Customer Churn Early-Warning System

Executive Summary

Customer churn is rarely a surprise; it is usually preceded by visible signals such as reduced product usage, increasing support friction, plan downgrades, and explicit renewal concerns. This project delivers a practical early-warning dashboard that brings those signals into one place so a Customer Success team can act before a customer cancels. The dashboard produces a prioritized list of at-risk customers, explains why each customer is considered risky, and suggests what to do next, such as scheduling a check-in, providing onboarding support, or conducting a pricing/value review.

To strengthen insight from customer conversations, the system can also run a local language model (LLM) on support-ticket text to detect cancellation intent and other warning signals. Because the model runs locally, ticket data does not need to be sent to the internet.

Project Overview

This project is an interactive dashboard that helps a Customer Success team identify customers who may stop using the product or cancel soon. It combines everyday business signals—subscriptions, revenue, product usage, support activity, and churn events—into a single operational view. The dashboard outputs an early-warning action list showing which customers to contact first, why they appear risky, and what actions are recommended.

Problem Statement and Goal

Customer Success teams often learn about churn risk too late—after renewal discussions fail or after usage declines to near zero. The goal of this system is to provide a consistent, explainable, and operational method to detect churn risk early and convert insights into concrete next steps. This supports two outcomes: proactive retention outreach to the right customers, and improved understanding of what drives churn risk across customer segments such as plan tiers.

Datasets Used

The system uses two categories of data: structured SaaS data and unstructured ticket text.

Structured SaaS Data (five CSV tables)

The SaaS dataset is split into five CSV files because each file represents a different business domain that typically comes from a different system (billing, product analytics, support operations, and churn tracking). Keeping them separate mirrors real-world pipelines and allows each domain to be refreshed independently while still being combined during analysis.

Accounts (accounts CSV)
This table describes who the customer is at the account level. It includes identifiers (`account_id`), descriptive metadata (`account_name, industry, country`), and lifecycle fields such as `signup_date` and `referral_source`. Fields like `plan_tier, seats, is_trial`, and `churn_flag` support segmentation and context.

Subscriptions (subscriptions CSV)
This table describes what the customer purchased and how billing works. Each subscription links back to an `account_id` and includes `start_date, end_date, plan_tier, billing_frequency, auto_renew_flag`, and revenue fields such as `mrr_amount` and `arr_amount`. Upgrade and downgrade flags are important because plan changes often precede churn.

Usage (feature usage CSV)
This table captures product usage over time at the subscription level. It typically includes `usage_date, feature_name, usage_count`, and `error_count`. This table enables trend signals such as usage drop in the most recent 30 days compared to the prior 30 days.

Churn Events (churn CSV)
This table records churn outcomes and churn explanations. Each churn event links to `account_id` and includes `churn_date, reason_code`, and additional context such as `refund_amount_usd` or preceding plan change flags. This table provides the churn label and supports churn-reason reporting.

Support Tickets (support tickets CSV)
This table captures customer friction and support experience. It links to `account_id` and includes `submitted_at, resolution_time_hours, first_response_time_minutes, satisfaction_score`, and `escalation_flag`. Ticket volume and poor support experience often correlate with churn risk.

Unstructured Ticket Dataset (ticket text CSV, optional)

A separate support-ticket text dataset can be used for LLM analysis and demos. It typically includes free-text fields such as `subject` and `body` (and sometimes tags, queue, priority, language, and version). If a true `customer_id` is not present to map tickets to the SaaS customers, the ticket dataset is treated as a demo dataset for showcasing text extraction, not as a production-grade customer linkage.

Architecture

The system architecture consists of several key components.

Data Sources
Structured SaaS data (accounts, subscriptions, usage, churn events, support tickets) and optional unstructured ticket text.

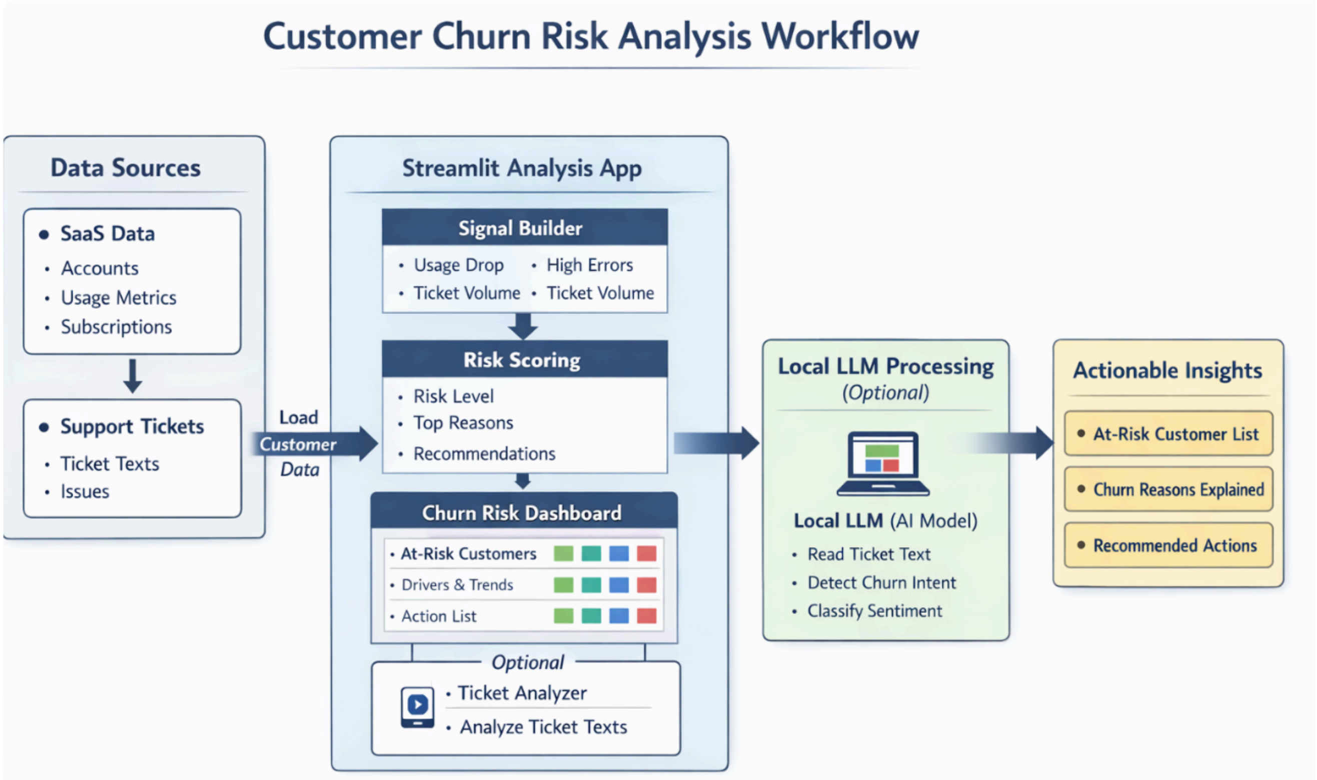
Processing Pipeline
Data loading, datatype normalization (dates and IDs), customer health metric computation, and optional LLM ticket enrichment.

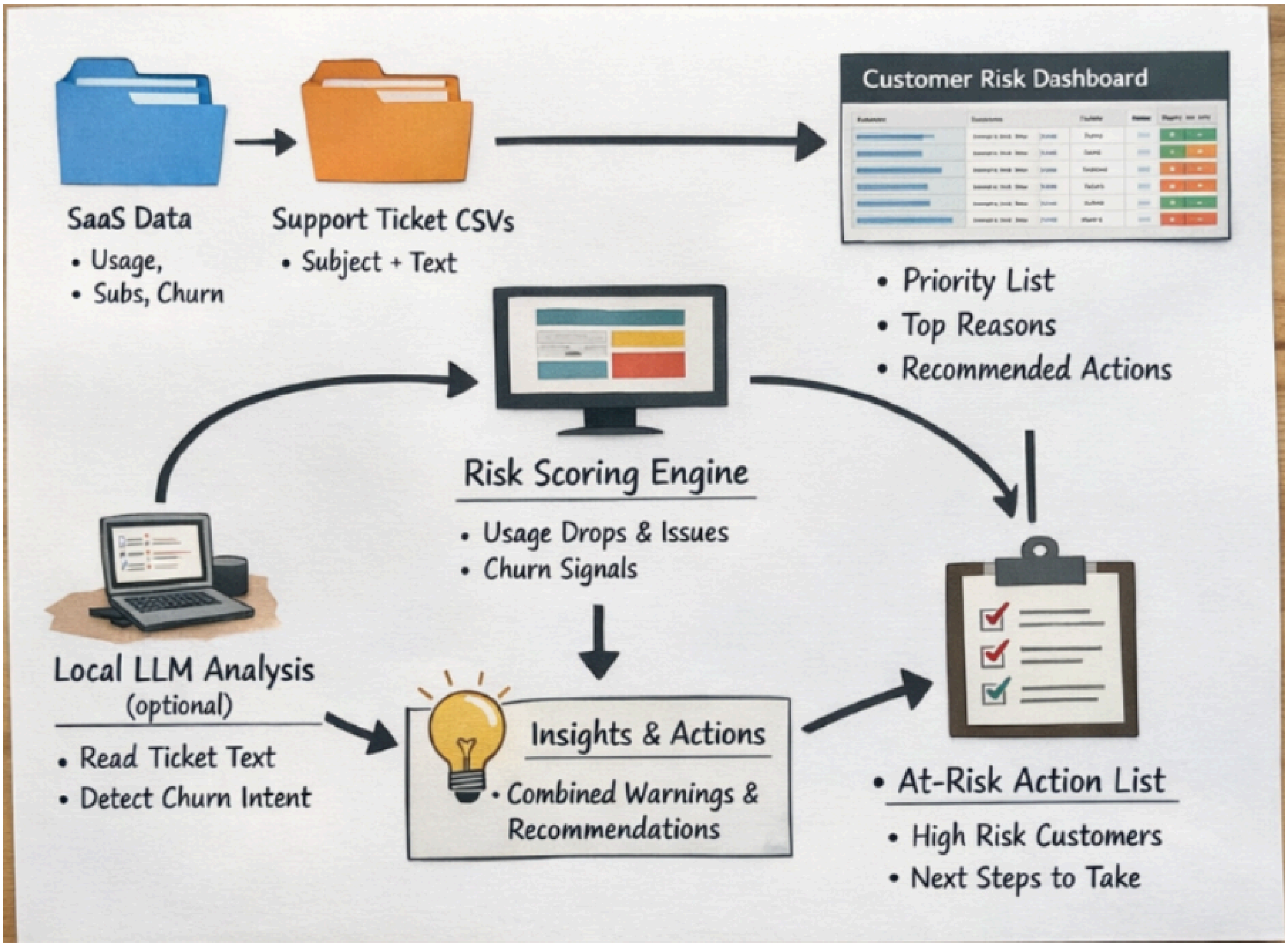
Scoring Engine
A rule-based scoring layer converts computed signals (usage drop, support load, tenure/onboarding risk, and optional churn intent from ticket text) into a risk score, a risk level, and an explanation. The output includes “top reasons” and “recommended actions” so that results are explainable and operational.

Dashboard UI
Streamlit is used to provide an interactive dashboard with four tabs: Executive, Drivers, At-Risk List, and Ticket Analyzer.

LLM Runtime (local)
A locally installed Ollama model (for example, llama3.1:8b or llama3.1:3b depending on performance needs) can enrich ticket text by producing structured JSON fields such as summary, category, sentiment, churn intent, and recommended action. This enrichment is optional because the dashboard can run fully on structured SaaS signals even without ticket text analysis.

This architecture enables integration of structured and unstructured data to provide a churn risk early-warning system with actionable insights. Figure 1 shows the end-to-end flow from raw CSVs to risk scoring and dashboard outputs.





Tools and Implementation

The system is implemented using Python. Streamlit provides the dashboard interface; Pandas and NumPy perform data loading and transformations; Ollama provides the local runtime for running the selected LLM; and CSV files act as the dataset layer for input data.

End-to-End Method (How the System Works)

The workflow begins by loading the SaaS CSV files and normalizing IDs and dates. Subscriptions are treated as "customers" at the scoring level (customer_id is derived from subscription_id), and account-level attributes (such as account_name and industry) are joined in to make the dashboard readable.

Next, customer health metrics are computed. Usage is aggregated into two windows: the most recent 30 days and the prior 30 days. This supports a directional signal—whether usage is trending down. Support tickets are aggregated for the last 30 days to reflect recent friction. Churn events are used to label accounts as churned (for evaluation and reporting).

Finally, the scoring engine evaluates each customer row and assigns a churn risk score (0–100), a risk level (High/Medium/Low), top reasons, and recommended actions. If the ticket LLM is enabled, ticket signals such as churn intent can be added as an extra input to scoring.

Risk Scoring Method (Explainable Rules)

This system uses explainable rules rather than a trained predictive model. This approach prioritizes transparency: stakeholders can understand why a customer is flagged and what triggered the score.

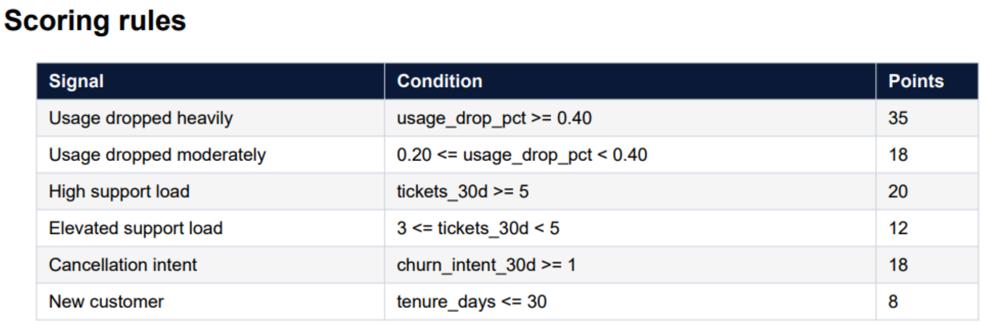
Usage trend is computed using two windows:
`usage_30d` = total `usage_count` in the last 30 days
`usage_prev30d` = total `usage_count` in the prior 30 days
`usage_drop_pct` = (`usage_prev30d - usage_30d`) / `usage_prev30d` (only when `usage_prev30d > 0`)

Support load is computed as:
`tickets_30d` = number of tickets submitted in the last 30 days (grouped by `account_id`)

Tenure is computed as:
`tenure_days` = `anchor_date - subscription start_date`, where `anchor_date` is the latest usage date available.

These signals are passed into a rule-based scoring function that assigns risk points when conditions are met. The score is capped at 100 and mapped to a risk level. The same rules generate "top reasons" and "recommended actions" that explain the score and translate it into operational next steps.

As an example, if `usage_prev30d = 200` and `usage_30d = 50`, then `usage_drop_pct = (200-50)/200 = 0.75`, meaning a 75% drop. This triggers a usage-drop risk condition, increases the risk score, and adds an explanation reason and recommended actions.



Scoring rules

Signal	Condition	Points
Usage dropped heavily	usage_drop_pct >= 0.40	35
Usage dropped moderately	0.20 <= usage_drop_pct < 0.40	18
High support load	tickets_30d >= 5	20
Elevated support load	3 <= tickets_30d < 5	12
Cancellation intent	churn_intent_30d >= 1	18
New customer	tenure_days <= 30	8

Dashboard preview

The dashboard is organized into four tabs: Executive, Drivers, At-Risk List, and Ticket Analyzer. It also describes the user controls for selecting the number of tickets to process with the LLM and setting the timeout per ticket, explaining how these settings help optimize ticket analysis workload and processing time.
Ticket data can be loaded from:

- **Folder mode:** choose a CSV from `data/raw/tickets` using a dropdown, or
- **Upload mode:** upload a ticket CSV directly from the dashboard, which is then saved locally and becomes available for processing.

This design allows quick testing of different ticket datasets without editing the source code.

SaaS dataset upload
The SaaS data model currently expects five related tables because each table contributes different signals (accounts, subscriptions, usage, churn events, support tickets). A SaaS "Upload" mode would allow replacing these files through the UI (either as five separate uploads or a zipped bundle), enabling analysts to evaluate alternate datasets without code changes.

Timeout Behavior and What It Means

The dashboard includes an LLM timeout control ("Timeout per ticket (seconds)") to prevent the application from hanging while analyzing ticket text. When the LLM is slow or the system is CPU-constrained, an output may appear like:

```
"error": "timeout_after_11s"
```

This output means the model did not return a response within the selected time limit. It is not a crash; it is a **protective safeguard**. The fields such as summary/category/sentiment remain empty because the model did not finish processing in time. Increasing the timeout, reducing "Tickets to process," or using a smaller local model typically improves completion.

LLM

The system is designed so that the churn early-warning engine works even when the LLM is disabled or timeouts occur. The churn risk scoring is primarily built on **structured signals** (usage drops, support load, churn history, tenure, plan changes). The LLM adds an additional layer: it reads ticket text to detect **language signals** such as cancellation intent, frustration, pricing pushback, or urgency.

The LLM is described as optional for three reasons:

1. **Privacy and deployment:** ticket text can be sensitive; local processing avoids sending data to external services.
2. **Reliability:** scoring can still run using only structured signals if the model is slow, unavailable, or rate-limited.
3. **Cost and compute:** local LLM inference can be heavy on CPU; structured scoring remains fast and consistent.

With LLM, the dashboard performs **automated reasoning + decision support**, not just visualization. It detects risk, explains drivers, and generates actions and outreach drafts. The LLM simply enhances the agent with natural language understanding when available.

Local LLM Runtime

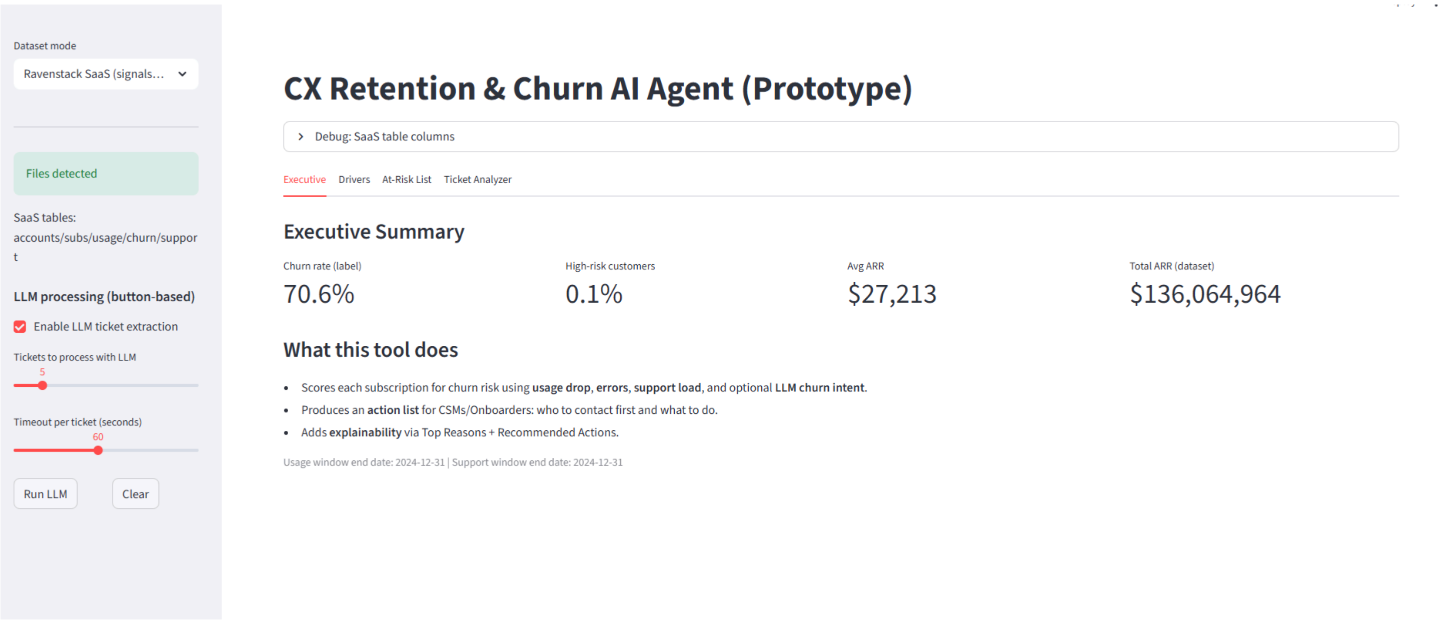
The project uses a locally installed Ollama model (for example `llama3.1:3b` or `llama3.1:8b`). This prototype was tested with llama3.1:3b for speed on CPU and llama3.1:8b for better quality when more compute is available. A local model was chosen because it is practical for demos, supports offline processing, and keeps sensitive ticket text on the same machine. The model's role is not to "predict churn" mathematically; it is to convert unstructured ticket text into structured fields such as:

- short summary of the issue,
- category (billing, bug, onboarding, performance, etc.),
- sentiment (negative/neutral/positive),
- churn intent flag,
- recommended action.

These fields can then be used as additional signals alongside usage/support metrics.

- **Executive Summary**

  The Executive tab is designed for leadership weekly review. It answers: How risky is the customer base overall, and how much revenue is potentially at risk? It summarizes churn rate (from churn labels), the percentage of customers classified as high risk, and revenue context such as average ARR and total ARR in the dataset. This tab helps stakeholders track overall retention health without reading individual customer details.



Dataset mode

Ravenstack SaaS (signals…

Files detected

SaaS tables: accounts/subs/usage/churn/suppor t

LLM processing (button-based)

☑ Enable LLM ticket extraction

Tickets to process with LLM: 5

Timeout per ticket (seconds): 60

CX Retention & Churn AI Agent (Prototype)

Executive Summary

Churn rate (label)	High-risk customers	Avg ARR	Total ARR (dataset)
70.6%	0.1%	$27,213	$136,064,964

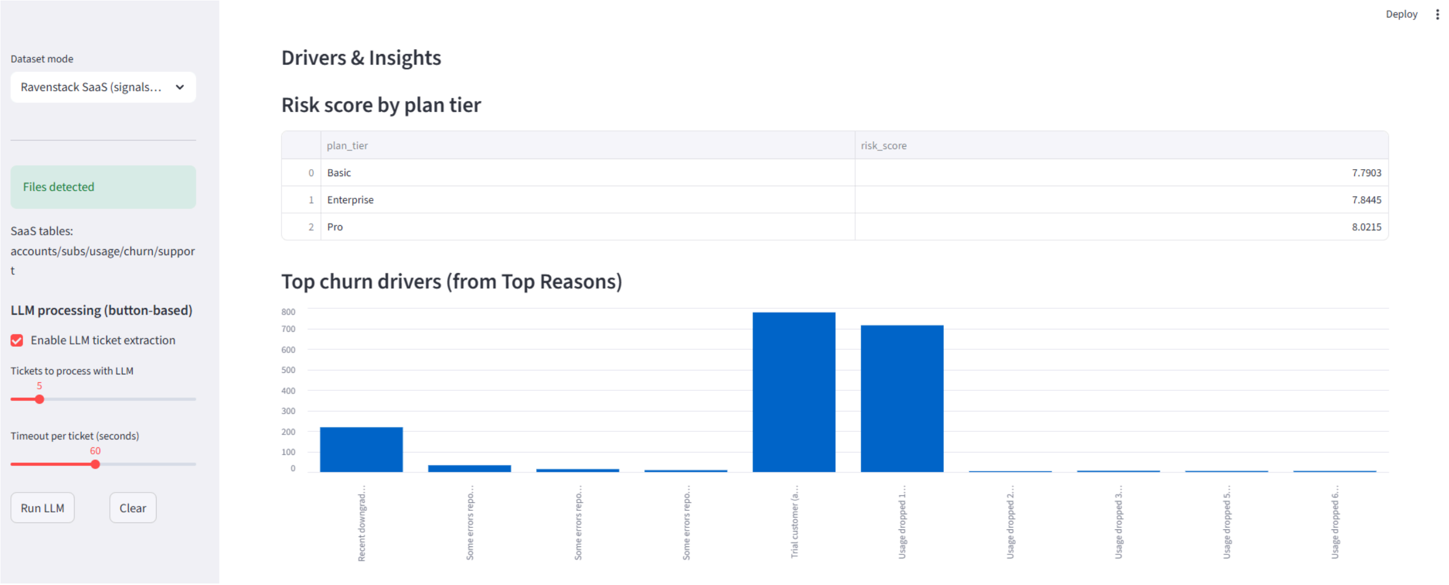
What this tool does

- Scores each subscription for churn risk using **usage drop**, **errors**, **support load**, and optional **LLM churn intent**.
- Produces an **action list** for CSMs/Onboarders: who to contact first and what to do.
- Adds **explainability** via Top Reasons + Recommended Actions.

Usage window end date: 2024-12-31 | Support window end date: 2024-12-31

- **Drivers & Insights**

The Drivers tab is designed for data analysts and CX managers. It explains why customers are being flagged so teams can focus on the biggest systemic problems (for example onboarding gaps, support quality issues, or product adoption problems). The plan tier table shows how churn risk differs across plans (Basic vs Pro vs Enterprise), which helps answer whether churn risk is concentrated in a specific segment. The churn driver chart shows how often each risk reason appears across customers; the x-axis lists driver categories (such as usage drop or support load), and the y-axis shows the number of customers flagged for each driver. The churn reasons section summarizes churn event reason codes and how frequently each appears in churn events.



Dataset mode

Ravenstack SaaS (signals…

Files detected

SaaS tables: accounts/subs/usage/churn/support

LLM processing (button-based)

☑ Enable LLM ticket extraction

Tickets to process with LLM: 5

Timeout per ticket (seconds): 60

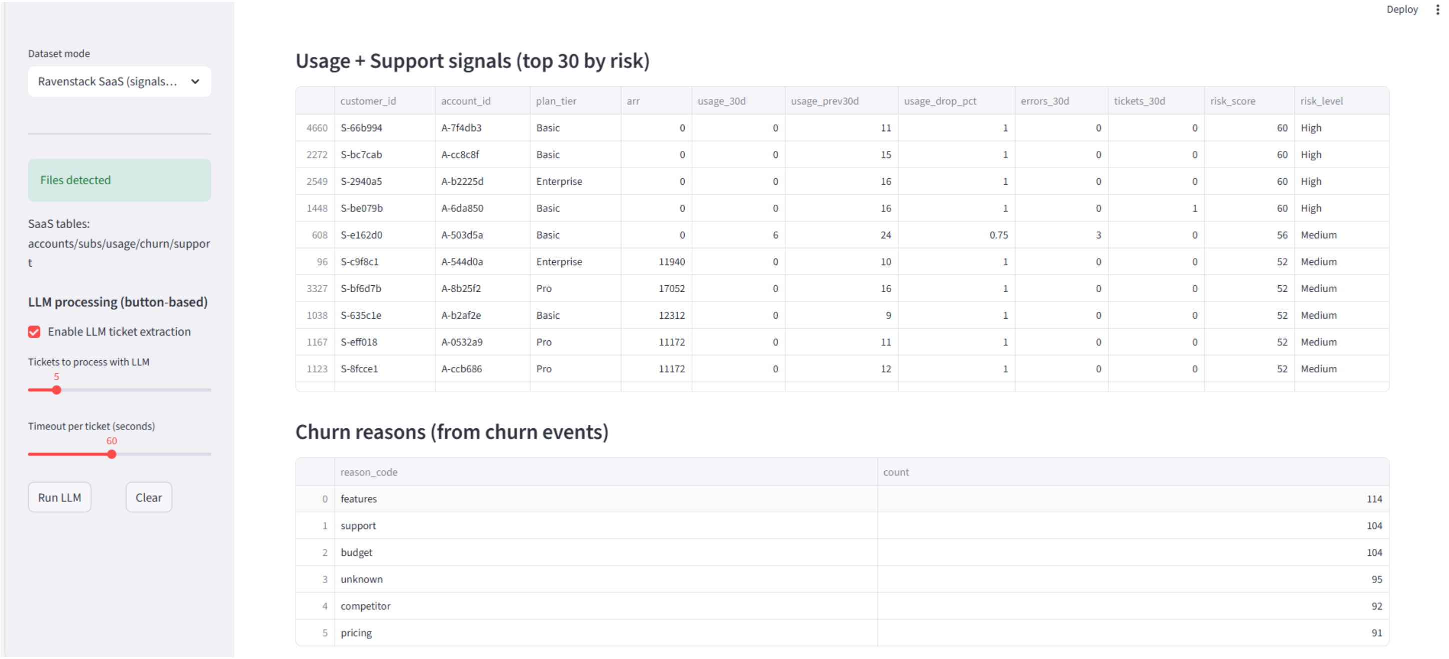
Run LLM | Clear

Drivers & Insights

Risk score by plan tier

	plan_tier	risk_score
0	Basic	7.7903
1	Enterprise	7.8445
2	Pro	8.0215

Top churn drivers (from Top Reasons)



Dataset mode

Ravenstack SaaS (signals…

Files detected

SaaS tables: accounts/subs/usage/churn/support

LLM processing (button-based)

☑ Enable LLM ticket extraction

Tickets to process with LLM: 5

Timeout per ticket (seconds): 60

Run LLM | Clear

Usage + Support signals (top 30 by risk)

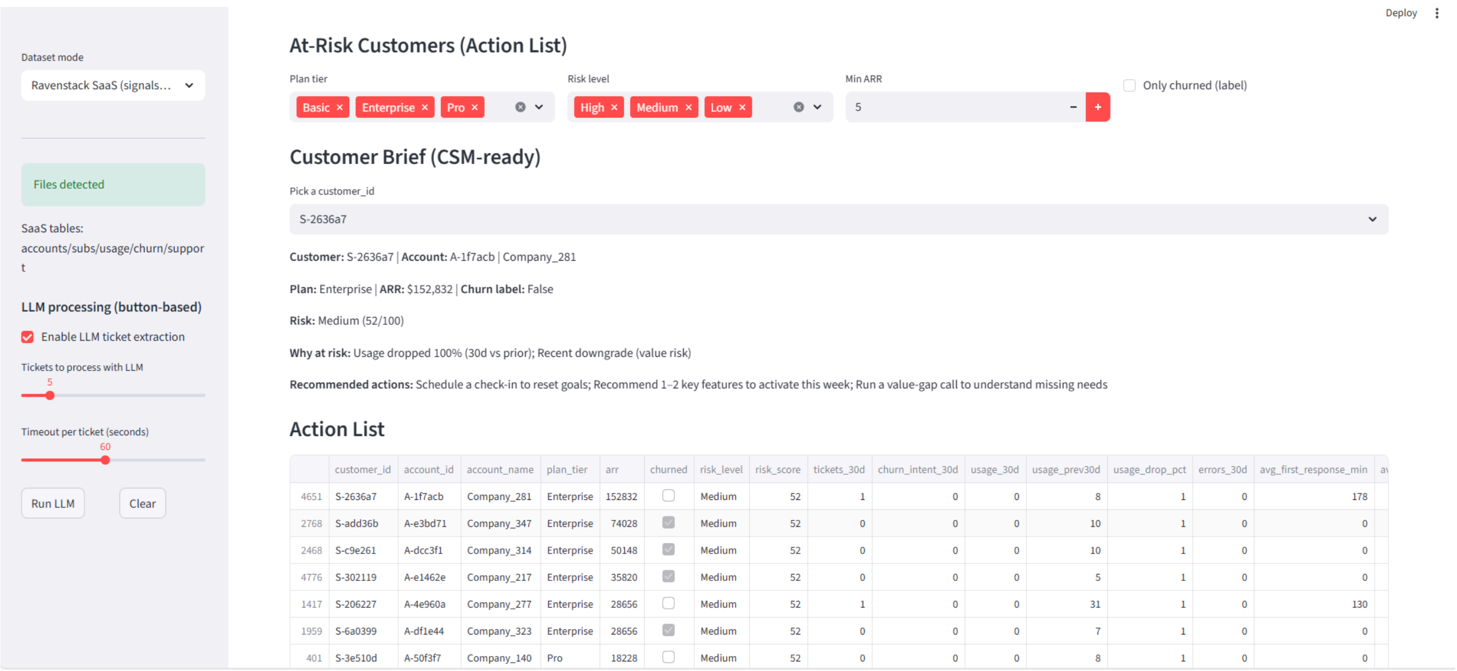
	customer_id	account_id	plan_tier	arr	usage_30d	usage_prev30d	usage_drop_pct	errors_30d	tickets_30d	risk_score	risk_level
4660	S-66b994	A-7f4db3	Basic	0	0	11	1	0	0	60	High
2272	S-bc7cab	A-cc8c8f	Basic	0	0	15	1	0	0	60	High
2549	S-2940a5	A-b2225d	Enterprise	0	0	16	1	0	0	60	High
1448	S-be079b	A-6da850	Basic	0	0	16	1	0	1	60	High
608	S-e162d0	A-503d5a	Basic	0	6	24	0.75	3	0	56	Medium
96	S-c9f8c1	A-544d0a	Enterprise	11940	0	10	1	0	0	52	Medium
3327	S-bf6d7b	A-8b25f2	Pro	17052	0	16	1	0	0	52	Medium
1038	S-635c1e	A-b2af2e	Basic	12312	0	9	1	0	0	52	Medium
1167	S-eff018	A-0532a9	Pro	11172	0	11	1	0	0	52	Medium
1123	S-8fcce1	A-ccb686	Pro	11172	0	12	1	0	0	52	Medium

Churn reasons (from churn events)

	reason_code	count
0	features	114
1	support	104
2	budget	104
3	unknown	95
4	competitor	92
5	pricing	91

- **At-Risk List**

The At-Risk tab is the operational queue for Customer Success execution. Filters narrow the list to the segment that matters (for example, only High risk or only customers above a minimum ARR). Selecting a customer shows a brief summary explaining why the customer is at risk and what to do next. This tab can also provide structured “agent-like” support such as a step-by-step next-best-action workflow, outreach drafts, and session memory notes to track follow-up outcomes. “Agent Actions” area appears after selecting a customer from the At-Risk List. It contains three practical features designed to help Customer Success teams act immediately.



Dataset mode

Ravenstack SaaS (signals…

Files detected

SaaS tables: accounts/subs/usage/churn/support

LLM processing (button-based)

☑ Enable LLM ticket extraction

Tickets to process with LLM 5

Timeout per ticket (seconds) 60

At-Risk Customers (Action List)

Plan tier: Basic × Enterprise × Pro ×

Risk level: High × Medium × Low ×

Min ARR: 5

□ Only churned (label)

Customer Brief (CSM-ready)

Pick a customer_id: S-2636a7

Customer: S-2636a7 | **Account:** A-1f7acb | Company_281

Plan: Enterprise | **ARR:** $152,832 | **Churn label:** False

Risk: Medium (52/100)

Why at risk: Usage dropped 100% (30d vs prior); Recent downgrade (value risk)

Recommended actions: Schedule a check-in to reset goals; Recommend 1–2 key features to activate this week; Run a value-gap call to understand missing needs

Action List

	customer_id	account_id	account_name	plan_tier	arr	churned	risk_level	risk_score	tickets_30d	churn_intent_30d	usage_30d	usage_prev30d	usage_drop_pct	errors_30d	avg_first_response_min
4651	S-2636a7	A-1f7acb	Company_281	Enterprise	152832	□	Medium	52	1	0	0	8	1	0	178
2768	S-add36b	A-e3bd71	Company_347	Enterprise	74028	☑	Medium	52	0	0	0	10	1	0	0
2468	S-c9e261	A-dcc3f1	Company_314	Enterprise	50148	☑	Medium	52	0	0	0	10	1	0	0
4776	S-302119	A-e1462e	Company_217	Enterprise	35820	☑	Medium	52	0	0	0	5	1	0	0
1417	S-206227	A-4e960a	Company_277	Enterprise	28656	□	Medium	52	1	0	0	31	1	0	130
1959	S-6a0399	A-df1e44	Company_323	Enterprise	28656	☑	Medium	52	0	0	0	7	1	0	0
401	S-3e510d	A-50f3f7	Company_140	Pro	18228	□	Medium	52	0	0	0	8	1	0	0

Next Best Action (workflow checklist).

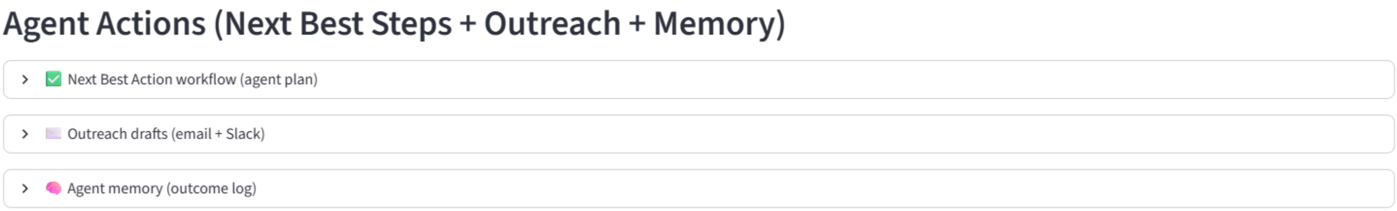
This feature generates a step-by-step plan based on the customer's risk reasons (for example: usage dropped, support load is high, onboarding risk, or cancellation intent). The plan is rule-based and produces a structured checklist such as "schedule check-in," "confirm success criteria," "activate key features," and "follow up in 7 days." This makes the system feel agent-like because it converts analysis into an executable workflow rather than leaving the user to interpret data manually.

Outreach Drafts (Email + Slack templates).

This feature creates communication drafts automatically using the customer's plan tier, ARR, top reasons, and recommended actions. It does not require the language model. The output includes an email subject, email body, and a short Slack message draft that can be copy-pasted into the team's workflow. This removes friction for Customer Success teams by shortening the time from detection to outreach.

Agent Memory (session outcome log).

This feature stores a simple follow-up record inside the Streamlit session, including an outcome (contacted, meeting scheduled, issue resolved, renewed, downgraded, cancelled, no response) plus optional notes. This acts as an "agent memory" within the session so follow-ups are not lost during the demo or review. It creates a minimal feedback loop and makes the dashboard feel like an operational tool rather than a static report.



Agent Actions (Next Best Steps + Outreach + Memory)

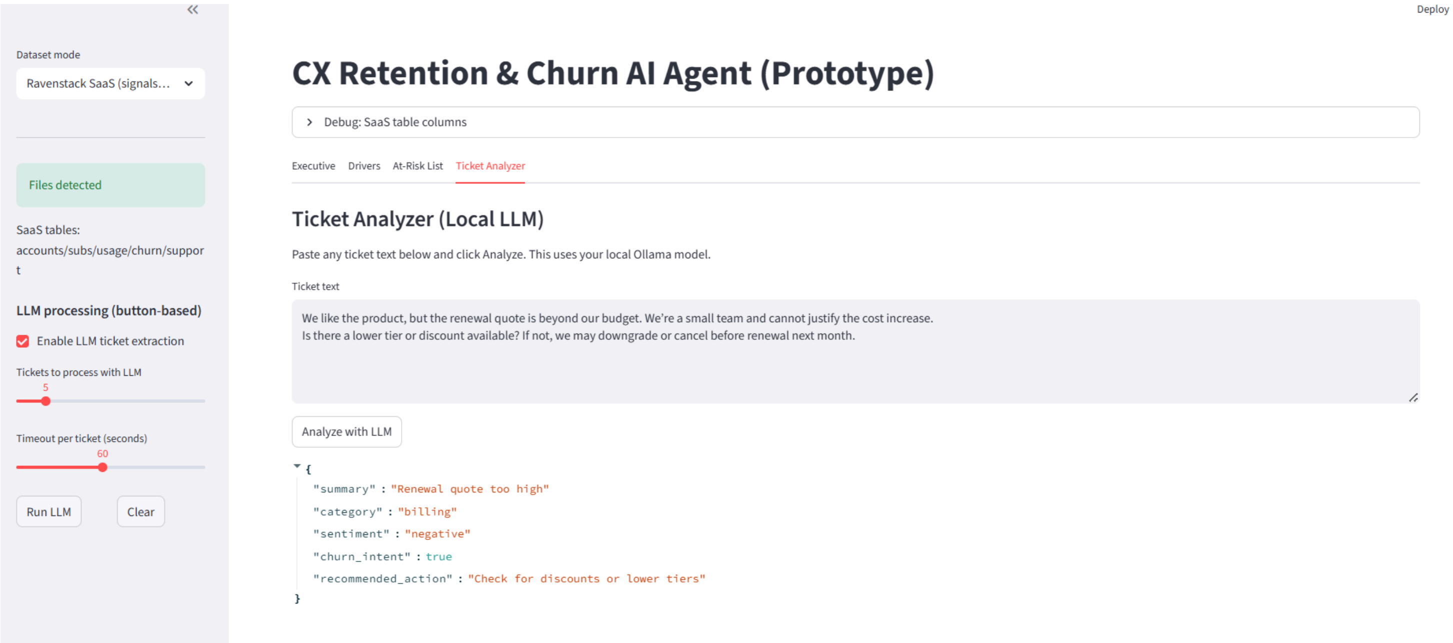
› ✅ Next Best Action workflow (agent plan)

› Outreach drafts (email + Slack)

› Agent memory (outcome log)

- **Ticket Analyzer**

The Ticket Analyzer tab runs the local LLM on a single pasted ticket and returns structured JSON. It helps validate how the model is interpreting customer language and demonstrates the text-to-signal step during presentations. For example, if a ticket says the renewal quote is too high and cancellation is being considered, the model can classify the ticket as pricing-related with churn intent and recommend a retention action such as a plan fit discussion or value review. If the model times out, the output will show a timeout error rather than freezing the app.



Dataset mode: Ravenstack SaaS (signals…

Files detected

SaaS tables: accounts/subs/usage/churn/support

LLM processing (button-based)

☑ Enable LLM ticket extraction

Tickets to process with LLM: 5

Timeout per ticket (seconds): 60

CX Retention & Churn AI Agent (Prototype)

› Debug: SaaS table columns

Ticket Analyzer (Local LLM)

Paste any ticket text below and click Analyze. This uses your local Ollama model.

Ticket text

We like the product, but the renewal quote is beyond our budget. We're a small team and cannot justify the cost increase.
Is there a lower tier or discount available? If not, we may downgrade or cancel before renewal next month.

```
{
  "summary" : "Renewal quote too high"
  "category" : "billing"
  "sentiment" : "negative"
  "churn_intent" : true
  "recommended_action" : "Check for discounts or lower tiers"
}
```

Features

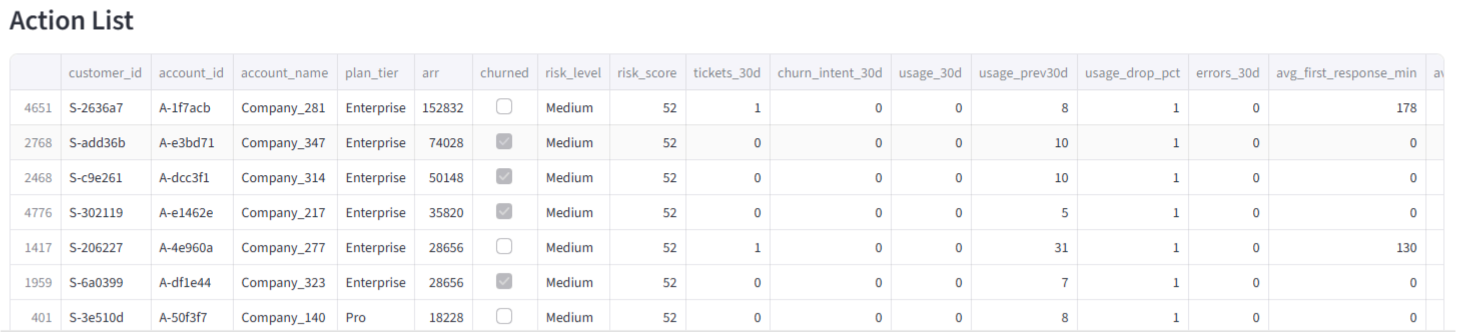
The system combines three capabilities: (1) multi-source signal integration, (2) explainable scoring, and (3) an execution-oriented dashboard.

It enables interactive filtering and sorting in the At-Risk List, supports near real-time refresh when datasets update, integrates structured metrics with optional unstructured ticket analysis, and provides risk scoring and categorization for prioritization. The dashboard design supports both high-level stakeholder reporting and day-to-day Customer Success workflows.

Outputs

The dashboard produces a churn risk score and risk level per customer, explanation text (“top reasons”), recommended actions, next-best-action workflow steps, outreach drafts (email and Slack templates), a session-level memory log for outcomes, and JSON output from the LLM.

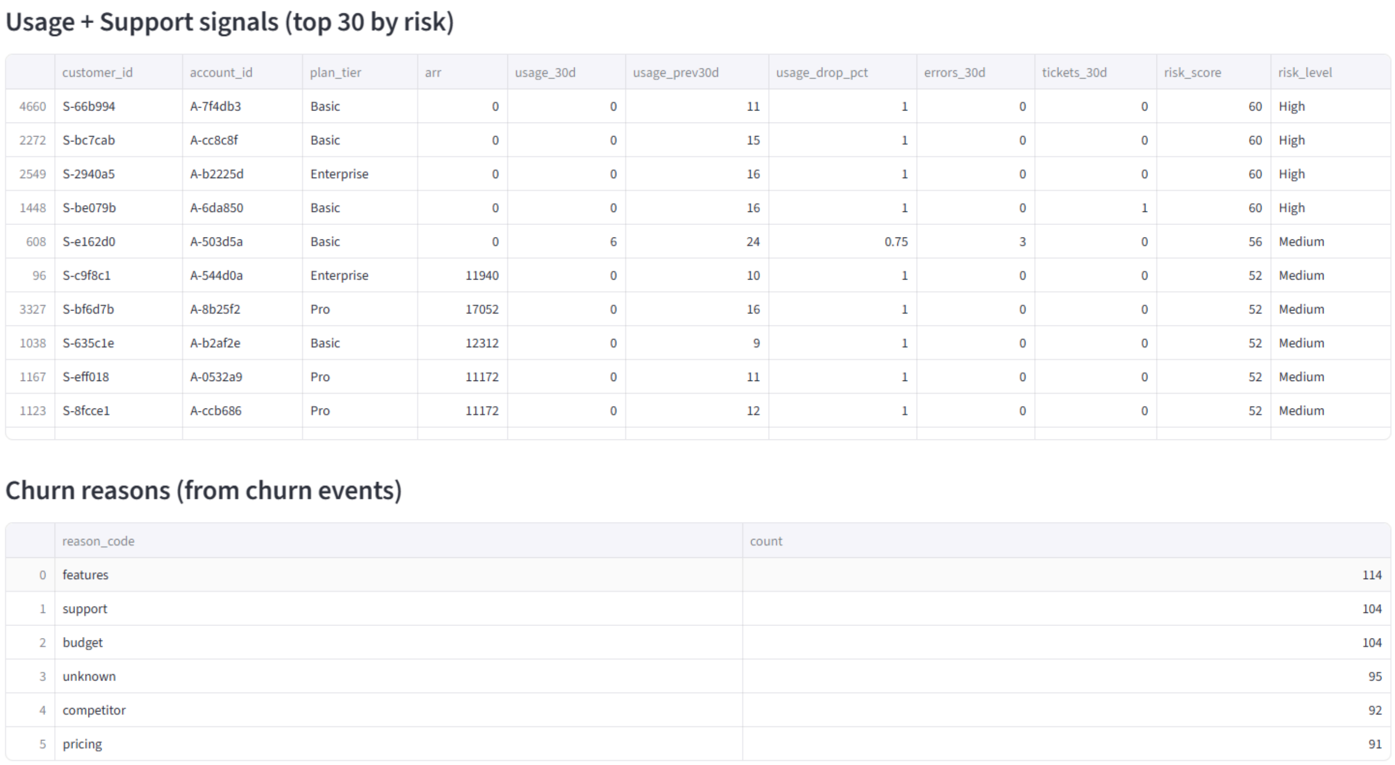
Output: prioritized at-risk customer list with explanations and recommended actions.



Action List

	customer_id	account_id	account_name	plan_tier	arr	churned	risk_level	risk_score	tickets_30d	churn_intent_30d	usage_30d	usage_prev30d	usage_drop_pct	errors_30d	avg_first_response_min	av
4651	S-2636a7	A-1f7acb	Company_281	Enterprise	152832	☐	Medium	52	1	0	0	8	1	0	178	
2768	S-add36b	A-e3bd71	Company_347	Enterprise	74028	☑	Medium	52	0	0	0	10	1	0	0	
2468	S-c9e261	A-dcc3f1	Company_314	Enterprise	50148	☑	Medium	52	0	0	0	10	1	0	0	
4776	S-302119	A-e1462e	Company_217	Enterprise	35820	☑	Medium	52	0	0	0	5	1	0	0	
1417	S-206227	A-4e960a	Company_277	Enterprise	28656	☐	Medium	52	1	0	0	31	1	0	130	
1959	S-6a0399	A-df1e44	Company_323	Enterprise	28656	☑	Medium	52	0	0	0	7	1	0	0	
401	S-3e510d	A-50f3f7	Company_140	Pro	18228	☐	Medium	52	0	0	0	8	1	0	0	

Output: Top 30 customers at risk and reason of Churns



Usage + Support signals (top 30 by risk)

	customer_id	account_id	plan_tier	arr	usage_30d	usage_prev30d	usage_drop_pct	errors_30d	tickets_30d	risk_score	risk_level
4660	S-66b994	A-7f4db3	Basic	0	0	11	1	0	0	60	High
2272	S-bc7cab	A-cc8c8f	Basic	0	0	15	1	0	0	60	High
2549	S-2940a5	A-b2225d	Enterprise	0	0	16	1	0	0	60	High
1448	S-be079b	A-6da850	Basic	0	0	16	1	0	1	60	High
608	S-e162d0	A-503d5a	Basic	0	6	24	0.75	3	0	56	Medium
96	S-c9f8c1	A-544d0a	Enterprise	11940	0	10	1	0	0	52	Medium
3327	S-bf6d7b	A-8b25f2	Pro	17052	0	16	1	0	0	52	Medium
1038	S-635c1e	A-b2af2e	Basic	12312	0	9	1	0	0	52	Medium
1167	S-eff018	A-0532a9	Pro	11172	0	11	1	0	0	52	Medium
1123	S-8fcce1	A-ccb686	Pro	11172	0	12	1	0	0	52	Medium

Churn reasons (from churn events)

	reason_code	count
0	features	114
1	support	104
2	budget	104
3	unknown	95
4	competitor	92
5	pricing	91

Limitations

This system is an early-warning prototype and has known limitations. Rule-based scoring is explainable but is not a trained predictive model; it may not maximize statistical accuracy and should be refined with real churn outcomes. Ticket-to-customer linkage depends on the presence of a real customer identifier in ticket data; if missing, ticket mapping becomes a demonstration-only signal. Local LLM analysis depends on machine performance and timeouts; low timeout values or slower hardware can produce timeout outputs.

Challenges Encountered

The project faced practical challenges common in real analytics prototypes. Data alignment across multiple tables required careful joining keys and datatype normalization. Some datasets may contain missing dates or inconsistent IDs, which can reduce the completeness of derived signals. Running a local LLM can be slow for larger inputs, so the system required safeguards such as per-ticket timeouts to prevent the dashboard from freezing.

Future Improvements

Future work can make the system more production-ready by strengthening data linkage and improving scoring quality. A reliable customer-to-ticket join using CRM or support-system IDs would make text insights fully actionable. A trained churn prediction model could complement rule scoring once sufficient churn labels exist. Additional time-series views (risk score over time, churn reasons over time, adoption milestones) would improve trend tracking. Background processing and caching would make LLM enrichment faster and smoother for larger ticket batches. Exportable playbooks and integration into email/CRM workflows would further improve operational adoption.