. Those choices should enter the scorecard before candidate results are inspected.

The companion's settings are `minimum_quality` and `maximum_latency_ms`, plus a critical-failure flag supplied for every run. The sample requires mean quality at least 0.75, mean latency at most 1,000 milliseconds, and zero critical failures. These are fictional teaching requirements, not generally recommended thresholds.

Separate evidence from policy

The measured or reviewed values describe candidate behavior on cases. The thresholds describe what the release owner requires. A scorecard should preserve both. If the quality threshold changes from 0.75 to 0.85, candidate A becomes ineligible even though its evidence is unchanged. That is a policy change, not a new evaluation result.

Scorecard element	Example	Who should explain it
Evidence	A quality mean is 0.80	Evaluation owner
Mandatory requirement	Quality must be at least 0.75	Release owner
Preference	Favor lower cost among acceptable options	Product or budget owner
Uncertainty	Only two synthetic cases were supplied	Evaluation owner and reviewer

Define the decision's consequence

A limited experiment, a broad production release, and an internal demonstration need different evidence. Do not reuse the same card without revisiting its scope. A small challenge set may support a focused regression decision while being inadequate for a traffic-wide quality claim.

Write the consequence in the decision record. Include the permitted users, task types, time window, and rollback or stop condition when relevant. This book does not deploy anything, but the scorecard should make the later action concrete enough to review.

Preserve the option to collect more evidence

A scorecard does not have to produce a winner. It can report no eligible candidates, several unresolved tradeoffs, or insufficient coverage. Those are useful outcomes. Forcing a total rank can make uncertain evidence look decisive and encourage a release simply because one row happens to be first.

Exercise 1

Write a release-decision sentence for a fictional invoice assistant. Name one mandatory requirement, one preference, and one condition under which the correct scorecard outcome would be "collect more evidence."

Continue with the full guide

This free sample ends after the introduction and Chapter 1. The complete guide contains nine teaching chapters, worked exercise solutions, primary references and a separate Python companion project. The full PDF, reflowable EPUB and project files are supplied together.

Use the complete guide to work through the remaining exercises and inspect the supplied examples. The companion code and fictional data are separately MIT licensed; the complete ebook has the single-reader terms described on its product page.

The examples are teaching fixtures, not production customer results. Independent human technical, editorial and assistive-technology reader review remains pending. Software access, subscriptions, consulting and guaranteed results are not included.

[View the complete guide and purchase options.](#)

You may share this labelled free sample unchanged. Keep the full purchased PDF and EPUB private. Thank you for reading.