Building Diagnostic Analytics to Focus Applied Science Teams

A Framework for Problem Decomposition and Resource Allocation in Marketplace Systems

Anjan Goswami

Abstract

Applied science teams in marketplace and discovery systems face a persistent resource allocation problem: dozens of simultaneous failure modes compete for engineering attention, and the most technically interesting problems are rarely the highest-impact ones. This paper presents a diagnostic analytics framework that decomposes system failures into quantified root causes, separates business problems from technology problems, and uses that decomposition to drive team structure and project selection. The framework was validated at a major e-commerce retailer, where it revealed that over 60% of search failures were supply-side problems invisible to traditional ML metrics, ultimately contributing to +23% revenue lift and +17% conversion improvement. We describe the framework's four components—evaluation infrastructure, root cause taxonomy, exploration budget management, and demand intelligence—and discuss how each generalizes beyond the original setting.

1 The Resource Allocation Problem in Applied Science

Every applied science leader faces the same question: *given N engineers and scientists, which problems should they work on?*

In marketplace systems—e-commerce search, content recommendation, job matching, ad ranking—this question is especially acute. A search engine can fail for dozens of reasons: poor ranking models, vocabulary gaps, missing inventory, pricing mismatches, catalog errors, intent misunderstanding, feedback loops, cold-start problems. Each failure mode has a different prevalence, a different business impact, and a different intervention. Some require ML. Some require data engineering. Some require action from entirely different teams—merchandising, supply chain, marketing—that the science leader does not control.

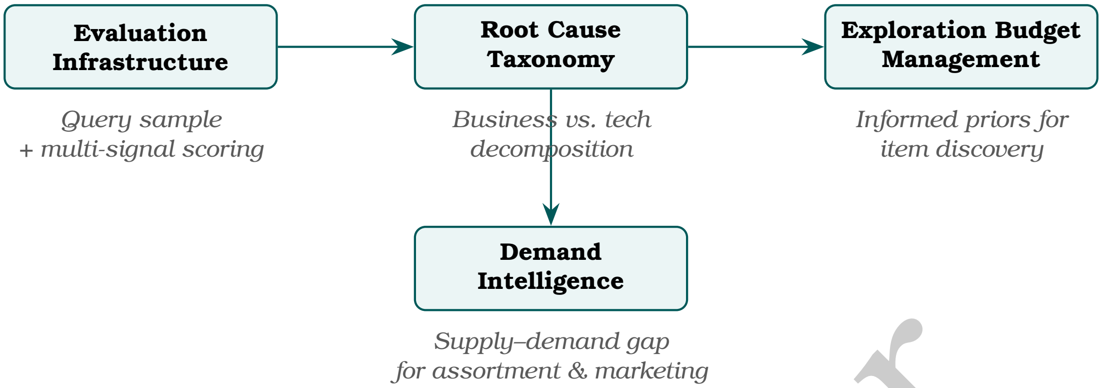
Without a rigorous diagnostic framework, science teams default to one of three heuristics:

1. **Technical interest**: Teams work on problems that are intellectually challenging or publishable. This maximizes researcher satisfaction but not business impact.
2. **Loudest voice**: Teams work on whatever problem the most senior stakeholder is currently frustrated about. This maximizes political responsiveness but not systematic coverage.
3. **Model-first**: Teams build better ranking/recommendation models and assume improved models will fix everything. This is the most common default and the most expensive mistake.

The framework described here replaces these heuristics with measurement. The core claim is simple: **the science leader's highest-leverage activity is not model selection—it is problem decomposition.** If you correctly identify *which* problems to solve and *in what proportion*, even straightforward methods produce outsized business impact. If you misidentify the problems, no amount of modeling sophistication will help.

2 Framework Overview

The diagnostic framework has four components, each producing a specific output that feeds into team prioritization:


Evaluation Infrastructure
Query sample + multi-signal scoring
Root Cause Taxonomy
Business vs. tech decomposition
Exploration Budget Management
Informed priors for item discovery
Demand Intelligence
Supply–demand gap for assortment & marketing

Each component is described below with its design principles, implementation patterns, and the specific output it produces for team prioritization decisions.

3 Component 1: Evaluation Infrastructure

3.1 Design Principle

Before you can prioritize, you need a stable measurement surface. The evaluation infrastructure answers: *what fraction of user experiences are failing, and how do we detect them reliably?*

Most teams rely on a single signal—typically an offline metric like NDCG or an online metric like conversion rate. Single-signal evaluation is insufficient because different failure modes manifest in different signals. An assortment gap produces low conversion but may show "relevant" results from adjacent categories (satisfying a relevance judge). A ranking failure produces low NDCG but may still convert if the right item appears somewhere on the page.

3.2 Multi-Channel Detection

The framework uses three independent channels:

- **Relevance judgments**: Crowdsourced evaluation of top results on a graded scale. Captures perceived quality independent of behavioral outcomes. Detection target: queries where at least one top result is marginally relevant or worse.
- **Behavioral analysis**: Queries in the bottom quintile for conversion (or click-through, or add-to-cart) flagged independently of relevance scores. Captures cases where results *appear* relevant but fail to drive action—often a signal of pricing, availability, or assortment problems.
- **User feedback**: Direct signals from feedback channels. Sparse and biased but captures frustration modes invisible to both relevance judges and behavioral data (e.g., "I searched for X and couldn't find it").

The union of these channels identifies a broader set of failures than any single signal. The intersection validates severity: queries flagged by all three channels are high-confidence failures.

3.3 Sampling Design

The query sample must be **representative** (reflecting actual traffic distribution), **stable** (maintained over multiple quarters for longitudinal tracking), and **sized for power** (large enough to detect meaningful differences in each root cause category). Traffic-based stratified sampling from 3–6 months of search data typically produces a sample in the low thousands of queries that satisfies all three requirements.

3.4 Output

A scored set of queries, each labeled as performing or underperforming, with the specific failure channel(s) that flagged it. This becomes the input to root cause analysis.

Implementation note: At one major retailer, approximately one in four queries were confirmed as poorly performing after expert review of crowd-flagged candidates, and these queries represented a disproportionate share of total search traffic—demonstrating that failures are concentrated, not uniformly distributed.

4 Component 2: Root Cause Taxonomy

4.1 Design Principle

The single most valuable analytical step is classifying every failure by its root cause across **two independent dimensions**: is this a business problem or a technology problem?

This sounds simple. In practice, it is rarely done, because it requires the science team to acknowledge that many failures are outside their scope—an uncomfortable conclusion when the team is under pressure to demonstrate impact.

4.2 Dual Taxonomy Construction

Each underperforming query is classified across:

- **Business issue categories**: assortment gaps (product not carried), pricing mismatches, availability/out-of-stock, channel routing issues (e.g., store-only intent), demand generation opportunities.
- **Technical issue categories**: concept detection, synonym matching, attribute understanding, query classification, baseline ranking, query normalization, result diversification, retrieval failures.

The dual classification forces the critical question: *if we built the perfect model, would this query be fixed?* If the answer is no—because the product doesn't exist in the catalog, or the price is wrong, or the item is out of stock—then no amount of ML investment will help.

4.3 The Typical Finding

In every marketplace system where we have applied this framework, the business-side share is larger than the technology-side share. The specific ratio varies by system maturity:

4.4 Output

A quantified decomposition: each root cause category has a measured prevalence and an estimated traffic impact. This becomes the team's project roadmap. Projects are ranked by the traffic-weighted prevalence of the root cause they address, not by technical novelty or stakeholder enthusiasm.

System Maturity	Business Problems	Tech Problems
Early-stage marketplace	60–70%	30–40%
Mature marketplace, early ML	50–60%	40–50%
Mature marketplace, mature ML	40–50%	50–60%

Table 1: Typical root cause distribution by system maturity. Business-side problems dominate in less mature systems, narrowing as ML capabilities improve and assortment processes are optimized.

4.5 Routing Non-Technical Problems

For the business-side problems, the science team's role shifts from *solving* to *enabling*. The team builds the analytics and intelligence that merchandising, supply chain, and marketing teams need to act. This requires Director-level relationships across organizational boundaries—the science leader must be credible with non-technical executives and able to translate root cause data into actions those teams can take.

5 Component 3: Exploration Budget Management

5.1 Design Principle

In any ML-driven discovery system, behavioral training data reflects the system's *past* ranking decisions. Items that rank well accumulate clicks and purchases, which train the model to rank them higher. Items never shown never generate signal. The system converges on a local optimum.

This is not a hypothetical concern. In a typical e-commerce system, a small fraction of items generates the vast majority of revenue. This concentration may reflect genuine demand—or it may reflect the feedback loop's self-reinforcing dynamics. You cannot distinguish these from observational data alone. An item's low observed demand is confounded by low exposure.

Breaking this loop requires a structured **exploration budget**: a controlled fraction of traffic allocated to discovering items whose true demand is unknown.

5.2 Informed vs. Naive Exploration

The critical design decision is *what* to explore. Naive exploration—randomly promoting unexposed but relevant items—typically destroys value. Users experience lower-quality results on the explored traffic, conversion drops, and the experiment is killed.

Informed exploration uses **external priors** to select items with the highest expected value of information:

- **Competitive intelligence**: Items that perform well on competing platforms (high reviews, strong sales rank) but have low exposure on your platform. The asymmetry suggests an information gap, not a demand gap.
- **Inventory signals**: Items with high planned inventory levels and low current exposure. The supply chain has already bet on demand; search should test that bet.
- **Demand forecasting**: Predicted revenue based on item attributes (price, category, review potential, semantic similarity to high-converting items) for items without behavioral history.
- **Catalog diversity**: Product variants and base items that are suppressed due to catalog modeling errors (e.g., a variant incorrectly labeled as a base item, consuming a results slot without showing alternatives).

5.3 Results Pattern

The consistent empirical finding across our implementations:

Exploration Strategy	**Typical Outcome**
Random relevant items	Negative revenue impact (often -3% to -5% RPV)
Inventory-informed	Modest positive lift (+1% to +2%)
Competitive intelligence	Strong positive lift (+5% to +8%+ conversion)
Demand forecast-guided	Positive, magnitude depends on model quality

Table 2: Typical exploration strategy outcomes. The prior determines whether exploration creates or destroys value.

5.4 Output

An explore–exploit architecture with a principled allocation of traffic, a strategy for selecting which items to explore, and A/B testing infrastructure to measure the value of each strategy. The science team *owns* the exploration budget and invests it where external signals predict the highest information gain.

6 Component 4: Demand Intelligence

6.1 Design Principle

Traditional assortment planning relies on historical sales data, which only captures transactions that actually happened. It is silent about demand that was never fulfilled. The demand intelligence component uses search logs to identify unmet demand *before* it shows up as lost revenue.

6.2 The Two-Sided Gap

The marketplace can be decomposed into four quadrants:

	Queries Exist	**No Queries**
Items Exist	Normal marketplace (optimize ranking)	Supply without demand (route to demand generation)
No Items	Demand without supply (route to assortment/buyers)	Blind spot (unknown unknowns)

Table 3: The two-sided demand–supply gap matrix. Each quadrant requires a different intervention.

6.3 Method: Topic Models and KL Divergence

We formalize the gap as an information-theoretic problem. Customer search queries represent a probability distribution over latent topics (the **demand signal**). The product catalog represents a separate distribution (the **supply signal**). We construct separate LDA topic models on each corpus and measure the **Kullback–Leibler divergence** between them:

$$D_{\mathrm{KL}}(P_{\mathrm{demand}} \| P_{\mathrm{supply}}) = \sum_{t} P_{\mathrm{demand}}(t) \log \frac{P_{\mathrm{demand}}(t)}{P_{\mathrm{supply}}(t)}$$

The per-topic decomposition produces a signed, ranked list of gaps. Each topic comes with its constituent keywords, making the gap directly actionable:

- **Demand ≫ supply**: keywords routed to buyers as sourcing criteria.
- **Supply ≫ demand**: keywords routed to SEM/marketing for demand generation campaigns.
- **Approximate equilibrium**: optimize ranking within existing assortment.

Items with topic overlap to query clusters but zero impressions indicate retrieval or indexing failures—a different root cause from assortment gaps, requiring a different fix.

This methodology was published at The Web Conference 2019: Goswami, A., Mohapatra, P., and Zhai, C. "Quantifying and Visualizing the Demand and Supply Gap from E-commerce Search Data using Topic Models." *WWW '19 Companion*, pp. 348–353.

6.4 Output

A continuously updated gap report: which topics have excess demand (sourcing opportunities), which have excess supply (marketing opportunities), and which have retrieval failures (engineering fixes). This feeds both the science team's technical roadmap and the business teams' assortment and marketing decisions.

7 Putting It Together: From Framework to Team Structure

The four components map directly to team structure and quarterly planning:

Component	**Team**	**Quarterly Output**
Evaluation Infrastructure	Evaluation team	Updated bad query rates, channel-level breakdown, longitudinal trends
Root Cause Taxonomy	Relevance science team	Updated root cause distribution, prioritized project backlog
Exploration Budget	Discoverability & analytics team	Exploration strategy results, enriched behavioral data, discovery dashboard
Demand Intelligence	Analytics team + cross-functional partners	Gap reports routed to buyers, SEM, and supply chain

Table 4: Mapping framework components to team structure and deliverables.

The quarterly cadence is essential. Re-evaluating the same representative sample every 3–6 months provides a stable measurement surface for tracking progress across all intervention streams. As one root cause category shrinks (e.g., assortment gaps decrease as buyers act on gap reports), the relative share of remaining categories increases, and the team's project mix shifts accordingly.

8 Validation

This framework was implemented at a major e-commerce retailer with an 80+ person search science organization. Key results:

- The root cause taxonomy revealed that **over 60% of search failures were supply-side problems**—assortment gaps and missing products—not technology problems. This finding redirected engineering effort from ranking model improvements (which could only address ~35–40% of failures) toward demand intelligence and buyer enablement.
- The exploration budget, guided by competitive intelligence priors, produced **5–8%+ conversion lifts** across multiple product categories, while naive random exploration destroyed value.
- The demand–supply gap system routed actionable intelligence to merchandising teams that had previously operated without visibility into unmet customer demand.
- The combined effect across all three intervention streams: **+23% search-attributed revenue lift** and **+17% sales conversion improvement**.
- The work produced peer-reviewed publications at WWW 2019, IEEE Big Data 2015, and SIGIR ECOM 2018, formalizing the methods for the broader research community.

9 Generalization

The framework generalizes to any system where users express intent through queries or actions and the platform matches them to a supply of items, content, or providers:

- **Job marketplaces**: Query = job seeker search; Supply = posted jobs. The gap analysis identifies roles in demand that employers aren't posting, and posted roles that no one is searching for.
- **Content platforms**: Query = user browsing/search behavior; Supply = content catalog. The exploration budget breaks the feedback loop that over-promotes viral content at the expense of niche quality.
- **B2B procurement**: Query = purchasing searches; Supply = supplier catalog. The root cause taxonomy distinguishes catalog completeness problems from matching problems.
- **Ad marketplaces**: Query = user context; Supply = advertiser inventory. The demand intelligence component identifies high-value contexts with insufficient advertiser competition.

In each case, the same diagnostic sequence applies: build multi-channel evaluation, decompose failures into business versus technology root causes, manage an exploration budget with informed priors, and build a demand intelligence layer from behavioral logs.

10 Conclusion

The most common failure mode in applied science organizations is building sophisticated solutions to the wrong problem. The framework described here—evaluation infrastructure, root cause taxonomy, exploration budget management, and demand intelligence—replaces intuition-driven project selection with measurement-driven problem decomposition.

The critical insight is not any individual method. It is the discipline of asking, before any model is built: *what fraction of our failures would a perfect model fix?* If the answer is less than half—as it typically is in marketplace systems—then the science leader's primary job is not model selection. It is building the analytics that make the right problems visible and routing them to the right teams.