EXECUTIVE PLAYBOOK · AGENTICMCPSTORES.COM · 2025

How to Build AI Agents Specialized in Purchase Decisions

A practical guide for developers and AI product teams building personal shopper agents — with a focus on trust, decision quality, and commercial control.

AUDIENCE

Developers · PMs · AI Architects · Ecommerce Teams

DOMAIN

AI · Ecommerce · Trust & Safety · Compliance

EDITION

2025 · English · Operational Playbook

Table of Contents

Executive Summary

A good shopping agent is not an autonomous seller. It is a decision-support system. Its value does not come from sounding fluent. It comes from understanding context, reducing uncertainty, explaining its reasoning, disclosing incentives, and knowing when not to act.

Teams building this kind of agent should work across five disciplines at once: **need capture**, **grounding on the live catalog**, **ranking with controlled diversity**, **commercial and regulatory governance**, and **operational observability**.

The central rule of this playbook is simple: **the agent proposes; the platform decides**. Any action with financial, reputational, or regulatory impact should pass through a policy, permission, and review layer aligned to the agent's autonomy level.

DOCUMENT	ORIENTATION	FORMAT
Operational playbook	AI product + engineering	Editorial for PDF

How to read this document

The first half defines design, architecture, and recommendation flow. The second half covers trust, tooling, fraud, UX failure modes, and continuous improvement. The final glossary keeps source traceability intact for later review or compliance adaptation.

Table of Contents

1. Agent Purpose and Scope Definition

An AI agent built for shopping should not behave like a generic chatbot with polished answers. Its job is not to “talk about products.” Its job is to **reduce uncertainty** and **help users make better decisions**.

That shift matters. A useful AI personal shopper does not start by selling. It starts by clarifying:

- what the user is actually trying to solve;
- which constraints matter;
- which trade-offs they accept;
- how much confidence they need before buying;
- and what information they need in order to act.

The most common temptation is to build an agent that answers any question about any product. That creates noise, vague recommendations, and false usefulness. A serious shopping agent has a defined scope:

- a specific category or a controlled set of categories;
- a reliable catalog source;
- clear recommendation criteria;
- explicit limits about what it knows and what it does not know;
- and an actionable output.

Actionable output is not just “I recommend this product.” When relevant, it should include:

- why the recommendation appears;
- how it fits stated preferences;
- a comparative alternative;
- the main downside or trade-off;
- the price range;
- and the next step.

The underlying logic is consistent with the trust and governance approach promoted by NIST: it is not enough for the system to produce an answer; the organization must incorporate reliability considerations into design, deployment, use, and evaluation [S1][S2].

In practice, the agent's purpose should be written like this:

> *"Help the user move from a fuzzy need to an explained, purchasable shortlist aligned with their constraints."*

If you cannot write the purpose in a sentence like that, you do not yet have an agent. You have a demo.

1.1 The Agent's Internal Contract

The playbook becomes much stronger when the agent is born with an explicit internal contract. The supplemental Perplexity report is right to emphasize this point: the agent should have clearly defined covered domains, excluded domains, and transparency obligations.

Example operating contract:

- primary objective: help users discover suitable products and optimize their purchase with long-term satisfaction in mind, not just average order value;
- covered domains: discovery, comparisons, compatibility, sizing, return policies, care instructions, alternatives;
- excluded domains: promising unauthorized discounts, pretending to be human, giving out-of-scope advice, bypassing store rules;
- permanent obligation: disclose that it is AI, explain its limits, and escalate to a human when needed.

This translates well into system instructions such as:

> *"You are a personal shopping agent. You do not promise coupons, refunds, or conditions not present in the catalog or explicit rules. You do not execute financially impactful actions without going through the authorization layer. You always disclose that you are AI and escalate to human support when certainty or permission is insufficient."*

2. Core Principles of a Trustworthy Personal Shopper

An artificial personal shopper does not become valuable by sounding intelligent. It becomes valuable by feeling **useful, honest, and predictable**.

2.1 Clarity Before Persuasion

Users do not want the agent to “do marketing.” They want to understand quickly whether an option fits. That requires simple language, explicit conclusions, and direct comparisons.

Best practices:

- say first who the product is for;
- then say when it is not a good fit;
- separate facts from inferences;
- and summarize in plain language.

2.2 Constraint-Based Recommendation

Most bad recommendations come from poor context capture. Before ranking products, the agent should identify hard and soft constraints.

Hard constraints:

- maximum budget;
- compatibility;
- size, dimensions, or format;
- delivery deadline;
- technical requirements;
- geographic availability.

Soft constraints:

- style preferences;
- price sensitivity;
- brand preference;
- durability goals;
- ease-of-use priority;
- rejection of options that are too mainstream or too premium.

2.3 Commercial Honesty

If there is commission, affiliation, sponsorship, or any incentive, that must be made clear and shown close to the recommendation. The FTC stresses that material connections should be disclosed clearly and conspicuously so that users can properly weigh the recommendation [S3]. It also warns that vague terms or disclosures hidden behind “legal” or “disclosure” links are not enough [S3].

2.4 Enough Explanation, Not Excessive Explanation

The agent does not need to reveal its full internal mathematical chain. It does need to explain enough for the user to understand:

- why this option appeared;
- which data points were taken into account;
- what the main trade-off is;
- and how to ask for a revision or adjust criteria.

The ICO guidance distinguishes between process explanations and outcome explanations. For a personal shopper, that becomes very practical:

- process explanation: who designed the system, which signals it uses, and how it is controlled;
- outcome explanation: why this specific recommendation appears in this specific case [S7].

2.5 Controlled Diversity

An agent that keeps pushing only the most popular items creates a poor, repetitive, biased experience. Research on popularity bias in LLM-based recommenders shows that this risk is real, and that it can be mitigated through deliberate design and prompting that introduces less obvious but still relevant options [S6].

2.6 Governance From the Start

NIST does not treat risk management as an appendix. It treats it as part of the system. That means designing from the beginning with explicit mechanisms for `Govern`, `Map`, `Measure`, and `Manage` [S1][S2]. In a shopping agent, that means disclosure rules, traceability, logging, evaluation, and human escalation are not things you add later. They are built into the product.

---

3. Recommended Architecture for Shopping Agents

The most useful architecture for this kind of agent is not monolithic. It is layered. Each layer solves a different job.

3.1 Interaction Layer

This is where the conversation lives. Its job is not to decide the purchase. Its job is to:

- capture intent;
- clarify constraints;
- reformulate the need;
- and maintain continuity.

Possible technologies:

- a primary LLM for dialogue;
- an intent classifier;
- category-specific question templates;
- session memory.

3.2 User Context Layer

This layer stores what the agent needs in order to personalize without inventing:

- stated preferences;
- previous purchases;
- browsing signals;
- price sensitivity;
- comparison history;
- and previous rejection reasons.

Not all memory should be permanent. It is useful to distinguish between:

- session memory;
- explicit memory authorized by the user;
- and ephemeral context signals.

3.3 Catalog and Availability Layer

Without a clean catalog, the agent becomes a fantasy recommendation machine. This layer must answer reliably:

- which products exist;
- which attributes they have;
- which variants exist;
- real stock status;

- current price;
- and relevant commercial restrictions.

Google Cloud positions search and personalized recommendations across web and mobile as part of the commerce search stack [S4]. That is useful because it separates conversation from commercial grounding.

3.4 Retrieval and Grounding Layer

This layer prevents improvisation. It should retrieve:

- candidate products;
- verified attributes;
- shipping or return policies;
- verified and aggregated reviews;
- structured comparisons;
- editorial buying guides.

3.5 Recommendation and Ranking Layer

This is where the actual decision happens. It can combine:

- hard rules;
- heuristic scoring;
- recommendation models;
- contextual signals;
- and ranking by expected value.

Amazon Personalize describes a `Next-Best-Action` approach that scores the likelihood of a user taking an action and orders results by `propensity score`, while also being able to elevate higher-value options when likelihood is similar [S5]. This fits well for agents that do not only recommend products but also next steps: compare, save, buy, request size confirmation, or review an alternative.

3.6 Commercial Policy and Safety Layer

There should be a separate layer for:

- affiliate disclosures;
- allowed claims;
- sensitive categories;
- advisory limits;

- blocking unverified claims;
- and human escalation.

Without this layer, the agent may end up making statements the brand cannot support. The FTC makes clear that you should not make claims the advertiser cannot substantiate [S3].

3.7 Observability and Evaluation Layer

Every serious agent needs instrumentation to answer:

- what it recommended;
- with which context;
- why it did so;
- what the user accepted;
- what they rejected;
- and where it failed.

The right architecture is not the prettiest one. It is the one that makes every important recommendation auditable.

3.8 Autonomy Levels

Not all shopping agents should have the same power. The supplemental report proposes a practical scale worth adopting [S8]:

- Level 1 `informing`: only answers and guides;
- Level 2 `assisting`: prepares carts, drafts, or comparisons, but the user always confirms;
- Level 3 `acting`: executes limited, rule-based actions such as applying an eligible coupon or creating an exchange within predefined thresholds.

The core rule is simple:

> *the agent proposes; the platform decides.*

That means actions with financial or regulatory consequences should not flow directly from the LLM into the payment, discount, or refund backend. They should go through a governance and authorization layer [S8].

4. Conversation Flow and Decision Structure

The biggest UX mistake in these agents is recommending too early. A useful conversation follows a structure.

4.1 Phase 1: Ground the Need

Useful questions:

- "What exactly are you trying to solve?"
- "Who is this purchase for?"
- "What is your maximum budget?"
- "What is non-negotiable?"
- "What disappointed you in similar products before?"

This is not about interrogating the user. It is about creating a purchase brief.

4.2 Phase 2: Lock Decision Criteria

The agent should translate freeform answers into operational criteria. For example:

- priority: durability;
- price tolerance: medium;
- urgency: high;
- main use: daily;
- aesthetic preference: understated;
- explicit rejection: overly premium brands.

Showing this summary back to the user improves trust:

> *"I'll prioritize durability, fast delivery, and a budget under €120. I'll avoid premium options and products with polarized reviews."*

4.3 Phase 3: Present an Explained Shortlist

Instead of dumping 10 products, it is better to offer:

- one primary recommendation;
- one balanced alternative;
- one budget option;
- one reasonable premium option, when relevant.

Each option should answer:

- what problem it solves;
- why it made the shortlist;
- what trade-off it carries;
- and who it fits best.

4.4 Phase 4: Guided Comparison

Comparison should not be just about specs. It should be about decision-making.

Good comparisons:

- “If comfort matters most, A wins.”
- “If battery life matters most, B is the better trade-off.”
- “If you want to reduce risk, C is the safer option.”

4.5 Phase 5: Action Push Without Pressure

The agent should not pretend to be neutral if the goal is purchase, but it also should not push blindly.

Good outputs:

- “If you want to minimize risk, this is the one I would buy.”
- “If you want one more comparison pass, I can make a table with only the decision-critical differences.”
- “If none of these feel right, I can search outside this shortlist.”

4.6 Phase 6: Post-Decision Learning

After a purchase or rejection, the agent should capture:

- which criterion mattered most;
- which objection blocked the decision;
- whether the issue was price, trust, or fit;
- and which answers created friction.

That learning is often more valuable than many superficial metrics.

5. Data, Catalog, and Memory: What the Agent Needs to Know

The quality of a personal shopper depends less on the model and more on the data system.

5.1 What Data Is Essential

Minimum viable:

- structured catalog;
- current price;
- stock or availability;
- category attributes;
- quality or rating signals;
- relevant purchase policies.

Ideal:

- browsing history;
- purchase history;
- returns;
- brand affinities;
- question history;
- and usage context.

Amazon Personalize highlights the importance of separating users, items, and action/interaction datasets to better understand interests and predict the next action [S5].

5.2 Which Data Should Not Be Treated as Truth

Some data helps, but should not carry absolute weight:

- raw sales volume;
- historical popularity without context;
- unverified reviews;
- unverified brand claims;
- poorly maintained internal labels.

If these signals enter without filters, they bias the system toward what creates the most noise, not toward what fits best.

5.3 Useful Memory vs. Intrusive Memory

Not everything should be remembered. It makes sense to store only what improves future decisions:

- usual size or measurements;
- durable preferences;

- important constraints;
- favorite or blocked brands;
- price thresholds.

Things you should avoid persisting automatically:

- unnecessary sensitive details;
- inferred emotional states;
- questionable preferences detected in a single session;
- or unconfirmed conclusions.

5.4 Practical Knowledge Structure

Think in four blocks:

1. `Profile`
2. `Session context`
3. `Verified catalog`
4. `Policies and rules`

This structure prevents mixing user taste with catalog facts or commercial rules.

5.5 What to Do When Data Is Missing

A good agent does not fill gaps with theatrical confidence.

It should say:

- “I don’t have a reliable signal on durability.”
- “I don’t see enough evidence to recommend between these two.”
- “I can give you a provisional recommendation, but compatibility still needs confirmation.”

Honesty about uncertainty increases credibility.

---

6. Recommendation Engine, Ranking, and Diversity

The heart of a personal shopper is not the conversation. It is the system that decides what goes up and what goes down in the shortlist.

6.1 Recommended Decision Order

1. Filter hard constraints

2. Generate candidates
3. Re-rank by fit
4. Apply commercial policy and disclosures
5. Inject controlled diversity
6. Explain the output

6.2 How to Score Without Breaking Trust

A useful score can combine:

- fit to the need;
- fit to the budget;
- compatibility;
- perceived quality;
- availability;
- delivery speed;
- expected economic value;
- and diversity relative to what has already been shown.

The problem is not using a score. The problem is hiding that a score exists and pretending the system has human intuition.

6.3 Next Best Action vs. "Top Product"

Many agents fail because they only recommend one product. But sometimes the next best action is not buying.

It may be:

- comparing two options;
- checking a variant;
- waiting for restock;
- changing price range;
- or requesting human assistance.

The `Next-Best-Action` approach from Amazon Personalize is valuable here because it orients the experience around the most likely and most valuable action for that user, not only around the item with the highest abstract score [S5].

6.4 Popularity Is Not Quality

Research on `popularity bias` warns about the classic vicious loop:

- popular products get more exposure;
- that exposure creates more interactions;
- those interactions make the system recommend them even more;
- and the long-tail catalog disappears from the experience [S6].

To reduce this:

- limit the weight of raw popularity;
- use explicit popularity-bias metrics;
- introduce exploration;
- and give the LLM diversity instructions when it acts as an explanation or re-ranking layer [S6].

6.5 Useful Diversity, Not Decorative Diversity

Useful diversity means:

- showing meaningfully different alternatives;
- covering different price levels;
- avoiding four nearly identical products;
- and making room for less obvious but relevant options.

Decorative diversity means injecting variety only to look smart. That confuses more than it helps.

6.6 Explaining the Ranking

Each output should be explainable in a summary like:

- “This is first because it fits your constraints best.”
- “This second option gives up some battery life but costs less.”
- “I included this third one to give you a less mainstream option with stronger durability reviews.”

Explanation does not only improve UX. It also helps internal debugging.

7. Trust, Transparency, and Compliance

Trust in a shopping agent is not built through friendly tone. It is built through visible rules.

7.1 Incentive Disclosure

If there is an economic relationship behind the recommendation, say it clearly, close to the recommendation, in plain language. The FTC is explicit:

- the disclosure must be clear and conspicuous;
- it must be close to the recommendation;
- and it must not be hidden behind links, ambiguous labels, or generic buttons [S3].

Useful examples:

- “We may receive a commission if you buy through this link.”
- “This recommendation includes affiliate products.”
- “Some of the options shown have a commercial relationship with us.”

Poor examples:

- “affiliate link”
- “commissionable link”
- a simple “buy now” button
- or a legal note placed far away from the recommendation [S3]

7.2 Do Not Imply False Experience

The agent should not speak as if it has personally tried the product. The FTC also makes clear that a recommendation cannot fake experience or support claims that are not backed by evidence [S3].

That means avoiding lines such as:

- “I tried it and it works great”
- “This is clearly the best on the market”
- “Most people will get this result”

if the system has no evidence to support them.

7.3 Explain Typical Outcomes, Not Outliers

If the agent uses testimonials, ratings, or extreme outcomes, it should make clear whether they represent typical performance. “Results not typical” does not solve the problem if the user still interprets the result as what they should expect [S3].

Practical application:

- do not use exceptional cases as the core argument;
- present expected ranges;
- distinguish between “featured case” and “normal result.”

7.4 User-Oriented Explanation

The ICO guidance is especially useful for shopping agents because it reminds us that explanations should not be one-size-fits-all. They should answer what the user needs to understand in context [S7].

A personal shopper should be able to offer at least six simplified explanation types:

- reason explanation: why this recommendation;
- data explanation: which signals were used;
- responsibility explanation: who is accountable for the system;
- fairness explanation: what was done to reduce obvious bias;
- impact explanation: what consequences the decision may have;
- review explanation: how to challenge the output or ask for help.

7.5 Governance and Monitoring

NIST promotes continuous governance logic, and the FTC expects reasonable training and monitoring programs when networks of third parties can generate deceptive behavior [S1][S3]. Translated into product terms:

- audit recommendations;
- review disclosures;
- limit generated claims;
- and create processes to correct errors quickly.

If you delegate part of the system to third parties, that does not remove your responsibility [S3].

7.6 Negative Prompts and Prohibited Techniques

In addition to defining what the agent should do, it is useful to explicitly define what it must not do. The supplemental report contributes a practical list of `negative policies` that works well as a policy layer:

- do not invent scarcity, urgency, or countdowns unless they come from a verified system;
- do not hide relevant costs such as shipping, required accessories, or restrictive return terms;

- do not use emotional pressure, guilt, or aggressive FOMO to close a purchase;
- do not relabel standard price as a discount;
- do not teach the user how to exploit policies, bugs, returns, or coupon systems.

One good internal policy would be:

> *"Do not use dark patterns or encourage practices that harm the store or other customers. If the user asks how to exploit failures, deceive the system, or force discounts, refuse the request, summarize the legitimate policy, and offer official support."*

7.7 Zero-Trust and Inter-Agent Interaction

If this system is going to operate in an ecosystem with other agents, plugins, or external tools, it is no longer enough to think only about classic user fraud. You have to think about inter-agent fraud. The supplemental report summarizes this risk well and proposes a useful zero-trust reading.

Recommended controls:

- strong identity per agent or service, not a single shared generic API key;
- capability manifest per tool, with scopes, limits, and short TTLs;
- micro-segmentation: the advisory agent should not directly call sensitive refund or discount tools;
- continuous behavioral monitoring;
- circuit breakers and kill switches for anomalous sessions or agents.

In practice, this protects against:

- indirect prompt injection through catalogs, reviews, or metadata;
- impersonation of external agents;
- memory or context poisoning;
- and adversarial buyer agents designed to abuse coupons, returns, or chargebacks [S8].

8. Tools, Stack, and Operational Building Blocks

There is no universal stack. There is, however, a sensible one.

8.1 Conversational Layer

Tools:

- OpenAI, Anthropic, or equivalent models;
- prompt routing by category;
- session memory;
- intent classifier;
- output guardrails.

Recommended use:

- clarify the need;
- summarize criteria;
- explain differences;
- generate comparisons;
- ask for confirmations before recommending.

8.2 Search and Commercial Grounding

Tools:

- Vertex AI Search for Commerce;
- your own indexed catalog search engine;
- vector store for guides, policies, and editorial content.

Google Cloud frames search and personalized recommendations for websites and mobile apps as a central part of the commerce stack [S4]. That works especially well when you want to mix search, filters, and personalization.

8.3 Recommendation and Ranking

Tools:

- Amazon Personalize for `Next-Best-Action` or ranking;
- internal scoring models;
- category-specific rules;
- LLM-based reranking only after grounding.

Best practices:

- use the LLM for interpretation and explanation;
- use more deterministic systems for hard ranking;

— and never let the model invent candidates outside the catalog.

8.4 Commercial Policy and Compliance

Necessary pieces:

— disclosure service;

— allowed-claims library;

— prohibited-language validator;

— sensitive-category detector;

— human escalation flow.

8.5 Evaluation and Observability

Tools:

— per-conversation traces;

— logging of shortlist and final ranking;

— dashboard for acceptance and rejection;

— A/B tests;

— popularity-bias monitor;

— and manual review of critical cases.

Amazon Personalize documentation explicitly mentions impact evaluation through `metric attribution` and A/B testing [S5]. That is a practical hint: do not measure only CTR. Measure real decision impact.

8.6 Minimum Viable Stack

For a serious first launch:

— 1 LLM for conversation;

— 1 clean catalog index;

— 1 rules and scoring engine;

— 1 disclosure layer;

— 1 evaluation dashboard;

— 1 human fallback.

That is worth more than a “brilliant” architecture with five agents nobody can audit.

8.7 Useful Tools and Patterns for This Category

The supplemental report also contributes a useful tool taxonomy [S8]. Ordered by role, it looks like this:

- plug-and-play platforms: Gorgias AI Agent, Alhena AI, Zowie, Kustomer, Fin;
- headless search or recommendation engines: Algolia, Doofinder, Nosto, Klevu, Wizzy;
- cloud stacks for custom agents: OpenAI, Anthropic, Bedrock, Vertex;
- analytics layer: observability and revenue uplift tooling for assisted sessions.

The right decision is usually:

- use headless engines as `tools` for the agent, not as the agent itself;
- use the LLM to interpret, ask questions, and explain;
- and use classical services for persistent tasks such as price alerts, auto-buy, or reorder workflows.

---

9. What to Avoid and the Mistakes That Destroy Credibility

This section should be read before a single line of code is written.

9.1 Recommending Without Clarifying Context

Mistake:

- recommending from the very first question.

Consequence:

- lower relevance;
- more rejection;
- false personalization.

9.2 Treating Popularity as the Dominant Signal

Mistake:

- confusing what is most viewed with what is most suitable.

Consequence:

- recommendation bubbles;
- long-tail disappearance;
- worse discovery;

— and lower perceived value [S6].

9.3 Pretending to Be Neutral When There Is Incentive

Mistake:

— hiding affiliation, sponsorship, or preferred-margin logic.

Consequence:

— trust loss;

— legal risk;

— and perceived manipulation [S3].

9.4 Making Unsupported Claims

Mistake:

— “this product always lasts longer”;

— “this is the best option for everyone”;

— “you will get exactly this result.”

Consequence:

— regulatory exposure;

— and credibility erosion [S3].

9.5 Relying on Biased or Manipulated Reviews

Mistake:

— using inflated reviews;

— incentivizing only positive opinions;

— delaying or hiding negative ones.

Consequence:

— biased outputs;

— misleading product perception;

— worse user decisions [S3].

9.6 Not Offering an Exit When Confidence Is Low

Mistake:

— always answering with certainty.

Consequence:

— the agent looks smart until it makes a serious mistake.

A good agent knows how to say:

— “I don’t have enough evidence”;

— “I need to confirm compatibility”;

— “this is a case where human review is better.”

9.7 Turning Explanation into Smoke

Mistake:

— using technical language to disguise weakness.

Consequence:

— the user does not understand;

— the internal team cannot debug either;

— and the recommendation stops being defensible.

9.8 Building Without an Evaluation System

Mistake:

— launching and measuring only superficial engagement.

Consequence:

— you optimize conversation, not decision quality;

— and you never learn whether the agent actually helps users buy better.

9.9 Failure Modes That Trigger Cart Abandonment

The supplemental report contributes a particularly useful collection of recurring failure modes because it ties the problem directly to UX and revenue. These should be made explicit in the playbook:

— product hallucinations: wrong price, availability, or specifications;

— context amnesia: forgetting constraints or prior sessions;

— broken personalization: recommending irrelevant or already-owned products;

— commercial aggressiveness: intrusive popups, forced upsells, or interruptions during checkout;

— inability to handle complex queries: failing on multi-intent requests or lacking a clear escalation path;

— manipulation via fake quality signals: overly precise metrics, synthetic reviews, or suspicious social proof.

The practical lesson is blunt:

— one serious factual error can destroy trust;

— forgetting context creates frustration;

— and pushing too early turns the agent into a nuisance rather than a helper.

9.10 Abuse Signals and Response Protocol

For fraud and abuse, the agent should not “accuse.” It should detect, add friction, and escalate. The supplemental report proposes a useful pattern:

Signals:

— multi-step operations executed too fast;

— attempts to access out-of-scope data;

— identity inconsistencies;

— systematic buy-cancel-refund abuse;

— anomalous addresses or grouped claim patterns;

— suspicious statistical precision in review or rating inputs.

Response:

1. silent flag;
2. progressive friction;
3. human review;
4. circuit breaker;
5. intelligence sharing, when applicable.

This avoids two frequent mistakes:

— blocking legitimate users too early;

— or letting coordinated abuse pass because each individual action looked valid in isolation.

10. Launch, Evaluation, and Continuous Improvement Best Practices

A personal shopper is never really “finished.” It is operationally trained.

10.1 Start With a Category That Has Real Decision Complexity

Good initial categories:

- consumer electronics;
- beauty or personal care;
- sports;
- home;
- technical fashion;
- products with many variants.

Bad initial categories:

- messy catalogs;
- categories with incomplete attributes;
- or products that are too undifferentiated.

10.2 Define Decision Metrics, Not Just Interaction Metrics

Useful metrics:

- shortlist acceptance rate;
- click-through on recommended product;
- add-to-cart after recommendation;
- assisted conversion;
- time to useful shortlist;
- rate of criteria changes;
- post-recommendation satisfaction;
- and need for human escalation.

Technical metrics:

- HR@5 and HR@10 for ranking [S6];
- popularity-bias metrics such as `log popularity difference` [S6];
- catalog coverage;
- option repetition;
- grounding error rate;
- and ratio of non-actionable recommendations.

10.3 Human Batch Review

Every week, you should manually review samples of:

- best recommendations;
- worst recommendations;
- abandoned conversations;
- cases with disclosure;
- and sensitive categories.

Review questions:

- Was the recommendation defensible?
- Was the disclosure visible?
- Was there a better alternative that was not shown?
- Did the agent correctly understand the main constraint?
- Did the explanation actually help?

10.4 Improve in Layers

Recommended optimization order:

1. catalog quality
2. need capture
3. hard rules
4. ranking
5. explanation
6. memory
7. fine-grained experimentation

Most teams try to improve prompts when the real problem is data or catalog quality.

10.5 Design an Elegant Human Fallback

Fallback should not feel like a system error. It should feel like a responsible decision.

Example:

> *“I see two very close options, and one key constraint still needs confirmation. If you want, I can keep the current shortlist or escalate this to a human advisor.”*

10.6 Document the System So It Can Be Explained

The ICO stresses the need for documentation, traceability, and accessible explanations for AI-assisted decisions [S7]. In a shopping agent, that means documenting:

- which signals are used;
- which signals are not used;
- how prioritization works;
- when disclosure is mandatory;
- when a response is blocked;
- and who is accountable for the system.

10.7 Final Rule

If the agent sells more but leaves the user with less clarity, you did not build a good personal shopper. You built a layer of commercial pressure.

If the agent helps users understand better, compare better, and buy with less regret, then you are creating real advantage.

---

11. Source Glossary

[S1] NIST AI RMF Playbook

- Title: `NIST AI RMF Playbook`
- Link: `https://www.nist.gov/itl/ai-risk-management-framework/nist-ai-rmf-playbook`
- Contribution: operational framework for `Govern`, `Map`, `Measure`, and `Manage`; trustworthiness and continuous governance.

[S2] NIST AI RMF: Generative AI Profile

- Title: `Artificial Intelligence Risk Management Framework: Generative Artificial Intelligence Profile`
- Link: `https://www.nist.gov/publications/artificial-intelligence-risk-management-framework-generative-artificial-intelligence`
- Contribution: trustworthy design, responsible deployment, and evaluation of GenAI systems.

[S3] FTC Endorsement Guides

- Title: `FTC's Endorsement Guides: What People Are Asking`

- Link: `https://www.ftc.gov/business-guidance/resources/ftcs-endorsement-guides`
- Contribution: affiliate disclosure, proximity of disclosure, prohibition of misleading claims, network monitoring, and advertiser accountability.

[S4] Google Cloud Commerce Search / Recommendations

- Title: `Vertex AI Search for commerce documentation`
- Link: `https://cloud.google.com/retail/docs`
- Contribution: search plus personalized recommendations, commercial grounding, real-time events, and web/mobile usage.

[S5] Amazon Personalize

- Title: `Next-Best-Action recipe - Amazon Personalize`
- Link: `https://docs.aws.amazon.com/personalize/latest/dg/native-recipe-next-best-action.html`
- Contribution: real-time personalization, exploration, automatic updates, propensity scoring, user/item/action datasets, and impact measurement.

[S6] Technical paper on popularity bias

- Title: `Large Language Models as Recommender Systems: A Study of Popularity Bias`
- Link: `https://arxiv.org/pdf/2406.01285`
- Contribution: popularity bias, metrics such as `HR@5`, `HR@10`, and `log popularity difference`, and the use of prompting to introduce less popular but relevant options.

[S7] ICO Guidance on Explaining AI Decisions

- Title: `Explaining decisions made with AI`
- Link: `https://ico.org.uk/for-organisations/uk-gdpr-guidance-and-resources/artificial-intelligence/explaining-decisions-made-with-artificial-intelligence/`
- Contribution: outcome and process explanations, accountability, data, fairness, impact, traceability, documentation, and review pathways.

Closing Note

The safest way to build a useful shopping agent is to think of it as a decision-support system, not as an autonomous seller. When the agent understands context, explains its reasoning, discloses incentives, limits bias, and knows when to escalate, it becomes a trust tool. When it hides incentives, improvises claims, and confuses popularity with fit, it becomes a product, brand, and compliance risk.