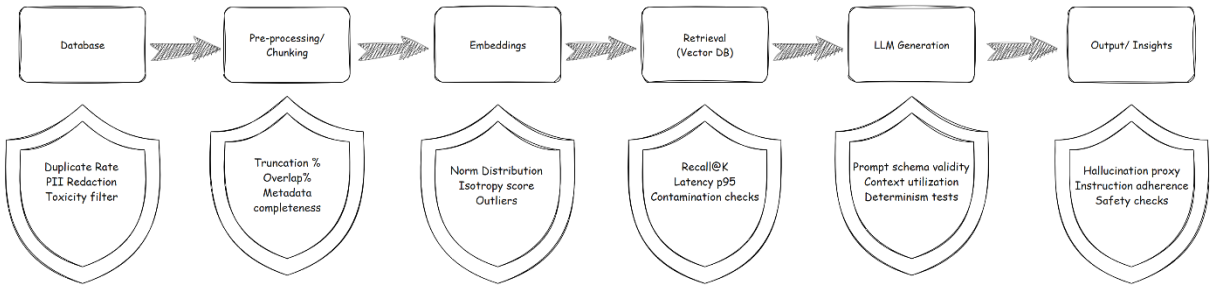
Sanity Checks in AI & GenAI

I've learned that working with data experiments and GenAI features, the difficulty isn't in building the model, rather it's in the *construct and setup*.



Here are some sanity checks that have saved me (after I got burned skipping them before):

Instrumentation glitches

In LLM and RAG systems, the risks aren't just "hallucinations" - they start earlier, in the setup. A small preprocessing bug (like dropping metadata) or a skewed embedding distribution can silently poison retrieval. That's why it's critical to track guardrail metrics:

- Latency spikes & error rates in ingestion, retrieval, and generation
- Embedding health (norm distribution, isotropy scores) to detect collapse or corruption
- Truncation & overlap rates in chunking to ensure inputs fit the context window without losing meaning
- Schema/JSON validity for prompts, tools, and function calls - even one broken template can cascade into bad outputs

External factors

LLMs and RAG systems don't live in a vacuum. Ingested data which are impacted by external factors such as holiday surge or a major product launch, can corrupt vector search indexes and distort results. If these are not accounted for, the model may "learn" short-lived anomalies as if they were permanent truths. To guard against this:

- Tag and partition data by time so seasonal spikes don't get blended in with general information
- Monitor distribution drift in queries and embeddings (KL divergence, PSI) to catch when user behavior suddenly changes
- Apply metadata filters at retrieval (e.g., "last 6 months only" or "exclude promo periods") to avoid serving outdated or anomalous context

- Maintain golden evaluation sets from stable periods to benchmark whether new data is genuinely useful or just noise

Selection bias

Just like biased training data produces biased models, retrieval pipelines can quietly skew results if the split between control and treatment queries isn't representative. In practice, this means some user groups (e.g., advanced users vs. first-timers) may see systematically different results.

- Run A/A tests on retrieval and generation - verifying that two identical setups return comparable distributions of context and output
- Segment test sets by user type, region, or intent to ensure no hidden group gets under-served
- Track recall@K per segment rather than only at the aggregate level

Sample ratio mismatches

If randomization fails, your A/B experiments on prompts, retrievers, or re-rankers will produce misleading outcomes.

- Monitor traffic splits with χ^2 goodness-of-fit tests to confirm equal routing across model variants
- Track retrieval depth parity (avg docs retrieved per arm) so you're comparing apples to apples
- Set alerts when one experiment arm drifts in query volume, latency, or error rates - often a silent infra bug, not a model difference

Novelty effect

New GenAI features (like chatbots or copilots) often see a spike in usage just because they're shiny. That bump fades, but if you don't separate novelty from sustained value, you'll overestimate impact.

- Track usage over time (day-30 retention vs. day-1 excitement)
- Segment adoption metrics (power users vs. casuals) to see who sticks around
- Compare task success rate, not just engagement - are users achieving goals more effectively, or just playing around at launch?

These sound obvious in hindsight, but I promise - skipping even one can turn a "success story" into a mirage. If you're building AI, GenAI, or just running experiments, **sanity checks are your cheapest insurance policy**