[DISTRIBUTIONAL]

Distributional's Data Pipeline

Intro

Distributional has updated and continues to update our core functionality.

See docs.dbnl.com for the latest on our data pipeline, analytics workflow, and enterprise platform.

Production AI behavior is a black box, which leaves AI teams struggling to improve, fix, and scale their products. Distributional's adaptive analytics platform empowers these teams to have confidence in AI behavior—the interplay and correlations between users, context, tools, models, and metrics. The product (DBNL) provides a detailed snapshot of aggregate AI product behavior, regularly surfaces behavioral signals, links evidence to facilitate deep root cause analysis, and enables teams to track relevant signals over time. This empowers AI teams to better understand the behavior of their users and AI products so that they can fix and improve those products with confidence.

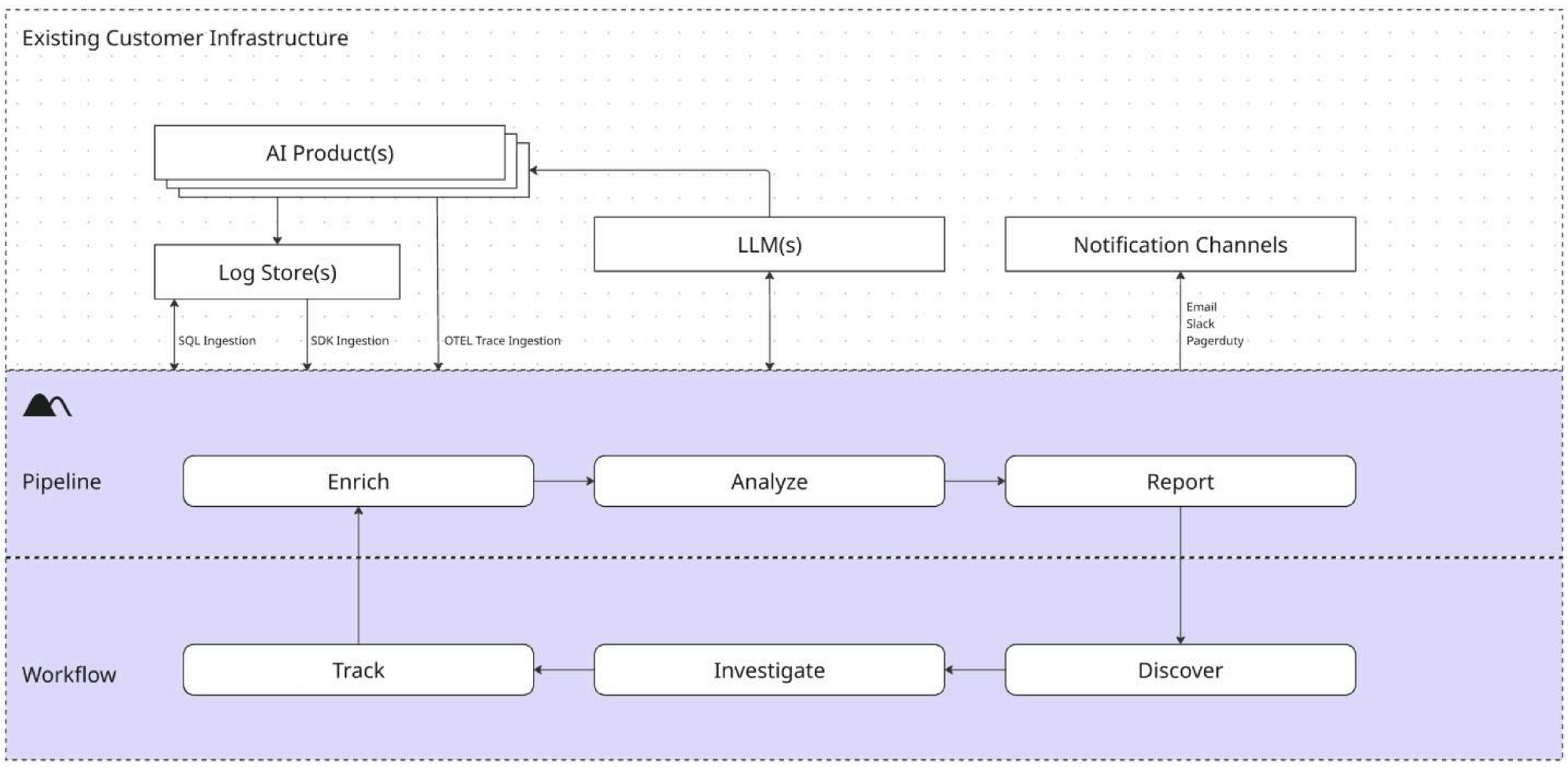
Distributional's adaptive analytics product includes an analytics workflow that is supported by a data pipeline and powered by an enterprise platform. The focus of this paper is the Distributional Data Pipeline that automatically finds behavioral signals hidden in your production AI logs in three steps: enrich, analyze, and publish.

Table Of Contents

Data Pipeline

Distributional's Data Pipeline automatically enriches production AI logs, analyzes these enriched logs to uncover behavioral signals, and publishes signals to various channels to empower efficient, effective product decision making.

If you are a team running AI in production at scale that is struggling to understand usage patterns and need systematic data analysis to make better product decisions, Distributional is designed for your circumstances. This pipeline makes it possible to uncover behavioral signals as you scale your AI products, and leverage these signals to continuously fix and improve your products as usage evolves over time.

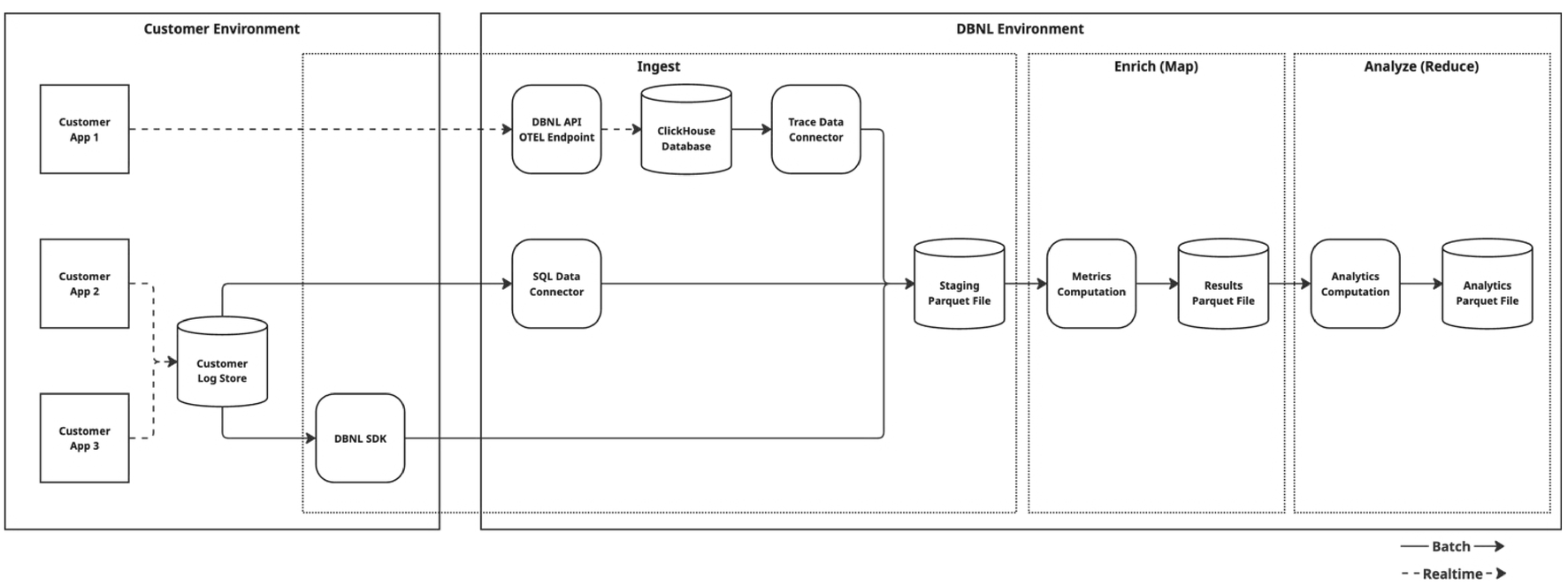


Data Model

Before diving into how the Distributional platform is designed, it's important to understand some key terms related to how Distributional implements adaptive analytics for production AI. These help to break down the types of computation in the system done to create a statistical representation of behavior for production AI.

SETUP

Distributional deploys a Data Pipeline that analyzes your production AI logs. First, DBNL ingests production AI logs with one of three methods: SDK, SQL, or OTEL. Second, DBNL enriches production AI logs with LLM-as-judge, NLP, and other behavioral Metrics. Third, Distributional analyzes these enriched production AI logs with various unsupervised learning and statistical techniques to uncover behavioral signals. Finally, this pipeline publishes insights, signals, and results to dashboards and channels.



CONCEPTS

Columns

A single Log represents production AI behavior. As part of the ingestion step, Distributional requires you to flatten each log into multiple Columns. The only required Columns for a given Log are `INPUT`, `OUTPUT`, and `TIMESTAMP`, but we recommend you include the full span from a trace and apply the DBNL semantic convention.

Insights

A DBNL Insight is a human readable explanation and quantification of behavioral signals generated from unsupervised analysis of enriched Logs as part of the Data Pipeline. Insights represent signals to investigate with the Explorer and a curated sample of Logs. Insights can relate to temporal change, interesting Segments, or newly discovered outliers. DBNL is designed to make it easy to track relevant insights as Metrics or Segments to confirm improvements and fixes over time.

Metrics

A Metric is a mapping from Columns into meaningful numeric values representing cost, quality, performance, or other behavioral characteristics. DBNL computes Metrics daily as part of an enrichment step that makes deeper analysis of these Logs possible. DBNL comes with pre-built LLM-as-judge and standard Metrics. DBNL also makes it easy to edit one of these templates to produce a custom Metric, or apply a function with our query language. These Metrics appear in the Logs, Explorer, and Dashboard in DBNL.

Segments

Segments are saved filters on Logs corresponding to a specific behavioral signal discovered manually or from an DBNL Insight. All Segments are computed and published to the Segments Dashboard. When a new Segment is saved, the DBNL Data Pipeline incorporates that into future analysis as a signal this is a meaningful bifurcation of the data.

Semantic convention

DBNL ingests data using traces produced using telemetry frameworks with different semantic conventions as well as tabular Logs with a user defined format. To compute Metrics and derive Insights consistently across different data ingestion formats, we define a semantic convention for the data as stored within DBNL. If you are using OTEL Trace Ingestion and the OpenInference semantic convention, this happens automatically. If you are using SDK Log Ingestion or t, you need to ensure that your Column names adhere to our semantic convention for best results.

Enrich Production AI Logs

The first step in the pipeline is to augment raw, unstructured production AI logs into a variety of Metrics that can be parsed, clustered, analyzed, correlated, and judged. Distributional's pipeline automatically computes these properties off of your production AI logs. If you have Metrics you already prefer from development, these can easily be added to give you a richer picture of behavior against performance expectations. The goal is to create as rich, deep, and broad a picture of behavior as possible so you don't miss potential interesting signals.

DEFAULT METRICS

Distributional starts by enriching your production AI logs with default metrics:

- ANSWER_RELEVANCY: Determines if the INPUT is relevant to the OUTPUT
- USER_FRUSTRATION: Assesses the level of frustration of the INPUT based on tone, word choice, and other properties.
- TOPIC: Classifies the conversation into a topic based on the INPUT and OUTPUT. This Metric is created after topics are automatically generated from the first 7 days of ingested data.
- CONVERSATION_SUMMARY (immutable): A summary of the INPUT and OUTPUT, used as part of TOPIC generation.
- SUMMARY_EMBEDDING (immutable): An embedding of the CONVERSATION_SUMMARY, used as part of TOPIC generation.

CUSTOM METRICS

Distributional also makes it easy to add your own custom metrics off of editable templates and using our functions. Custom metrics are most useful to track specific KPIs, monitor quality signals, measure performance, validate requirements, or debug recurring issues. Good metrics are actionable, measurable, relevant, and consistent.

STANDARD METRICS VERSUS LLM-AS-JUDGE METRICS

Standard metrics are functions that can be computed using non-LLM methods. They can be built using the functions available in the DBNL t. Common examples are response length, input complexity, question mark detection, or string similarity.

LLM-as-judge metrics call an LLM to assess production AI logs against a specified attribute. Each LLM-as-judge metric is one of two types:

- Classifier Metric: Outputs a categorical value equal to one of a predefined set of classes.
- Scorer Metric: Outputs an integer in the range [1, 2, 3, 4, 5].

Use standard metrics when you need fast, deterministic calculations (e.g., word counts, text length, keyword matching, readability scores). Use LLM-as-judge metrics when you need semantic understanding (e.g., relevance, tone, quality, groundedness).

As you add metrics over time, Distributional's Data Pipeline automatically computes them daily from production AI logs.

Analyze Enriched Logs To Uncover Behavioral Signals

This pipeline automatically runs daily analysis of these enriched AI product logs to tease out interesting signals on AI product behavior. This pipeline applies a variety of techniques for this analysis. The pipeline classifies each trace according to topics that our pipeline has learned and defined for your team. It clusters distributions of these properties according to a variety of conventions, and correlates these clusters to identify any interesting, unexpected, or anomalous connection between properties. It runs an efficient LLM-as-judge on inputs and responses, and provides insight to any that violate a previously defined threshold or have drifted from prior behavior. And it runs change detection over time to surface Insights on any relevant shifts.

INSIGHTS

An Insight is a human readable explanation and quantification of behavioral signals generated from unsupervised analysis of enriched logs as part of the Data Pipeline. They are detected clusters defined by filters on Columns of log data that correspond to unique behavior. Most projects generate 5-20 new Insights per week. Projects with stable, consistent behavior may generate fewer Insights, while projects with volatile or rapidly changing behavior may generate more.

SEGMENTS

Segments are saved filters on logs corresponding to a specific behavioral signal discovered manually or from an insight.

You create a Segment when you've identified a meaningful behavioral pattern you want to track over time, such as:

- Error conditions: Logs containing specific error types or failure patterns
- High-value interactions: User sessions with purchases, conversions, or key actions
- Quality issues: Low-scoring responses that need monitoring
- User cohorts: Specific user groups (power users, new users, etc.)
- Performance bottlenecks: Requests exceeding latency thresholds

Once saved, Segments are automatically analyzed in future pipeline Runs, generating dedicated Metrics and appearing on Dashboards.

Publish Signals To Your Dashboards And Channels

The Distributional Data Pipeline translates behavioral signals into human readable Insights that are published in relevant dashboards and channels with supporting evidence that is linked in the Explorer and Logs tabs. This approach is designed to facilitate rapid root cause analysis and data-driven AI product decision making.

DASHBOARDS AND CHANNELS

Distributional's Dashboards are collections of histograms, time series and statistics of monitored Columns, tracked Segments, and generated Metrics for user-driven analysis. Distributional includes a few default Dashboards:

- Monitoring Dashboard: Distributional recommended graphs and statistics built from Columns and data from the DBNL semantic convention
- Agent Dashboard: Directed graph of all agent tool calls as a Sankey chart to allow for rapid exploration of tool call chains
- Segments Dashboard: Count graphs and statistics for all tracked Segments
- Metrics Dashboard: Histograms, time series and statistics of generated Metrics

Distributional's Notification Connections empower you to connect your preferred notification tools so you are informed when there are new behavioral Signals or meaningful changes to expected usage patterns. Distributional supports Slack and email with more channels coming soon.

Each day, DBNL ingests, enriches, and analyzes new Logs for your production AI, producing a fresh set of Insights. These Insights are published to both the Dashboards and channels to facilitate rapid and informed product decision making.

EXPLORER

The Explorer enables rapid analysis and triage of Segments by performing graphical and statistical comparison between different subsets of Logs over different time windows and/or filters. As new data is published to Dashboards and channels, Explorer updates as well.

There are three main types of exploration afforded by the Explorer:

- Single Segment: Quickly see all Metrics for a given time window and single filter on the Logs. This allows for an aggregate view of all Metrics.
- Segment Comparison: Compare two different filters on the Logs or a filter and its complement across the same time window. This allows for comparison of Metrics between filters or between filters and the rest of the Log data.
- Temporal Comparison: Compare a single filter across two adjacent time windows. This allows for a Metric comparison of before/after for a given Segment.

When investigating an issue, start with the time series to identify when it started, then use the histogram to understand what values are problematic, and finally check the Logs page to see which specific Logs exhibit the behavior.

LOGS

Similar to the Explorer, as new insights are published to the Dashboard and channels, the Logs are also updated, and specific Logs are linked as evidence to these fresh insights.

The Logs page allows the user to inspect specific Logs with certain properties defined by

- A specific time window (default: last 7 full days of data)
- Specific filters on Columns or Metrics (default: no filters)

Typically, the Logs page is visited as part of investigating a specific Insight or by clicking on part of a chart from a Dashboard, in which case the filters and time window will already be applied.

Individual Logs can be viewed in a variety of ways:

- Detailed View: All Columns and Metrics of the Log viewed together and optionally expanded.
- Trace View (if spans provided): The waterfall trace view of latency and timing for each individual span.
- Session View (if session_id provided): All associated Logs for the given session, along with Metrics.

As part of inspecting the Logs, the user can view the filtered Logs as charts and tables in the Explorer and save the specific filters as a Segment to publish it on the Segments Dashboard.

Check On Status

The Status page shows you all ongoing and previous DBNL Data Pipeline Runs for your project. These Runs represent the entire Data Pipeline, including:

- Data ingestion from the specified Data Connection for the Project
- Log enrichment by appending Metrics using the Model Connection
- Analysis and publishing of Insights

You can think of the Data Pipeline as having stages with typical durations:

- Ingest (10-30 seconds): Upload and validate data
- Enrich (60-80% of total time): Compute Metrics using Model Connection
- Analyze (10-20% of total time): Run unsupervised learning algorithms
- Publish (30-60 seconds): Update Dashboards and generate Insights

Typical pipeline run times depend on Log volume and model connection latency. Here is a rough estimate of what you'd expect based on volume of Logs:

Find Behavioral Signals In Your Production AI Logs

This pipeline automatically arms your team with a fresh set of daily signals on your AI products. These signals help your team continuously improve these products. The pipeline is designed to run this unsupervised analysis on 100% of your production AI logs at scale so don't miss any interesting signals. And it works with minimal effort from your team, so you get these signals "for free" while your team works on new AI features.

With Distributional's adaptive analytics platform, customers can complete the AI product feedback cycle. Empowered by continuous insights from DBNL, teams can understand usage patterns of their AI products, and use this information to improve and fix their products over time.

Learn more

- docs.dbnl.com
- distributional.com/blog

[DISTRIBUTIONAL]

About Distributional

Distributional's platform is designed to give customers complete control and optionality over their data and integrations. The platform can adapt to specific security, privacy, and regulatory needs, while still providing the quality and insights necessary to test AI applications.

If you'd like to learn more, reach out to the Distributional team. We're here to help!

Learn more at distributional.com

Follow us on:

- LinkedIn
- YouTube