

I have sessionized clickstream data for user interaction on website and mobile app with target variable satisfaction (satisfied , dissatisfied). There is a page sequence list with all page names in the order in each session with information like search_start, chat_start ect. hen I did a summery statistics , it turned out that people who are starting a chat are more dissatisfied than satisfied. Also sequences with more error pages in the page sequence list are more dissatisfied

I also have an activity sequence list, event sequence list, time gap list for page interactions, hit timestamp list, previous hit timestamps etc. How can I find user pain points (friction) from these sequences. I need to find and present specific pain points to the client. Please do your research and advice how can I do this. Consider industry best practices and other use cases.

Here's a structured approach, grounded in industry best practices and real-world use cases, to identify and present specific "pain points" (friction) from your sessionized clickstream sequences:

1. Data Preparation & Label Enrichment

First, ensure each session's sequence data (page events, activity events, time gaps, timestamps, etc.) is joined with the satisfaction label ("satisfied" vs. "dissatisfied"). Derive features such as total session length, number of error pages, count of chat starts, and average time gaps. Enrich each record with:

- **Transition pairs** (e.g. Page A → Page B)
 - **Error-event flags** (e.g. one-hot for each error page)
 - **Temporal metrics** (e.g. long dwell times > threshold)
This labeling creates a foundation where you can directly correlate specific sequence patterns to dissatisfaction (matomo.org).
-

2. Sequential Pattern Mining

Use algorithms like cSPADE or PrefixSpan to mine frequent subsequences that appear disproportionately in dissatisfied sessions. These methods uncover patterns such as "Homepage → Search → ErrorPage → ChatStart" that occur more often in unhappy users. Additionally, build a first-order Markov Chain over page states to compute transition probabilities; low-probability or "backtracking" transitions often signal confusion or friction (rudderstack.com, alooba.com).

3. Event & Time-Gap Analysis

Compute per-step dwell times and identify outliers. For instance, excessively long page dwell or repeated rapid clicks (“rage clicks”) often mark frustration. Tools like Mouseflow quantify “rage clicks” when users repeatedly click on non-interactive elements, signaling a broken UI or unmet expectation (mouseflow.com). Plot distributions of time gaps for satisfied vs. dissatisfied cohorts to highlight pages where delays spike.

4. Funnel & Path Visualization

Construct a conversion funnel (e.g. Landing → Search → Product → Checkout) and measure drop-off rates at each step. High drop-off at “Product → Checkout” in dissatisfied sessions pinpoints a pain point. Visualize full-session paths with a Sankey diagram to reveal “loops” or regressions (e.g. users returning repeatedly from Cart back to Product), visually emphasizing where users get stuck (matomo.org).

5. Clustering & Segmentation

Embed each session’s sequence into feature-vector space (e.g. via n-gram counts or sequence embeddings) and apply clustering (e.g. k-means, hierarchical). Identify clusters dominated by dissatisfied users—inspect their “centroid” sequences to pinpoint common friction patterns (e.g. high error-page clusters, chat-heavy clusters) (rudderstack.com).

6. Supervised Sequence Classification & Explainability

Train a sequence-aware classifier (e.g. LSTM, HMM-based features + XGBoost) to predict dissatisfaction. Use SHAP values to rank which steps or events contribute most to the prediction; e.g. “chat_start” and “error_page” may show high SHAP importance for the dissatisfied class (alooba.com). This quantifies the impact of each event in context.

7. Reporting & Client Presentation

For each identified pain point, present:

- **Pattern summary** (e.g. “80% of dissatisfied sessions include ErrorPage within first three steps”)
- **Visual evidence** (Sankey diagram, time-gap histogram)
- **Quantified impact** (drop-off %, increased session time)
Link each pain point to a clear action (e.g. fix broken link, simplify checkout flow, improve search relevancy).

8. Tooling & Use-Case Examples

- **Matomo** or **Amplitude** for built-in path & funnel analysis (matomo.org)
- **RudderStack** pipelines feeding data into Snowflake for large-scale pattern mining with cSPADE (rudderstack.com)
- **Mouseflow** or **FullStory** to capture rage clicks and heatmaps (mouseflow.com)

Implementation Roadmap

1. **Ingest & Label:** Consolidate sequences + satisfaction flags.
2. **Feature Engineering:** Derive transition, error, time-gap metrics.
3. **Pattern Mining:** Run cSPADE/PrefixSpan and Markov analysis.
4. **Visualization:** Build funnels and Sankey flows.
5. **Modeling:** Train sequence classifier + SHAP explainability.
6. **Reporting:** Compile dashboards and slide deck with visuals and action items.

By following this approach—combining unsupervised pattern discovery, supervised explainable models, and clear visualizations—you'll be able to pinpoint the highest-impact friction points and deliver a data-driven action plan to your client.