SaaS Customer Churn Analytics & Experiment Evaluation

Introduction

This project simulates an end-to-end data science workflow for a SaaS (Software as a Service) company using synthetic but realistic customer data. The dataset reflects customer demographics, subscription renewals, behavioral engagement, satisfaction feedback, and randomized experimentation. The primary goals were to understand churn drivers, evaluate experimental impact, build predictive models, and communicate insights through interactive dashboards. Together, these steps mirror the lifecycle of a full-stack data science project inside a real SaaS organization.

This project includes:

- Synthetic data generation using Python
- A Snowflake-based data warehouse with multi-layered schema design
- SQL-based cleaning, integration, and feature engineering
- Behavioral and satisfaction analytics
- Predictive modeling (Logistic Regression + Random Forest)
- SHAP interpretability analysis
- A/B testing evaluation
- Tableau dashboard development and data storytelling

Methodology

Data Generation & Simulation (Python)

This stage involved creating synthetic datasets that behaved like realistic SaaS usage, enabling full control over the analytical environment. The simulation incorporated variability in user behavior, subscription lifecycles, and satisfaction responses to ensure meaningful downstream insights.

Key actions included:

- Modeling user profiles (age, country, signup date, plan type).
- Generating subscription histories with renewals and cancellations.
- Simulating engagement (event counts, active days, recency).
- Creating satisfaction data (NPS, CSAT, comment characteristics).
- Randomly assigning users to experiment variants.
- Designing churn as a probabilistic outcome driven by recency and low activity

Data Warehouse & Schema Design (Snowflake)

A dedicated Snowflake data warehouse environment was set up to support a structured, scalable analytics workflow. The warehouse followed a layered architecture with separate schemas for raw ingestion, transformation, and analytics. This ensured clear separation between unprocessed CSV data, cleaned/staged data, and final analytic tables used for BI and modeling.

Warehouse and database setup included:

- Creating a compute warehouse for analytics
- Creating a dedicated analytics database
- Defining schemas

Raw Layer

The raw schema mirrors the original CSV input, preserving source structure for auditing and reproducibility. All fields were initially ingested as strings to maintain fidelity.

Raw tables created:

- `USERS_RAW` — demographics and user attributes
- `SUBSCRIPTIONS_RAW` — subscription lifecycle fields
- `EVENTS_RAW` — event-level engagement stream
- `SURVEY_RESPONSES_RAW` — NPS/CSAT outcomes and text
- `EXPERIMENT_ASSIGNMENTS_RAW` — A/B test assignment metadata

These tables served as the single source of truth for downstream transformations.

Transform Layer

The transform schema cleaned and validated raw fields, converted data types, and prepared standardized staging tables for analytics and modeling.

Key transformation operations included:

- Casting string dates into `DATE` or `TIMESTAMP` types
- Normalizing categorical fields such as plan type, device type, and experiment variants
- Engineering intermediate fields like `EVENT_DATE`, `DURATION_DAYS`, `IS_KNOWN_USER`
- Creating typed staging tables:
    - `STG_USERS`
    - `STG_SUBSCRIPTIONS`
    - `STG_EVENTS`
    - `STG_SURVEYS`
    - `STG_EXPERIMENT_ASSIGNMENTS`

These clean staging tables enabled consistent joins across domains.

Analytics / Star Schema Layer

A star-like schema was constructed on top of the cleaned staging tables to support both BI dashboards and feature engineering for ML.

Core tables included:

- Dimension Table:
    - `DIM_USERS` — one row per user with stable attributes
- Fact Tables:
    - `FACT_EVENTS` — cleaned event data
    - `FACT_SUBSCRIPTIONS` — subscription history and monetization
    - `FACT_SURVEYS` — NPS/CSAT summaries
    - `FACT_EXPERIMENT_ASSIGNMENTS` — experiment metadata
- Final Feature Table:
    - `FTR_CHURN` — unified modeling table with:
        - Lifecycle features
        - Engagement recency windows

        - Revenue and subscription features
        - Satisfaction aggregations
        - Experiment variant
        - Churn label

Data Cleaning & Feature Engineering (SQL)

User-Level Aggregation Tables (SQL)

Before building the final `FTR_CHURN` modeling table, several user-level aggregation tables were created to roll up behavior, subscriptions, surveys, experiments, and labels to a single row per user. This made it easier to engineer features consistently and align all signals at the user grain.

Key aggregation tables included:

- `USER_EVENTS_AGG` – Aggregates event behavior per user using a fixed reference date
- `USER_SUBSCRIPTION_AGG` – Summarizes subscription history using only valid, known-user subscriptions
- `USER_SURVEY_AGG` – Rolls up satisfaction metrics
- `USER_EXPERIMENT_AGG` – Consolidates experiment metadata for each user
- `USER_CHURN_LABEL` – Defines churn status as of a reference date

These aggregation tables were then joined together to produce the final **FTR_CHURN** feature table used for both BI dashboards and predictive modeling.

Exploratory Data Analysis (Python + SQL)

EDA was conducted primarily through SQL aggregations, Tableau visual exploration, and SHAP-driven model interpretation. These steps ensured that engineered features behaved as expected and allowed early identification of behavioral patterns tied to churn.

Techniques used included:

- **SQL-based summarization** of engagement, subscription depth, revenue, and satisfaction metrics using USER_EVENTS_AGG, USER_SUBSCRIPTION_AGG, and USER_SURVEY_AGG.

- **Behavioral and lifecycle visualizations in Tableau** showing churn trends across activity levels, recency windows, subscription counts, plan types, and satisfaction segments.
- **Segment-level churn comparisons** across countries, plan types, and NPS categories to identify vulnerable user groups.
- **Model-driven EDA using SHAP**, which provided detailed insight into the relative influence of key variables and confirmed early patterns observed in SQL and Tableau.

A/B Experiment Analysis

A simple randomized experiment was simulated to evaluate whether a treatment condition influenced user engagement or churn. The goal of this component was to demonstrate how experiment data fits into the broader analytics ecosystem.

Key steps performed:

- **Verified balanced assignment** across control and treatment groups to ensure the randomization process behaved as expected.
- **Compared churn rates** between the two variants to observe whether the treatment had any noticeable effect.
- **Reviewed basic engagement differences** (e.g., event counts, active days) between groups.

What was not performed:
Unlike a full experiment evaluation, this project did not conduct formal statistical significance testing, confidence interval estimation, uplift modeling, or revenue-based impact calculations. The intent of the experiment was illustrative rather than inferential.

Dashboard Development (Tableau)

Four dashboards and one summary page were created to communicate insights clearly and sequentially. The dashboards replicate professional SaaS BI patterns and encourage a guided exploration of retention dynamics.

Dashboards included:

- **Introductory Overview Page:** Provides project context, objectives, tech stack, and navigation guidance for stakeholders viewing the workbook.

- **Customer Churn Overview:** KPIs, churn trends, churn by plan.
- **Churn Drivers & Behavioral Insights:** Recency, activity levels, subscription tenure, revenue tiers.
- **Satisfaction & Segment Insights:** NPS, CSAT, promoter/detractor churn, country-level segmentation.
- **Experiment Summary:** Churn, engagement, revenue, and NPS differences by variant.
- **Recommendations Page:** Synthesized insights and clear next steps.

View the full interactive dashboard collection here.

Predictive Modeling (Python)

Two predictive models were developed to estimate churn risk and identify influential features. Logistic regression provided interpretability, while the random forest captured nonlinear behavior.

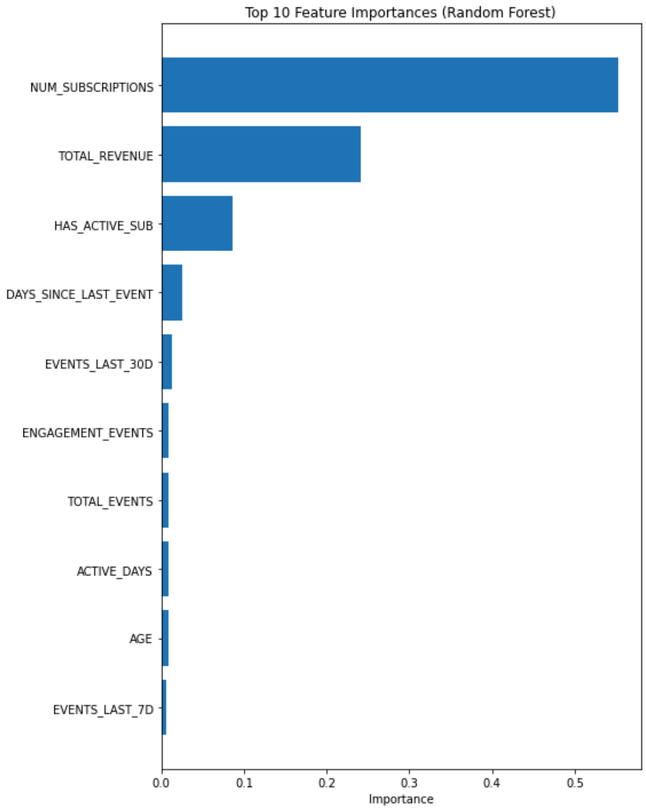
Modeling steps included:

- Train-test split with stratification
- One-hot encoding for categorical fields
- Model pipelines (preprocessing + estimator)
- Evaluation with AUC, precision, recall, F1, and confusion matrices
- Comparison of model performance
- Extraction of feature importance from the random forest

Feature Importance (Random Forest)

A traditional feature importance plot was generated from the random forest model to show the relative contribution of each input variable based on impurity reduction. The top features mirrored the SHAP results, with **Number of Subscriptions**, **Total Revenue**, **Has Active Subscription**, and **Days Since Last Event** ranking highest.

This provided an initial, model-driven confirmation of key churn drivers before applying more granular SHAP analysis.



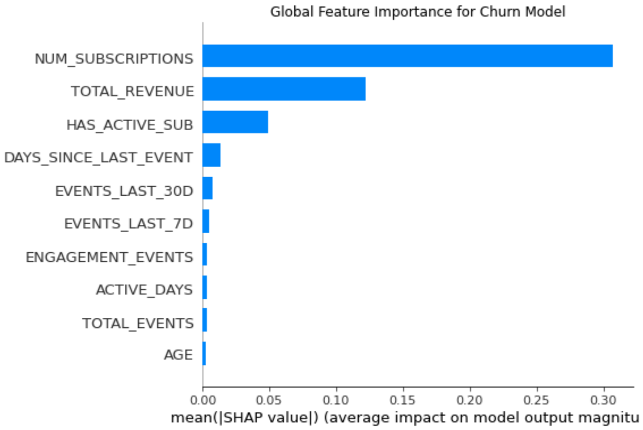
Model Interpretation & Explainability (SHAP)

To better understand the reasoning behind the model’s predictions, SHAP (SHapley Additive exPlanations) was applied to the trained random forest classifier. SHAP decomposes each prediction into the contributions of individual features, enabling both global model interpretation and customer-level insights. The following figures summarize the model’s behavior across the dataset.

SHAP outputs used:

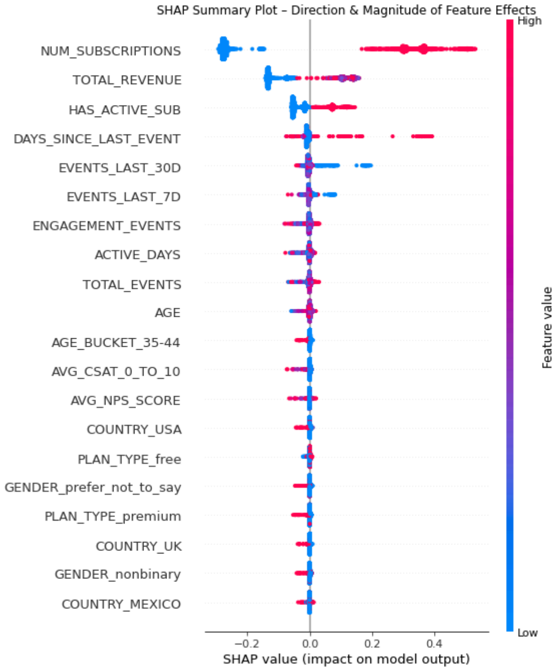
- Bar plot of mean absolute SHAP values (global importance)
- Beeswarm plot showing distribution and directionality
- Dependence plots illustrating interaction effects

SHAP Figure 1 — Global Feature Importance (Bar Plot)



This bar chart displays the mean absolute SHAP value for each feature, representing its overall contribution to the model's churn predictions. The highest bars indicate the variables the model relies on most. In this case, **Number of Subscriptions**, **Total Revenue**, **Has Active Subscription**, and **Days Since Last Event** dominate model influence. These results confirm that subscription lifecycle depth, customer value, and engagement recency are the strongest predictors of churn, aligning with well-established SaaS retention patterns.

SHAP Figure 2 — Beeswarm Summary Plot

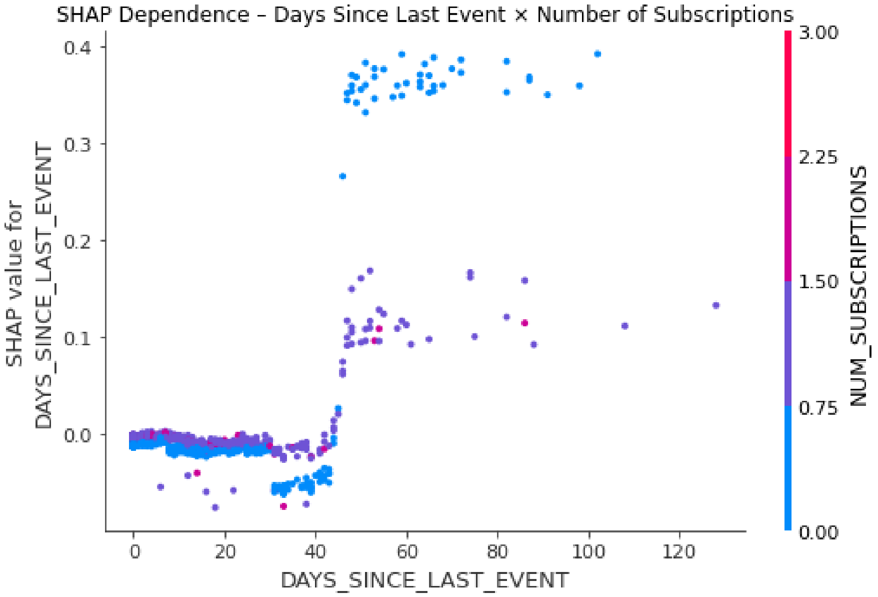


The SHAP beeswarm plot visualizes not only how important each feature is but also **how** it affects predictions. Each point is a customer.

- **Red points (higher feature values)** show how high counts or amounts push the prediction toward churn or retention.
- **Blue points (lower values)** show the opposite.

For example, users with **high subscription counts** strongly influence the model toward predicting churn in this synthetic dataset. Meanwhile, **long inactivity periods** sharply increase churn probability, confirming recency as a key behavioral driver. The spread of points indicates meaningful variance in how different customers are affected by each feature.

SHAP Figure 3 — Dependence Plot (Days Since Last Event × Number of Subscriptions)



This dependence plot highlights the interaction between **Days Since Last Event** and **Number of Subscriptions**. The rising SHAP curve shows that churn risk increases slowly at first but spikes sharply after roughly **40 days of inactivity**.

Color shows subscription count:

- Users with **multiple subscriptions** experience a *stronger* spike, meaning disengaged long-term customers may be especially vulnerable.

This insight supports targeted re-engagement strategies focused on recency thresholds.

Findings & Insights

Behavioral & Lifecycle Insights

Customer lifecycle characteristics emerged as the strongest predictors of churn.

Key findings:

- Users with multiple subscription renewals exhibited far lower churn rates.
- Total revenue strongly correlated with retention, reflecting customer value and tenure.
- Having an active subscription was one of the most stabilizing factors.
- Days since last event was a powerful behavioral signal—the risk spiked sharply after ~40 days of inactivity.
- Low lifetime event activity and fewer active days were characteristic of churned users.

Satisfaction Insights

Satisfaction scores were directionally associated with churn but had weaker predictive influence compared to lifecycle features.

Patterns observed:

- Promoters (NPS 9–10) churned significantly less than detractors.
- CSAT scores aligned with NPS trends but contributed less to prediction strength.

Predictive Modeling Insights

The predictive models consistently identified a set of core churn drivers.

Model results showed:

- Both models achieved ~98% AUC, demonstrating strong predictability.
- **Number of Subscriptions** was the single strongest predictor of churn.
- **Total Revenue** was the second most influential feature.
- **Has Active Subscription** played a major role in predicting churn risk.
- **Days Since Last Event** was the most important engagement-based feature.

These factors align with known SaaS churn dynamics, where lifecycle maturity and customer value eclipse short-term satisfaction signals.

SHAP Interpretability Insights

SHAP analysis confirmed and deepened understanding of model behavior.

Global interpretations:

- Subscription-related features dominated the model's decision structure.
- High subscription counts strongly influenced churn predictions.
- Long inactivity periods sharply pushed SHAP values toward churn.
- Active subscription status was a major churn-reducing factor.

Interaction insights:

- Users with longer subscription histories were *especially* vulnerable when they became inactive.
- Revenue and subscription depth interacted to reflect customer value.

Experiment Insights

The A/B test showed minimal evidence of treatment impact.

Experiment outcomes:

- Treatment and control churn rates were nearly identical.
- Engagement metrics showed only a minor uplift in the treatment group.

This test does not justify rollout and would require redesign for meaningful insights.

Recommendations

The modeling results highlight that subscription lifecycle depth and customer value are the strongest protections against churn, while engagement recency remains an important behavioral signal. These findings suggest several strategic opportunities to enhance customer retention by targeting high-risk users earlier and reinforcing renewal behaviors.

Recommended actions:

- **Increase subscription renewal rates:** Create targeted renewal messages and incentives for customers with only one or two past subscription cycles, as early-tenure users are most vulnerable.
- **Prioritize high-value users for premium retention actions:** Use **total revenue** to segment VIP or long-tenured users for proactive support or loyalty campaigns.
- **Intervene before subscription expiration:** Implement automated reminders, in-app nudges, or special offers for customers approaching subscription end dates.
- **Address lapsing engagement proactively:** Deploy outreach workflows tied to **days since last event**, especially for users with lower revenue or subscription histories.
- **Use churn modeling scores operationally:** Integrate predicted churn probabilities into CRM tools so customer success teams can take action on users at the highest risk.
- **Design experiments that target subscription depth:** For example, test onboarding enhancements or renewal incentives specifically for first-time or early-tenure subscribers.

Next Steps

Building on the insights that subscription history and revenue are the strongest churn signals, future work should focus on deepening lifecycle analytics and operationalizing predictive scoring. This will help reduce churn through targeted interventions aligned with the drivers identified in the model.

Future enhancements include:

- Developing a **subscription lifecycle modeling framework** to predict renewal probability and optimize retention strategies for early-tenure users.
- Enhancing churn prediction with **advanced models** (XGBoost, LightGBM) to refine the influence of revenue and subscription depth.
- Creating **automated workflows** that trigger when a user nears subscription expiration or shows decreasing revenue potential.
- Building **long-term customer value (LTV) models** to pair with churn predictions for better prioritization of retention efforts.
- Conducting segmentation to understand how subscription patterns vary across demographics, plan types, or engagement clusters.
- Creating a **real-time churn monitoring dashboard** for customer success teams using Snowflake + Tableau or Power BI.