UNIVERSITY OF PIRAEUS

SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

DEPARTMENT OF DIGITAL SYSTEMS

POSTGRADUATE PROGRAM

“INFORMATION SYSTEMS & SERVICES”

Behavioral AI for Enterprise Customer Lifecycle Management: From Transactional Systems to Explainable Predictive Decision Support

by

Evgenia Bairachtari

MSc Programme “Information Systems & Services”

UNIVERSITY OF PIRAEUS

Athens, May 2026

Validation Page

Ονοματεπώνυμο: Μπαϊραχτάρη Ευγενία

Τίτλος Μεταπτυχιακής Διπλωματικής Εργασίας: Behavioral AI for Enterprise Customer Lifecycle Management: From Transactional Systems to Explainable Predictive Decision Support

Η παρούσα Μεταπτυχιακή Διπλωματική Εργασία υποβάλλεται ως μερική εκπλήρωση των απαιτήσεων του Προγράμματος Μεταπτυχιακών Σπουδών "Πληροφοριακά Συστήματα & Υπηρεσίες" του Τμήματος Ψηφιακών Συστημάτων του Πανεπιστημίου Πειραιώς και εγκρίθηκε στις 21/05/2026 από τα μέλη της Εξεταστικής Επιτροπής.

Εξεταστική Επιτροπή

Επιβλέπων: Μιχαήλ Φιλιππάκης, Καθηγητής

Μέλος Εξεταστικής Επιτροπής: Μαρία Χαλκίδη, Καθηγήτρια

Μέλος Εξεταστικής Επιτροπής: Δημοσθένης Κυριαζής, Καθηγητής

ΥΠΕΥΘΥΝΗ ΔΗΛΩΣΗ ΑΥΘΕΝΤΙΚΟΤΗΤΑΣ

Εγώ, η Ευγενία Μπαϊραχτάρη, γνωρίζοντας τις συνέπειες της λογοκλοπής, δηλώνω υπεύθυνα ότι η παρούσα εργασία με τίτλο «Behavioral AI for Enterprise Customer Lifecycle Management: From Transactional Systems to Explainable Predictive Decision Support», αποτελεί προϊόν αυστηρά προσωπικής εργασίας και όλες οι πηγές που έχω χρησιμοποιήσει, έχουν δηλωθεί κατάλληλα στις βιβλιογραφικές παραπομπές και αναφορές. Τα σημεία όπου έχω χρησιμοποιήσει ιδέες, κείμενο ή/και πηγές άλλων συγγραφέων, αναφέρονται ευδιάκριτα στο κείμενο με την κατάλληλη παραπομπή και η σχετική αναφορά περιλαμβάνεται στο τμήμα των βιβλιογραφικών αναφορών με πλήρη περιγραφή.

Επιπλέον δηλώνω υπεύθυνα ότι η συγκεκριμένη Μεταπτυχιακή Διπλωματική Εργασία έχει συγγραφεί από εμένα προσωπικά και δεν έχει υποβληθεί ούτε έχει αξιολογηθεί στο πλαίσιο κάποιου άλλου μεταπτυχιακού ή προπτυχιακού τίτλου σπουδών, στην Ελλάδα ή στο εξωτερικό.

Παράβαση της ανωτέρω ακαδημαϊκής μου ευθύνης αποτελεί ουσιώδη λόγο για την ανάκληση του πτυχίου μου. Σε κάθε περίπτωση, αναληθούς ή ανακριβούς δηλώσεως, υπόκειμαι στις συνέπειες που προβλέπονται από τις διατάξεις της Ελληνικής και Κοινοτικής Νομοθεσίας περί πνευματικής ιδιοκτησίας.

Η ΔΗΛΟΥΣΑ

Ονοματεπώνυμο: Μπαϊραχτάρη Ευγενία

Αριθμός Μητρώου: me2468

Υπογραφή:

Acknowledgments

Completing this thesis gave me the opportunity not only to deepen my academic and technical knowledge, but also to reflect on the professional I continue to become. Throughout this journey, I learned to work across different environments, disciplines, and responsibilities — from enterprise systems and operational challenges to analytics and AI-driven technologies. More importantly, I realized how essential adaptability, persistence, and critical thinking are when working across both academic and enterprise environments.

This MSc experience allowed me to explore Information Systems beyond theory, connecting academic thinking with real-world application. It strengthened not only my technical understanding, but also the way I approach problems, think more critically, and balance different responsibilities. Completing this program while working in a full-time leadership role made the experience demanding, but also more meaningful. It reinforced my belief that technology creates value only when it helps people and organizations make clearer, better, and more informed decisions.

I would like to sincerely thank my supervisor, Professor Filippakis, for his guidance, trust, and support throughout this period. From the beginning, he encouraged independent thinking and practical reasoning. I am also grateful to all professors of the MSc program for fostering a learning environment closely aligned with real-world industry challenges.

I would also like to thank Redcare Pharmacy for the trust and opportunities that allowed me to develop this research within the context of real enterprise challenges. Working on this thesis alongside real organizational responsibilities shaped not only the direction of the work, but also its practical relevance and perspective. I am also grateful to my team and colleagues for their encouragement, interest, and support, which made this demanding period feel far less solitary.

A special acknowledgment goes to Mr. Politis at Logistic-i, who gave me my first opportunity in the world of ERP systems, Information Technology, and data. His guidance, trust, and continuous support played a defining role in shaping both my professional path and my decision to pursue this MSc. Those early years working alongside him challenged me, taught me resilience, and deeply influenced the way I approach responsibility, problem-solving, and professional growth today.

On a personal level, I would like to thank my family, and the people close to me for their patience, encouragement, and support even across the distance.

Finally, I am especially grateful to Karmo, whose quiet companionship brought balance during long nights of studying, writing, and reflection — whether through midnight walks, moments of calm in the park, or simply resting beside me while I worked.

Abstract in English

In enterprise environments, customer churn extends beyond a purely predictive problem, representing a behavioral and decision-making challenge that directly affects customer lifetime value, operational efficiency, and long-term growth. This complexity becomes particularly pronounced in digital healthcare and online pharmacy ecosystems, where customer interaction patterns differ across Prescription (RX), Over-the-Counter (OTC), and FREE product interactions.

This thesis develops a behavioral AI framework for customer churn interpretation, prediction, and decision support within a real-world online pharmacy environment. Rather than relying on static inactivity thresholds or purely model-driven approaches, the study begins with reconstructing customer behavior from transactional systems and progressively builds toward segmentation, personalized churn interpretation, predictive modeling, and operational decision logic.

The analysis is based on a large-scale transactional dataset covering 122.8 million order lines, 45.2 million completed orders, and 10.9 million customers over a 24-month period. Customer behavior is modeled at the ordering-customer level using leakage-safe behavioral signals derived exclusively from observable transaction patterns. Particular emphasis is placed on return dynamics, purchasing consistency, recency-frequency behavior, and structural differences across customer segments.

The findings demonstrate that inactivity cannot be interpreted uniformly across the population. Distinct behavioral rhythms exist across segments and individual interaction patterns making static churn definitions unreliable. To address this, the framework introduces segment-specific and personalized churn definitions based on empirical return behavior and customer-level expected interaction patterns.

Building on this foundation, the framework is extended into predictive Machine Learning (ML) models that forecast future customer states (ACTIVE, AT RISK, CHURNED) using behaviorally grounded features. Beyond predictive performance, the work emphasizes explainability, scalability, and operational usability.

The thesis concludes with the design of a decision-support framework that connects behavioral analytics, predictive intelligence, and operational prioritization. The proposed approach demonstrates how transactional systems, behavioral modeling, and ML can be integrated into a coherent, explainable, and scalable architecture for enterprise customer lifecycle management.

Abstract in Greek

Η αποχώρηση πελατών (customer churn) δεν αποτελεί απλώς ένα πρόβλημα πρόβλεψης, αλλά μια σύνθετη επιχειρησιακή πρόκληση που συνδέεται άμεσα με τη συμπεριφορά των πελατών και τη λήψη αποφάσεων. Επηρεάζει την αξία πελάτη, την αποδοτικότητα των ενεργειών και τη μακροπρόθεσμη ανάπτυξη των οργανισμών. Στο περιβάλλον των online φαρμακείων, η ερμηνεία της καθίσταται ακόμη πιο απαιτητική, καθώς τα πρότυπα αλληλεπίδρασης διαφέρουν ουσιαστικά μεταξύ προϊόντων Prescription (RX), Over-the-Counter (OTC) και προϊόντων FREE.

Η παρούσα εργασία αναπτύσσει ένα πλαίσιο behavioral AI για την κατανόηση, ερμηνεία και πρόβλεψη της αποχώρησης πελατών, καθώς και για τη σύνδεσή της με την επιχειρησιακή λήψη αποφάσεων σε ένα πραγματικό περιβάλλον online φαρμακείου. Αντί να βασίζεται σε στατικούς ορισμούς αδράνειας ή αποκλειστικά σε προγνωστικά μοντέλα, η προσέγγιση ξεκινά από την ανάλυση της πραγματικής συμπεριφοράς των πελατών, όπως αυτή αποτυπώνεται σε συναλλακτικά δεδομένα, και εξελίσσεται σταδιακά σε τμηματοποίηση, εξατομικευμένη ερμηνεία churn, προγνωστική μοντελοποίηση και εφαρμογή σε επίπεδο υποστήριξης λήψης αποφάσεων.

Η ανάλυση βασίζεται σε μεγάλο σύνολο συναλλακτικών δεδομένων, το οποίο περιλαμβάνει 122,8 εκατομμύρια γραμμές παραγγελίας, 45,2 εκατομμύρια ολοκληρωμένες παραγγελίες και 10,9 εκατομμύρια πελάτες σε χρονικό ορίζοντα 24 μηνών. Η μελέτη πραγματοποιείται στο επίπεδο του πελάτη που πραγματοποιεί την παραγγελία και χρησιμοποιεί συμπεριφορικά σήματα που προκύπτουν αποκλειστικά από την παρατηρήσιμη αγοραστική δραστηριότητα, διασφαλίζοντας την αποφυγή πληροφοριακής διαρροής. Ιδιαίτερη έμφαση δίνεται στη δυναμική επιστροφών, στη συνέπεια της αγοραστικής συμπεριφοράς, στους δείκτες recency – frequency και στις διαφορές που εμφανίζονται μεταξύ των ομάδων πελατών.

Τα αποτελέσματα δείχνουν ότι η αδράνεια δεν μπορεί να ερμηνευθεί με ενιαίο τρόπο για όλους τους πελάτες. Διαφορετικά πρότυπα συμπεριφοράς και ρυθμοί επαναγοράς οδηγούν σε διαφορετική σημασία της ίδιας χρονικής περιόδου αδράνειας. Με βάση αυτό, το προτεινόμενο πλαίσιο εισάγει ορισμούς churn ανά ομάδα πελατών, καθώς και εξατομικευμένους ορισμούς, οι οποίοι βασίζονται στην πραγματική συμπεριφορά επιστροφής και στα αναμενόμενα πρότυπα κάθε πελάτη.

Στη συνέχεια, η προσέγγιση επεκτείνεται σε προγνωστικά μοντέλα Machine Learning (ML), τα οποία προβλέπουν την εξέλιξη της κατάστασης των πελατών (ACTIVE, AT RISK, CHURNED) με βάση χαρακτηριστικά που είναι άμεσα συνδεδεμένα με τη συμπεριφορά τους. Πέρα από την προγνωστική ακρίβεια, η εργασία δίνει ιδιαίτερη έμφαση στην ερμηνευσιμότητα των αποτελεσμάτων, στη δυνατότητα επιχειρησιακής κλιμάκωσης και στην πρακτική εφαρμογή τους σε πραγματικά επιχειρησιακά περιβάλλοντα.

Η βασική συνεισφορά της εργασίας είναι ο σχεδιασμός ενός ενιαίου πλαισίου υποστήριξης λήψης αποφάσεων, το οποίο συνδέει την ανάλυση συμπεριφοράς, την προγνωστική μοντελοποίηση και την επιχειρησιακή προτεραιοποίηση ενεργειών. Η προσέγγιση αναδεικνύει πώς τα συναλλακτικά συστήματα, η μοντελοποίηση συμπεριφοράς και το ML μπορούν να συνδεθούν σε μία συνεκτική, επεξηγήσιμη και κλιμακούμενη αρχιτεκτονική για τη διαχείριση του κύκλου ζωής των πελατών σε σύγχρονα enterprise περιβάλλοντα.

Table of Contents

List of Figures

List of Tables

List of Acronyms

ACRONYM / TERM	DEFINITION
AI	Artificial Intelligence
BOM	Bill of Materials
CDM	Customer Data Model
CLV	Customer Lifetime Value
CRM	Customer Relationship Management
CV	Coefficient of Variation
DB	Database
DBFS	Databricks File System
DBR	Databricks Runtime
DELTA	Delta Lake
ERP	Enterprise Resource Planning
FREE	Free product interaction
KPI	Key Performance Indicator
LR	Logistic Regression
LTS	Long-Term Support
ML	Machine Learning
MLLIB	Machine Learning Library
OTC	Over-the-Counter
P80	80th Percentile
P90	90th Percentile
RF	Random Forest
RX	Medical Prescription
SPARK ML	Apache Spark Machine Learning
SQL	Structured Query Language
XGBOOST	Extreme Gradient Boosting

1. Introduction

1.1 Purpose of the Study

Customer retention has evolved from a marketing concern into a strategic challenge that directly affects revenue stability, operational efficiency, and long-term growth. In large-scale digital commerce environments, particularly in digital healthcare, understanding customer churn is no longer only about identifying inactivity, but about interpreting complex behavioral patterns and translating them into timely and effective business decisions (Reichheld & Sasser, 1990).

Traditional approaches often treat churn as a static outcome or a purely predictive modeling problem. However, in practice, churn emerges from dynamic customer behavior that varies across purchasing contexts, product interaction patterns, and individual engagement rhythms. As a result, organizations increasingly require not only predictive models, but structured decision-support frameworks capable of connecting transactional data, behavioral interpretation, and operational action.

Within this context, the purpose of this thesis is to develop a structured and scalable approach for understanding, defining, and predicting customer churn in a large-scale e-commerce pharmacy environment. Instead of treating churn purely as a predictive task, the study establishes a governed behavioral foundation that reconstructs customer interaction patterns over time. Particular emphasis is placed on purchasing dynamics, return behavior, behavioral consistency, and structural differences across customer groups in order to support more precise and operationally meaningful churn interpretation.

Building on this behavioral foundation, the thesis moves from customer understanding to segment-level and personalized churn definitions, before extending the framework into predictive Machine Learning (ML) and enterprise decision-support logic. The goal is to connect transactional data, customer behavior insights, predictive intelligence, and operational actions in a way that is clear, scalable, and meaningful for digital healthcare e-commerce.

1.2 Problem Statement

Customer churn represents one of the most critical challenges in e-commerce, as it directly affects revenue stability, marketing efficiency, customer lifetime value, and long-term business sustainability. However, in the context of an online pharmacy, churn becomes more complex than in traditional retail environments due to the coexistence of different customer journeys, purchasing motivations, and engagement structures (Gupta & Lehmann, 2003).

Customers do not interact with online pharmacy platforms through a single behavioral pattern. Some exhibit occasional and low-commitment purchasing behavior driven primarily by Over-the-Counter (OTC) or FREE products, while others follow more structured and recurring interaction patterns associated with Prescription (RX) medication needs. As a result, customer activity cannot be interpreted uniformly across the entire population.

These structural differences create substantial variability in ordering frequency, return probability, purchasing consistency, and lifecycle duration. Consequently, applying a single static churn definition across all customers leads to systematic misclassification and reduces the effectiveness of downstream retention strategies. Customers with different purchasing rhythms are evaluated under identical inactivity thresholds, masking important behavioral signals and limiting both analytical reliability and business impact.

At the same time, leading digital commerce organizations increasingly rely on personalized and data-driven retention strategies. Companies such as DocMorris, dm-drogerie markt, and Amazon demonstrate that sustainable retention performance is driven not only by predictive capabilities, but also by customer segmentation, ecosystem integration, and proactive engagement mechanisms embedded within the customer experience.

This creates a clear need for an analytical approach capable of capturing structural differences between customer groups, defining churn in a context-aware manner, supporting personalized customer-state interpretation, and translating analytical insights into actionable retention strategies. Addressing this challenge requires moving beyond purely model-centric approaches and establishing a governed analytical foundation that connects customer behavior, business logic, and predictive analytics within an explainable business framework.

Beyond churn prevention, the same behavioral perspective also creates a foundation for understanding how customers evolve across product categories, particularly in the transition from OTC-driven interaction toward RX-related behavior.

1.3 Industry Context and Competitive Landscape

Customer retention in e-commerce has evolved from a largely reactive process into a structured, data-driven discipline. Leading organizations no longer rely solely on post-hoc churn analysis, but increasingly design systems capable of proactively identifying disengagement patterns and triggering targeted interventions before customer relationships deteriorate.

This transformation is particularly evident in the online pharmacy and broader digital retail ecosystem, where retention performance is increasingly shaped by personalization, ecosystem integration, and customer interaction design rather than isolated promotional activities. Companies such as DocMorris, dm-drogerie markt, and Amazon illustrate different but complementary approaches to long-term customer engagement.

DocMorris reinforces customer retention through recurring healthcare-oriented interactions, including prescription follow-ups, digital health services, and engagement processes linked to ongoing treatment needs. This highlights the importance of embedding retention mechanisms directly into the healthcare customer journey (DocMorris AG, 2025).

In contrast, dm-drogerie markt focuses on ecosystem-based loyalty and cross-channel engagement through partnerships such as PAYBACK. By combining loyalty programs, personalized offers, and customer segmentation mechanisms, the company creates an integrated experience that encourages repeated interaction and long-term engagement (dm-drogerie markt, 2025).

Amazon provides a well-known example of ecosystem-driven retention, where customer interaction is deeply embedded into the platform experience through subscription models, frictionless logistics, personalized recommendations, and large-scale predictive analytics (Amazon, 2025).

Despite their differences, these organizations share a common principle: effective retention management depends on understanding how customers interact with the platform over time and how engagement patterns evolve across different customer groups. These practices reinforce the limitations of simplistic churn definitions and highlight the need for analytical approaches that connect behavioral understanding, predictive modeling, and operational activation.

1.4 Business Context: Redcare Pharmacy

Redcare Pharmacy N.V. is one of the leading online pharmacy platforms in Europe, operating across multiple markets including Germany, Austria, France, Belgium, Italy, the Netherlands, and Switzerland. As a digital-first healthcare organization, the company combines large-scale e-commerce operations with pharmaceutical services, offering a broad portfolio that includes Prescription (RX) medications, Over-the-Counter (OTC) products, and general health and wellness items (RedCare Pharmacy, 2025).

Unlike traditional e-commerce environments, online pharmacy ecosystems operate within a highly regulated and structurally complex setting. Customer activity is influenced not only by conventional purchasing dynamics but also by healthcare-specific constraints such as prescription requirements, reimbursement structures, recurring treatment needs, and regulatory obligations.

As a result, customer interactions differ across product categories. RX-related purchasing behavior tends to follow more structured and recurring patterns, while OTC and FREE interactions are typically more discretionary and behaviorally variable. This creates significant heterogeneity across the customer population and makes generalized churn interpretation particularly challenging.

At the same time, Redcare's business model relies heavily on sustained customer engagement and recurring purchasing activity. Customer retention is therefore not only a marketing objective, but a core driver of operational efficiency, customer lifetime value, and long-term growth.

Within this context, the ability to consistently understand, define, and predict customer churn becomes strategically important. This requires an analytical approach capable of moving beyond surface-level inactivity metrics and capturing the structural interaction patterns that differentiate customer groups and shape long-term engagement behavior.

This combination of large-scale digital commerce dynamics and healthcare-specific behavioral constraints makes Redcare Pharmacy a highly relevant environment for studying customer churn in a real-world operational environment.

1.5 Literature Review and Research Positioning

Customer churn has been extensively studied across multiple industries, particularly in telecommunications, banking, subscription services, and digital commerce environments. Existing literature consistently demonstrates a strong relationship between customer retention, profitability, and long-term business sustainability (Reichheld & Sasser, 1990). As a result, churn prediction has become one of the most widely applied use cases of data-driven customer analytics and ML.

In non-contractual environments, where customers are not bound by explicit subscription agreements, churn is typically inferred from inactivity rather than directly observed. Foundational research highlights the importance of modeling return behavior, recency – frequency dynamics, and probabilistic customer activity in order to estimate disengagement (Fader, Hardie, & Lee, 2005). However, in practice, many implementations still rely on fixed inactivity thresholds, which often oversimplify customer behavior and fail to account for structural differences across customer groups.

A large body of research focuses on predictive modeling techniques, comparing algorithms such as Logistic Regression, Random Forest, and Gradient Boosting in terms of classification accuracy (Verbeke, Dejaeger, Martens, Hur, & Baesens, 2012). While these approaches demonstrate strong predictive performance, they typically treat churn as a classification problem, placing less emphasis on how churn is defined and interpreted from a behavioral perspective.

At the same time, more recent literature emphasizes the importance of interpretability and explainability in ML, particularly in business-critical applications (Doshi-Velez & Kim, 2017). Although these contributions improve transparency at the model level, they often do not address the interpretability of the churn definition itself or its alignment with real customer behavior.

From a business perspective, the value of churn prediction extends beyond predictive accuracy. Predictive systems must integrate with operational processes, support decision-making, and remain aligned with governance and scalability requirements. However, many existing approaches do not sufficiently address how behavioral interpretation, predictive modeling, and business action connect within real organizational environments.

Taken together, these streams of research reveal a consistent gap: churn is often modeled accurately, but not always defined meaningfully or operationalized effectively. The disconnection between behavioral understanding, predictive modeling, and enterprise decision-making limits the practical value of many churn analytics approaches.

This thesis addresses this gap by developing a multi-layered behavioral framework that integrates governed data preparation, behavioral analysis, segmentation, personalized churn interpretation, leakage-safe predictive modeling, and business decision-support logic within a unified analytical architecture.

Rather than approaching churn exclusively as a predictive problem, the proposed approach positions churn as a behaviorally grounded signal that must be interpreted in context and translated into actionable decisions. In doing so, the work contributes a more realistic and operationally applicable approach to customer churn management in large-scale digital healthcare environments (Hadden, Tiwari, Roy, & Ruta, 2007).

This positioning does not aim to replace established churn prediction methods, but to strengthen their practical relevance by improving the behavioral validity of the churn labels and decision logic on which they depend.

1.6 Research Methodology and Thesis Structure

The thesis follows a layered methodological structure, moving from behavioral data preparation to churn interpretation, predictive modeling, and decision-support design.

Chapter 2 establishes the behavioral data foundation and defines the transactional dataset used throughout the analysis, with emphasis on governance and leakage-safe design.

Chapter 3 develops a behavioral understanding of customer activity through exploratory analysis of purchasing patterns and return dynamics.

Chapter 4 introduces the behavioral segmentation framework used to group customers into structurally meaningful segments.

Chapter 5 defines segment-level churn based on empirical behavioral differences.

Chapter 6 extends churn interpretation to the customer level through personalized behavioral baselines.

Chapter 7 evaluates the impact of personalization on churn interpretation and business decision-making.

Chapter 8 introduces predictive ML models for forecasting future customer states.

Chapter 9 translates predictive outputs into enterprise decision-support frameworks.

Chapter 10 summarizes the findings, limitations, and future directions.

2. Behavioral Data Foundation and Modeling Scope

A consistent definition of churn is a prerequisite for any data-driven customer retention strategy. However, churn is not an inherent property of transactional data. Instead, it emerges from how customer activity is scoped, filtered, reconstructed, and interpreted within a controlled analytical environment (Reichheld & Sasser, 1990).

This chapter establishes the transactional foundation used throughout the thesis by defining a clean, consistent, and intentionally scoped analytical dataset. The focus is not just on preparing data for modeling, but on making sure that segmentation, churn analysis, and predictive insights are built on consistent and understandable customer behavior patterns.

Particular emphasis is placed on entity consistency, transaction validity, and information leakage prevention. As a result, the dataset design prioritizes reproducibility, interpretability, and analytical robustness over unnecessary feature complexity or purely model-driven optimization.

2.1 Analytical Scope and Time Horizon

This analysis aims to define churn as a meaningful behavioral signal that can effectively support customer retention and reactivation efforts. Rather than treating churn as a purely analytical concept, the framework aims to identify inactivity patterns that can be translated into actionable interventions such as CRM campaigns, personalized recommendations, and engagement actions (Kumar & Reinartz, 2018).

Customer purchasing behavior is analyzed over the 24 months preceding the reference date of December 31, 2025, ensuring consistency and comparability across all subsequent analyses. This observation window was selected to provide sufficient behavioral depth while maintaining temporal relevance for churn interpretation and predictive modeling.

To maintain a clean and homogeneous analytical scope, the dataset is restricted to transactions where *sales_domain = 'os'* and *sub_company_id = 'Shop-COM'*. This restriction minimizes behavioral distortion caused by cross-market differences in regulation, logistics, operational processes, and organizational structures. By controlling the analytical environment at the market and operational level, the resulting churn signals remain behaviorally consistent and comparable throughout the analysis.

2.2 Churn Modeling Scope and Entity Definition

All analyses in this thesis are performed at the ordering customer level (*super_customer_id_order*), which represents the commercial entity interacting with the platform. Importantly, the dataset includes both online and assisted orders, allowing customer behavior to be analyzed across channels. This approach ensures that churn is defined based on the full observed purchasing behavior of the customer rather than being artificially biased by channel-specific interaction patterns.

For this reason, *super_customer_id_order* is used as the primary customer identifier, enabling stable behavioral tracking across time and interaction channels. This is particularly important in operational environments where customer activity may occur through multiple operational pathways while still representing the same underlying commercial relationship.

While patient-level behavior and medication adherence are clinically relevant in a pharmacy context, they are intentionally out of scope for this study, which focuses on commercial churn and retention actions. In many real-world scenarios, the ordering customer and the patient are not the same individual, such as family members placing orders on behalf of others. As a result, patient-level churn would be considerably less usable from a business and product perspective.

Although a customer dimension table was available, it was intentionally excluded from the modeling scope. The table is primarily patient-centric and consists largely of pre-derived business categorizations rather than raw behavioral signals. Incorporating such attributes would introduce entity inconsistencies, weaken behavioral transparency, and increase the risk of information leakage within the predictive framework (Kaufman, Rosset, Perlich, & Stitelman, 2012).

To ensure conceptual consistency and practical applicability, churn is therefore modeled exclusively at the ordering customer level and derived solely from observable transactional behavior reconstructed from completed purchase history.

Since churn is interpreted through recurring purchase behavior over time, establishing a consistent definition of what constitutes a “true order” becomes essential for all downstream behavioral metrics and churn-state calculations (Fader, Hardie, & Lee, 2005).

2.3 Order Integrity

For churn analysis, orders are modeled at the customer-order level using *main_order_id_bi*, which uniquely represents a completed customer purchase across sales domains and source systems. This prevents the artificial inflation of order counts caused by operational order splits at the ERP or sub-order level and ensures that behavioral metrics reflect actual customer purchasing activity rather than system-generated technical fragmentation.

To ensure that churn signals are derived exclusively from genuine purchasing behavior, the dataset is restricted to successfully processed transactions only. Order lines with statuses such as Cancelled or Returned are excluded because they do not represent completed purchases and would introduce noise into recency- and frequency-based behavioral signals. The condition *order_line_status = 'Processed'* is therefore used as the operational indicator of completed customer transactions throughout the thesis.

For the purpose of behavioral reconstruction, Processed order lines are treated as completed commercial interactions. This definition does not attempt to fully capture downstream financial outcomes, returns, refunds, or fulfillment exceptions. Instead, it establishes a stable transactional state used consistently for behavioral churn analysis.

This distinction is important because churn-related metrics such as recency, purchase frequency, and return gaps must reflect customer purchasing behavior rather than internal system artifacts or operational exceptions.

2.4 Transaction Value Definition and Interpretation

Customer value metrics are computed using the field *sales_price_net_rep_cur*, which represents the net reported transaction value in reporting currency after internal adjustments such as rebates or settlements (RedCare Pharmacy, 2025).

While internal documentation describes this field as the final price paid by the customer, it does not strictly represent out-of-pocket expenditure in all scenarios, particularly for RX transactions involving reimbursement mechanisms. Nevertheless, the metric provides a stable and comparable representation of transactional economic activity and is therefore considered sufficiently robust for behavioral churn analysis.

It is important to note that this variable reflects transaction value rather than customer cash expenditure or profit contribution. As a result, it is suitable for analyzing behavioral patterns such as engagement, order value, purchasing activity, and customer interaction intensity, but it should not be interpreted as a profitability or margin indicator.

This distinction is particularly important when interpreting RX-related behavior, where high transaction values may not directly correspond to high customer spending due to reimbursement structures and regulated pharmaceutical pricing mechanisms.

Order lines containing missing transaction values were excluded to ensure consistency across monetary metrics used throughout the analysis. Validation confirmed that such cases represent only a negligible share of total records and therefore have minimal impact on the analytical population.

Accordingly, *sales_price_net_rep_cur* is interpreted throughout the thesis as a standardized transaction-value proxy rather than a literal representation of patient out-of-pocket expenditure.

2.5 Product Scope Definition

The analytical scope focuses exclusively on core commercial product types that reflect observable purchasing activity. Specifically, the dataset is restricted to the product subtypes FREE, OTC, and RX, which represent the primary behavioral categories relevant for churn analysis and customer segmentation.

Operational or non-commercial transaction lines, such as vouchers, flyers, shipping-related entries, and records without relevant product subtype information, are excluded indirectly through this restriction. By limiting the analytical scope to these core commercial product groups, the dataset remains behaviorally interpretable while preserving alignment with observable customer purchasing behavior and business relevance.

This controlled product scope is particularly important because different product categories are expected to exhibit different ordering patterns, engagement structures, purchase rhythms, and churn dynamics throughout the customer lifecycle.

2.6 Structural Data Exclusions

Order lines representing bill-of-material components (*bom_type_name = 'Stücklistenposition'*) are excluded because they reflect internal system-level product structuring rather than customer-visible purchasing decisions.

Including such records would artificially inflate product counts, distort order-level behavioral metrics, and introduce misleading interpretations of customer activity. Excluding these records therefore ensures that all derived behavioral signals reflect actual customer purchasing behavior rather than internal ERP structuring logic.

2.7 Excluded Features and Modeling Boundaries

To ensure that churn is defined exclusively through observable customer behavior, several feature groups are intentionally excluded from the modeling dataset.

Channel and device attributes are excluded from predictive modeling and instead applied downstream during campaign activation, where they are operationally relevant for communication and intervention delivery.

Product-level identifiers such as *item_id* and *set_id* are excluded because churn interpretation is driven primarily by purchasing behavior rather than individual product preferences. Product information is therefore abstracted into high-level behavioral categories through *item_sub_type_name*.

Prescription document types are excluded from the core churn framework because churn interpretation is independent of reimbursement or insurance mechanisms. RX behavior is instead represented through high-level product-category indicators.

Pricing, promotion, and fulfillment-related attributes, including vouchers, marketplace indicators, and company-driven commercial mechanics, are excluded because they primarily reflect operational business processes rather than intrinsic customer behavior. Similarly, pre-derived business flags such as first-order indicators and RX/OTC classifications are intentionally omitted to preserve behavioral transparency and prevent information leakage within the predictive framework (Kaufman, Rosset, Perlich, & Stitelman, 2012).

2.8 Final Transactional Feature Set for Churn Modeling

The final feature set is intentionally minimal and behavior-driven. Rather than maximizing feature coverage, the framework prioritizes variables that directly reflect observable customer actions and can be interpreted consistently across customer groups.

This design supports robust behavioral analysis while limiting unnecessary complexity and reducing the risk of information leakage. The resulting transactional foundation supports segmentation, churn interpretation, and predictive modeling within a consistent analytical framework.

Table 2.1: Transactional Data Scope and Feature Selection Criteria

Field	Explanation	Condition
order_date	Date the order was placed by the customer.	last 24 months from Dec 31, 2025
sales_domain	Indicates the domain of the sales transaction (Own Stock, Now, Marketplace).	= '*os*'
sub_company_id	Normalized sub-company identifier (e.g., Shop-COM, Shop-BE).	= '*Shop-COM*'
main_order_id_bi	Order ID that identifies a single customer order across all sales domains.	
super_customer_id_order	ID of the ordering customer.	
order_line_status	Indicates whether the order line corresponds to a successfully completed customer transaction.	= '*Processed*'
item_sub_type_name	Item subtype classification (e.g., RX, OTC, FREE).	*IN ('FREE', 'OTC', 'RX')*
bom_type_name	Bill of Materials type description.	*IS NULL OR <> 'Stücklistenposition'*
sales_price_net_rep_cur	Net reported transaction value (reporting currency), used as a proxy for customer economic value.	*IS NOT NULL*

2.9 Dataset Overview and Key Statistics

Before proceeding to behavioral analysis, a validation step is performed to ensure that the cleaned dataset is complete, consistent, and suitable for downstream analysis. This step verifies the correct application of behavioral scope restrictions, transaction-validity filters, and structural exclusions, while also providing a high-level overview of dataset scale, customer coverage, and transactional structure.

The complete SQL implementation used to construct and validate the final scoped transactional dataset is provided in Appendix A.

Table 2.2 summarizes the final dataset after the application of all scope restrictions and validation rules.

Table 2.2: Overview of the Cleaned Transaction Dataset (24-Month Scope)

Category	Metric	Value
Time	Observation period	December 31, 2023 to December 31, 2025
Scale	Total order lines	122.8 million
	Total orders	45.2 million
	Total customers	10.9 million
Behavior	Average items per order	2.7
	Average orders per customer	4.2
Value	Total transaction value	€2.19 billion

The final dataset consists of approximately 122.8 million order lines, corresponding to 45.2 million completed orders placed by 10.9 million unique customers within the defined 24-month analytical window. The observation period spans from December 31, 2023 to December 31, 2025, ensuring full coverage of two complete annual business cycles.

From a structural perspective, the dataset reflects substantial customer activity and a strong repeat-purchase component. On average, each order contains approximately 2.7 order lines, while customers place an average of 4.2 orders during the observed period. These patterns indicate that customer interaction is not dominated by isolated transactions, but includes substantial repeat-purchase behavior over time.

The validation process also confirmed strong structural consistency across the dataset. No missing values were identified in critical behavioral identifiers such as *main_order_id_bi* and *super_customer_id_order* after the application of the scoped transactional filters. Remaining records also contain valid transaction-value information, supporting consistency across monetary behavioral metrics.

From a behavioral composition perspective, the dataset contains a balanced mix of FREE, OTC, and RX transactions, with OTC representing the largest behavioral component, followed by FREE interactions and a smaller but strategically important RX segment. This distribution is analytically significant because different product groups are expected to exhibit different ordering rhythms, engagement structures, and churn dynamics. Preserving this behavioral diversity is therefore essential for the segmentation, personalized churn interpretation, and predictive modeling framework developed in the following chapters.

Overall, the resulting dataset provides a controlled analytical basis for the remainder of the thesis. Its size, consistency, and transactional scope support robust behavioral analysis while preserving interpretability across subsequent analytical and predictive stages.

3. Behavioral Understanding Framework

3.1 Behavioral Understanding as a Prerequisite for Churn Definition

Customer churn is often treated as a classification problem in which customers are labeled according to predefined inactivity rules or predictive outcomes. However, without understanding how customer behavior evolves over time, such definitions risk becoming disconnected from actual engagement patterns. Inactivity emerges from purchasing frequency, return timing, engagement consistency, and interaction intensity, all of which may vary across customers and product categories (Reichheld & Sasser, 1990).

This chapter does not aim to define churn directly, but rather to uncover the behavioral patterns that later support churn interpretation. At this stage, no predictive models, churn classifications, or fixed thresholds are applied. Instead, the focus is on understanding purchasing behavior and temporal dynamics that influence long-term customer engagement.

Customer behavior is analyzed across two complementary analytical layers: order-line-level interaction patterns and order-level behavioral dynamics over time (Fader, Hardie, & Lee, 2005). The reusable SQL views supporting this behavioral reconstruction are provided in Appendix B.

3.2 Behavioral Analysis Framework and Layered Structure

Customer activity is analyzed across two complementary layers. The order-line level captures what customers purchase and how they interact with FREE, OTC, and RX product categories. The order level captures how behavior evolves over time through ordering frequency, return timing, and recurring engagement patterns. Together, these layers connect purchasing composition with longitudinal customer behavior.

Across both layers, the analysis focuses on three behavioral dimensions: engagement intensity, purchase frequency, and return behavior. Engagement intensity reflects the depth of customer interaction, purchase frequency captures the regularity of ordering activity, and return behavior describes how quickly customers return after a completed purchase.

This structure is necessary because customer interaction patterns differ across product categories and purchasing motivations. As a result, identical inactivity periods may represent different behavioral states depending on the underlying customer context.

3.3 The Role of Order-Line Level in Behavioral Interpretation

3.3.1 Justification for Order-Line Level Analysis

Although churn is ultimately interpreted at the customer level and behavioral dynamics evolve over time at the order level, these patterns cannot be fully understood without examining what customers purchase. The order-line level provides the product-interaction context required to interpret customer behavior, including product composition, category engagement, and purchasing structure within individual orders.

This phase is particularly relevant in an online pharmacy environment, where RX, OTC, and FREE interactions reflect different behavioral mechanisms. RX-related purchases are often associated with recurring and necessity-driven behavior, while OTC and FREE interactions tend to be more discretionary and variable over time. Similar inactivity periods may therefore represent different behavioral situations depending on the underlying product context.

For this reason, the order-line level functions as a contextual layer rather than a direct churn-definition layer. Churn itself is later evaluated through temporal behavioral dynamics such as recency, frequency, and return behavior, but these signals require product-interaction context to be interpreted correctly.

3.3.2 Product Subtype Composition and Its Behavioral Implications

This analysis examines how customer activity is distributed across the three core commercial product categories: FREE, OTC, and RX. Rather than treating all transactions as behaviorally equivalent, the objective is to identify differences in customer reach, transaction volume, and economic contribution across product subtypes. These differences provide important context for later segmentation because each product category reflects a distinct interaction structure.

Table 3.1 summarizes the distribution of customer activity across product subtypes in terms of transactional volume, customer reach, and revenue contribution.

Table 3.1: Customer Reach and Revenue Contribution Across Product Types

Product Subtype	Order Lines (M)	Orders (M)	Customers (M)	Customer Share (%)	Revenue Share (%)
FREE	51.1	29.3	8.7	46.86	40.69
OTC	63.0	29.4	8.4	44.97	33.23
RX	8.8	4.8	1.5	8.17	26.06

To further illustrate the relationship between customer reach and economic contribution across product categories, Figure 3.1 visualizes the relative contribution of each subtype to the customer base and overall transaction value.

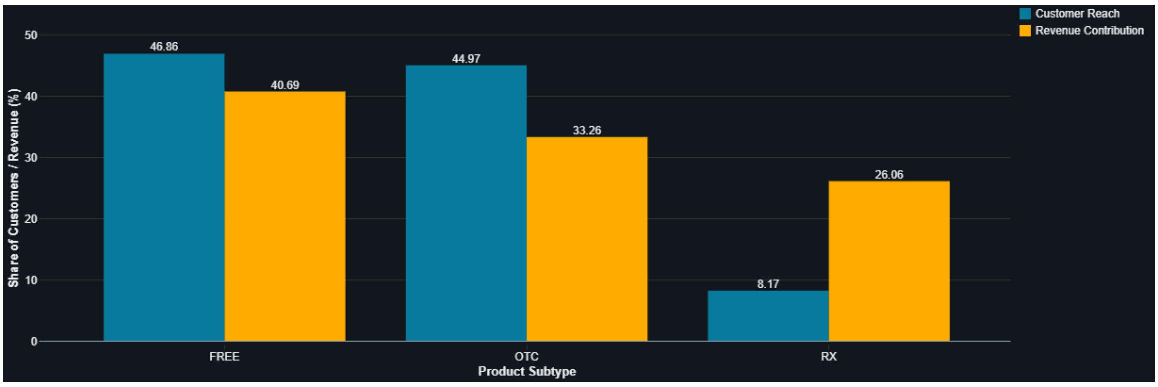

Figure 3.1: Customer Reach and Revenue Contribution Across Product Subtypes

The distribution reveals a clear divergence between customer reach and economic contribution. FREE and OTC interactions dominate the customer base, together representing more than 91% of customers, while RX-related interactions account for a much smaller customer share but contribute disproportionately to total transaction value.

This pattern indicates that RX behavior differs structurally from FREE and OTC interaction patterns. Customers interacting primarily through FREE or OTC products may exhibit more variable and lower-intensity purchasing behavior, whereas RX-related behavior is associated with more economically significant interactions. Product context therefore becomes an important input for later segmentation and churn interpretation.

3.3.3 Customer Order Frequency and Behavioral Maturity

Order frequency provides an initial indication of customer engagement depth. Customers with repeated purchasing behavior offer more reliable behavioral signals than customers with only one observed interaction, making frequency an important input for later churn interpretation.

The analysis therefore examines how customers are distributed across order-frequency levels within the 24-month observation window.

Table 3.2 summarizes the distribution of customers across order-frequency buckets within the 24-month observation window.

Table 3.2: Customer Distribution Across Order Frequency Levels

Order Frequency Bucket	Customers (M)	Customer Share (%)
1 order	4.0	37.19
2–3 orders	3.0	27.51
4–5 orders	1.4	12.65
6–10 orders	1.5	13.72
11–20 orders	0.8	7.04
21–50 orders	0.2	1.83
50+ orders	0.01	0.07

To further illustrate the concentration of customer activity across frequency levels, Figure 3.2 visualizes the percentage distribution of customers by order-frequency bucket.

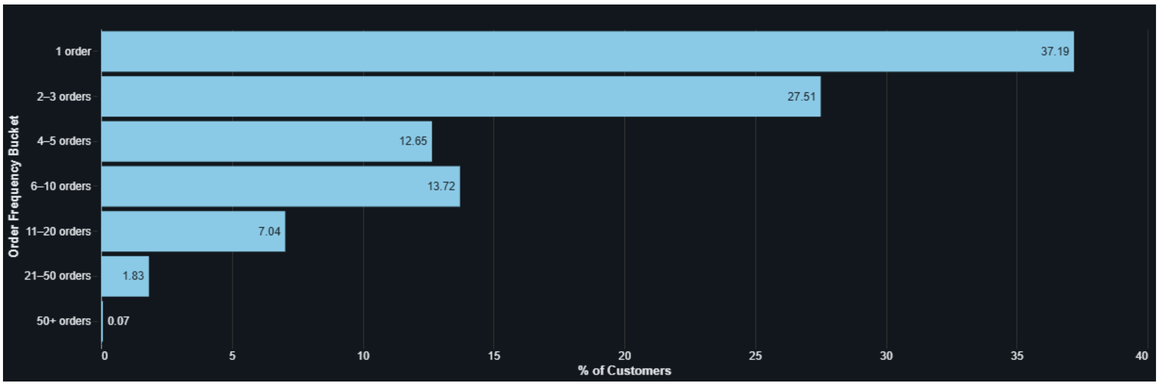

Figure 3.2: Customer Base is Dominated by Low-Frequency Purchasing Behavior

The distribution reveals a strongly concentrated engagement structure dominated by low-frequency purchasing behavior. More than one-third of customers place only a single order, while nearly two-thirds place no more than three orders during the observed period. This indicates that a large share of platform interaction remains acquisition-oriented or behaviorally shallow rather than characterized by stable long-term engagement.

Customer concentration declines rapidly as purchasing frequency increases, forming a clear long-tail structure. From a churn perspective, this distinction matters because one-time customers provide limited evidence of recurring behavior, while repeat customers establish observable return patterns that can support more meaningful inactivity interpretation.

3.3.4 Entry Product Mix and Initial Customer Interaction Patterns

This analysis examines the product composition of the first and only order placed by one-time customers. The objective is not to evaluate churn outcomes directly, but to identify which product interactions are most associated with early-stage, non-recurring customer activity.

Initial order composition provides useful behavioral context because RX, OTC, and FREE interactions reflect different levels of purchasing commitment and expected recurrence.

Table 3.3 summarizes the distribution of one-time customers across different entry product-mix combinations.

Table 3.3: Entry Product Mix Distribution of One-Time Customers

Product Mix	Customers (M)	Customer Share (%)
FREE Only	1.5	37.40
OTC Only	1.2	29.72
FREE + OTC	1.1	26.52
RX Only	0.2	4.44

RX + OTC	0.05	1.15
FREE + RX + OTC	0.02	0.39
FREE + RX	0.02	0.38

To further illustrate the concentration of customer entry behavior across product combinations, Figure 3.3 visualizes the percentage distribution of one-time customers by entry product mix.

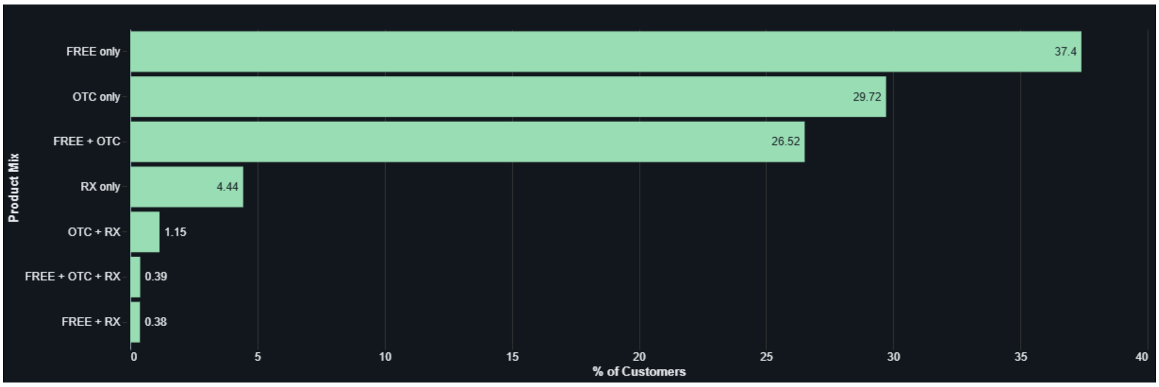

Figure 3.3: Entry Product Mix Distribution of One-Time Customers

The distribution reveals a highly concentrated acquisition structure dominated by FREE and OTC interactions. More than 93% of one-time customers enter the platform through FREE Only, OTC Only, or FREE + OTC combinations.

RX-related entry behavior represents only a small share of one-time activity, suggesting that RX-driven relationships are less frequently associated with isolated interactions. Entry product composition therefore provides early context for distinguishing low-commitment acquisition behavior from potentially more stable engagement trajectories.

3.4 Order-Level Behavioral Dynamics Over Time

3.4.1 Structural Distribution of Repeat Purchase Behavior

The analysis now explores how repeat purchasing behavior develops over time across the customer base. The aim is to determine whether customer return activity tends to cluster within specific time periods or remains broadly distributed throughout the observed timeframe.

Rather than relying on predefined churn assumptions, this step evaluates observed return behavior after completed purchases. This temporal structure is essential because churn interpretation depends on when repeat activity normally occurs.

Table 3.4 summarizes the cumulative share of repeat orders observed at selected inter-order thresholds across the full behavioral horizon.

Table 3.4: Cumulative Distribution of Repeat Orders Across Key Return Thresholds

Return Threshold (Days)	Cumulative Share of Repeat Orders (%)
7	10.08
14	18.52
30	36.46
60	59.84
90	73.21
180	89.83
365	98.08
540	99.66
731	100.00

To complement the milestone-based summary, Figure 3.4 visualizes the cumulative accumulation of repeat purchasing behavior across the complete observed return horizon.

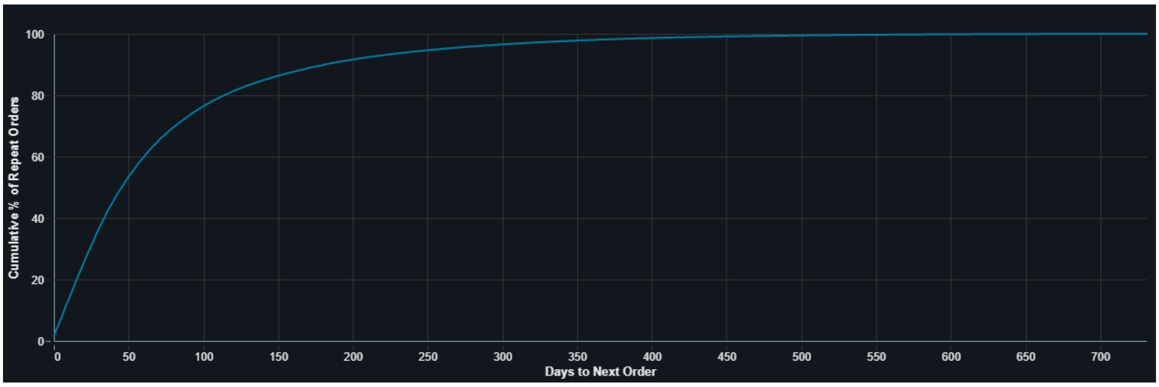

Figure 3.4: Cumulative Repeat-Order Distribution Across Return Windows

The distribution shows that repeat purchasing activity accumulates rapidly in the earlier part of the observed horizon before progressively flattening. More than one-third of repeat orders occur within 30 days, nearly three-quarters within 90 days, and almost all observed repeat purchasing behavior occurs within one year.

This pattern indicates that customer return behavior is not evenly distributed over time. Although late returns remain observable, their incremental contribution becomes limited beyond the first year. For churn interpretation, this supports the use of empirically observed return behavior rather than arbitrary inactivity assumptions.

3.4.2 Customer Purchase Frequency Distribution (Orders per Customer)

While inter-order timing explains how quickly customers return, purchase frequency describes the depth of engagement over the full observation horizon. This distinction is important because inactivity has different meanings for customers with different historical purchasing baselines.

Figure 3.5 illustrates the distribution of customer purchase frequency across the observed population.

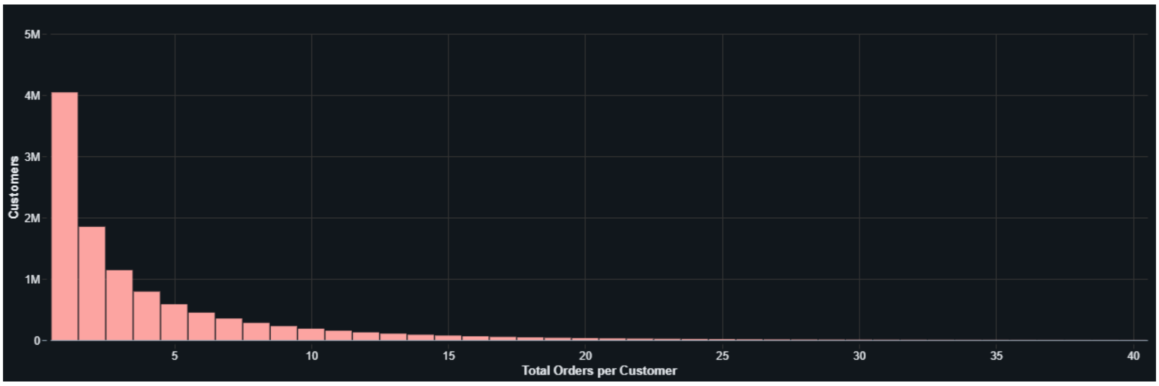

Figure 3.5: Customer Purchase Frequency Exhibits a Strong Long-Tail Distribution

To complement the raw frequency distribution, Figure 3.6 visualizes the cumulative concentration of customers as purchase frequency increases.

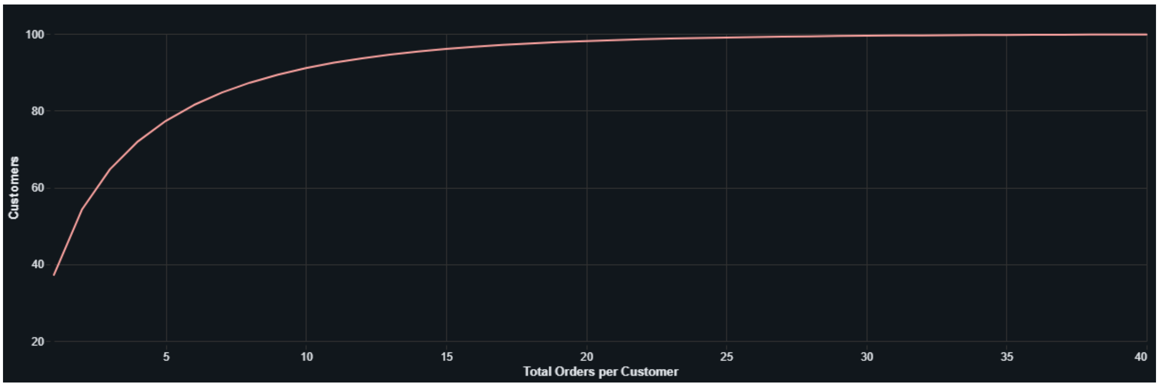

Figure 3.6: Cumulative Distribution of Customer Purchase Frequency

The distribution is strongly right-skewed, with most customers concentrated at low order-frequency levels and only a small share exhibiting highly frequent purchasing behavior. The cumulative view reinforces this long-tail structure by showing that consistently recurring engagement is limited to a relatively narrow part of the customer base.

Purchase frequency therefore becomes an important behavioral baseline for later churn interpretation. The same inactivity period may be less informative for low-frequency customers but more meaningful for customers with historically frequent ordering behavior.

3.4.3 Behavioral Structure and Interpretation of Customer Inactivity

The previous analyses show that repeat purchasing behavior differs both in timing and frequency. Customer inactivity therefore does not carry a fixed meaning across the population: the same elapsed time since the last order may be expected for one customer group while signaling disengagement for another.

This section identifies the main behavioral dimensions that shape inactivity interpretation before formal churn thresholds are introduced.

Return Timing Structure

The first dimension concerns how quickly customers typically return after a completed order. Inter-order activity is grouped into discrete return windows representing different stages of customer re-engagement.

Figure 3.7 summarizes the distribution of repeat orders across inter-order time buckets within the first observed year after purchase.

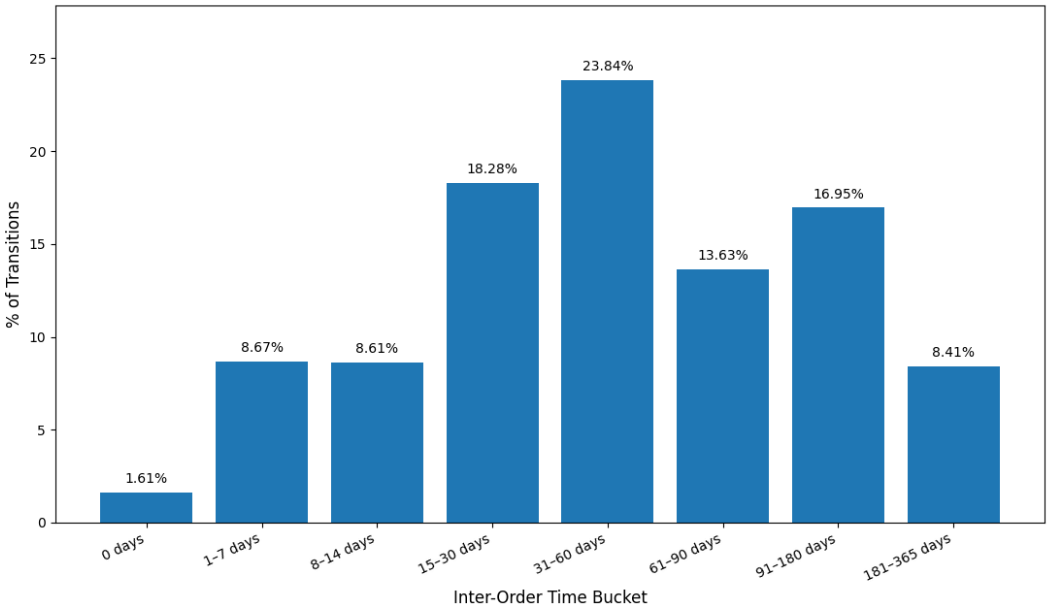

Figure 3.7: Repeat Orders Cluster Around Distinct Behavioral Return Windows

Repeat purchasing activity is concentrated within specific return windows rather than spread evenly over time. The strongest concentration occurs between 31 and 60 days, followed by 15 to 30 days and 91 to 180 days. Very early reorders and very late returns represent smaller shares of total repeat activity.

This structure indicates that inactivity becomes more meaningful once customers move beyond the dominant return windows observed across the population.

Return Accumulation Dynamics

The cumulative return structure adds a second perspective by showing how quickly repeat activity accumulates across the full horizon. As shown earlier, most repeat activity occurs relatively early, while additional accumulation slows over time. This means that the informational value of inactivity changes across the customer lifecycle.

Behavioral Stability and Ordering Consistency

Beyond timing and frequency, customer inactivity is also shaped by the consistency of individual purchasing behavior over time. Some customers exhibit relatively stable inter-order patterns, while others alternate between short bursts of activity and long inactive periods. To evaluate this variability, customers are grouped into behavioral stability segments based on the consistency of their inter-order gaps.

Table 3.5: Behavioral Stability Characteristics by Consistency Segment

Behavioral Consistency Segment	Customer Share (%)	Avg Gap (Days)	Median Gap (Days)	IQR (Days)
Very Consistent	7.40	130.7	122.0	26.3
Consistent	16.00	103.1	90.4	59.3
Variable	56.03	83.6	63.7	87.6
Highly Variable	20.57	78.1	28.2	111.4

Figure 3.8 complements the structural statistics by illustrating how customers are distributed across these behavioral stability segments.

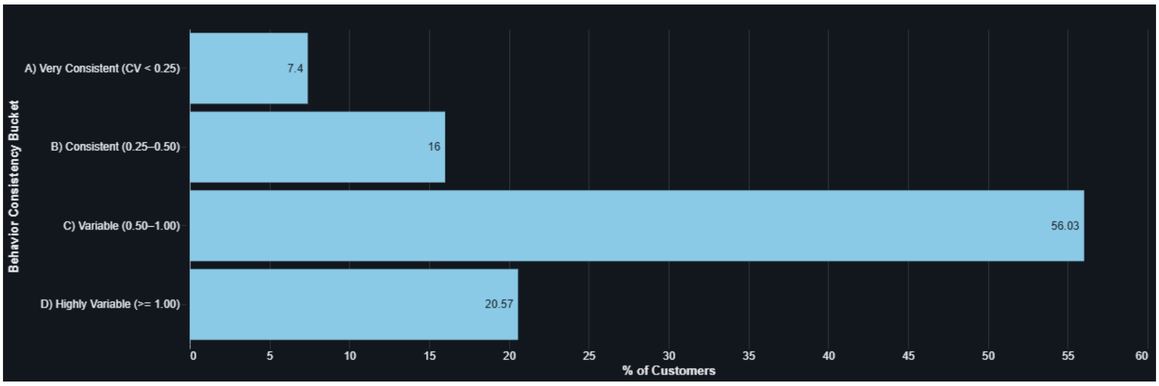

Figure 3.8: Customer Behavior is Predominantly Variable Rather Than Consistent

The results show that highly stable purchasing behavior represents a relatively small portion of the customer base, while most customers exhibit moderate or high variability in their ordering patterns. This confirms that repeat purchasing behavior is structurally heterogeneous.

For churn interpretation, consistency matters because deviations from expected rhythm are more informative for stable customers than for highly variable customers. Together, return timing, return accumulation, and behavioral consistency demonstrate why inactivity must later be interpreted relative to behavioral context rather than through a single universal threshold.

3.4.4 Customer Order Type Profile

While the previous sections focused on when and how frequently customers return, churn interpretation also requires understanding what customers purchase. FREE, OTC, and RX products reflect different purchasing contexts, engagement structures, and commercial characteristics.

At the order level, customers often interact across multiple product categories within the same transaction. Table 3.6 summarizes the distribution of observed order compositions before these patterns are translated into simplified customer-level behavioral profiles.

Table 3.6: Distribution of Order Composition Across Product Type Combinations

Order Composition	Orders (M)	Share of Orders (%)
FREE + OTC	15.8	35.02
FREE Only	12.6	27.75
OTC Only	12.0	26.60
RX Only	2.9	6.41
RX + OTC	1.0	2.30
FREE + RX + OTC	0.5	1.03
FREE + RX	0.4	0.88

Figure 3.9 complements the table by visualizing the relative distribution of observed order compositions across the transactional base.

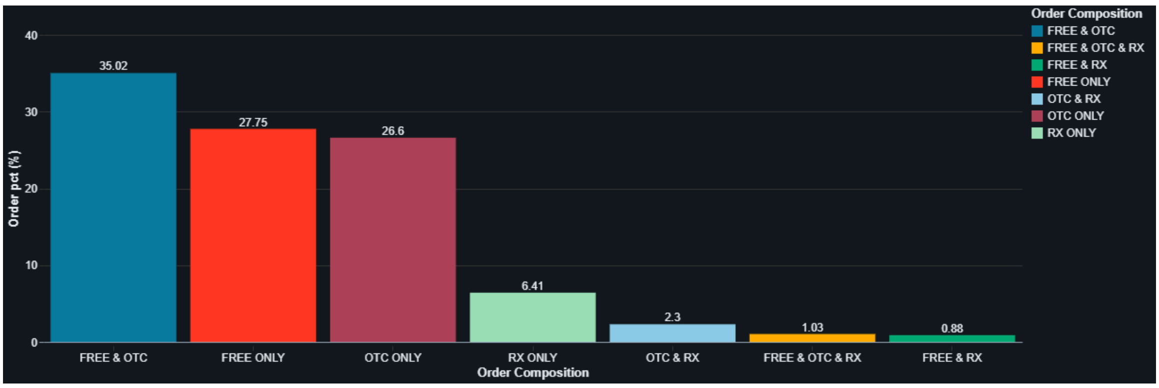

Figure 3.9: Customer Orders Are Structurally Mixed Across Product Types

The results show that customer interaction is structurally mixed rather than isolated by product category. The dominant order pattern combines FREE and OTC products, while FREE Only and OTC Only purchases also represent substantial shares of order activity. Purely RX-driven and mixed RX combinations remain comparatively smaller.

Because raw order composition is analytically rich but difficult to use directly for churn interpretation, these patterns are translated into a simplified customer-level segmentation logic. The abstraction follows a business-aligned hierarchy in which RX interaction dominates OTC and FREE behavior, while OTC dominates purely FREE interaction patterns.

Figure 3.10 illustrates the deterministic classification logic used to assign each customer to a single dominant behavioral profile.

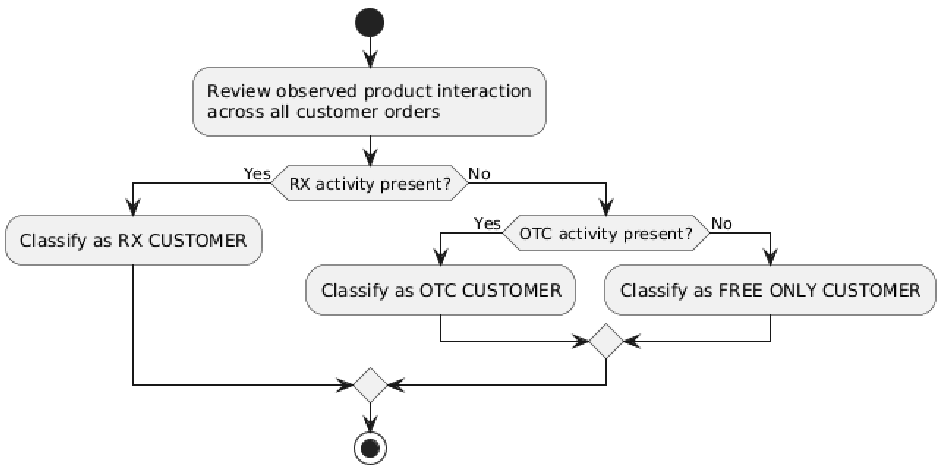

Figure 3.10: Business-Aligned Hierarchical Logic for Customer Order Type Classification

The resulting customer-level abstraction translates observed transactional behavior into mutually exclusive behavioral profiles that remain grounded in the actual composition of customer orders while supporting clearer analytical interpretation.

Table 3.7 summarizes the final customer distribution across these behavioral segments.

Table 3.7: Customer Distribution Across Behavioral Order Type Segments

Customer Segment	Customers (M)	Customer Share (%)
OTC CUSTOMER	7.2	66.21
FREE Only CUSTOMER	2.2	19.81
RX CUSTOMER	1.5	13.98

Figure 3.11 visualizes the relative size of each behavioral customer segment.

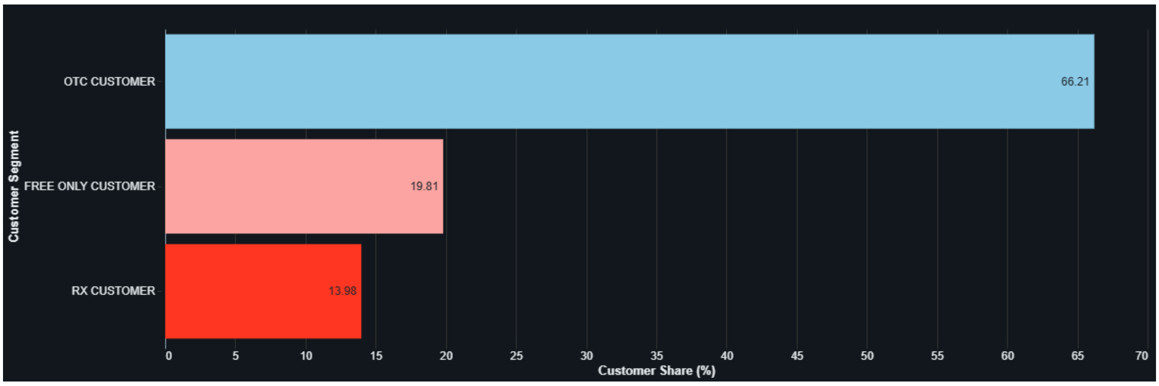

Figure 3.11: Customer Distribution by Behavioral Order Type Profile

The resulting segmentation shows a strong dominance of OTC CUSTOMER behavior across the customer base, while RX CUSTOMER represents a smaller but structurally distinct segment. FREE Only CUSTOMER forms a separate low-engagement profile characterized by lower-complexity interaction patterns.

The hierarchical abstraction ensures that any observed RX interaction elevates the customer into the RX CUSTOMER segment, reflecting the behavioral and commercial significance of prescription-related purchasing. This distinction is important because RX CUSTOMER behavior follows different engagement structures from OTC CUSTOMER and FREE Only CUSTOMER behavior.

3.4.5 Internal Behavioral Structure of Customer Segments in Recency-Frequency Space

After assigning customers to order-type segments, the analysis examines how activity is distributed within each segment over time. Segment membership alone does not imply uniform engagement: customers in the same segment may still differ in recency, frequency, and inactivity patterns.

To make this internal variation visible, customers are positioned within a recency–frequency framework defined by how recently and how frequently they interact with the platform. This helps assess whether each segment is behaviorally concentrated or internally dispersed.

Segment-Level Engagement Baseline

Before examining the detailed recency – frequency distributions, a high-level behavioral baseline is established for each customer segment. The goal of this step is not to define churn behavior directly, but to provide an initial directional comparison of engagement intensity across the previously defined order-type segments.

Table 3.8: Segment-Level Engagement Baseline Across Customer Segments

Customer Segment	Customers (M)	Avg Orders per Customer	Avg Recency (Days)
OTC CUSTOMER	7.2	4.28	223.41

FREE Only CUSTOMER	2.2	1.61	324.52
RX CUSTOMER	1.5	7.23	143.57

The segment-level baseline reveals a clear engagement gradient. RX CUSTOMER exhibits the highest average purchase frequency and lowest recency, while FREE Only CUSTOMER shows infrequent purchasing and longer inactivity. OTC CUSTOMER occupies an intermediate position. However, these averages do not show how behavior is distributed internally within each group, making a more granular recency–frequency view necessary.

Internal Recency-Frequency Structure of the RX CUSTOMER Segment

The RX CUSTOMER segment shows a strong concentration of high-frequency and recent activity, indicating structurally stable and recurring engagement behavior.

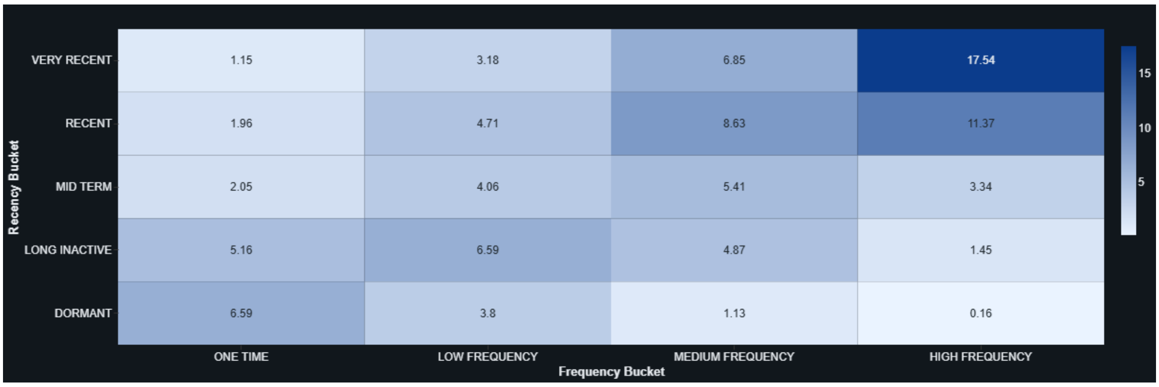

Figure 3.12: RX CUSTOMER Engagement Concentration

RX CUSTOMER activity is strongly concentrated in recent and high-frequency zones, confirming the recurring nature of prescription-related engagement. However, some customers are distributed across mid-term, long inactive, and dormant states.

Internal Recency-Frequency Structure of the OTC CUSTOMER Segment

The OTC CUSTOMER segment exhibits broad behavioral dispersion across engagement states, reflecting structurally heterogeneous interaction patterns.

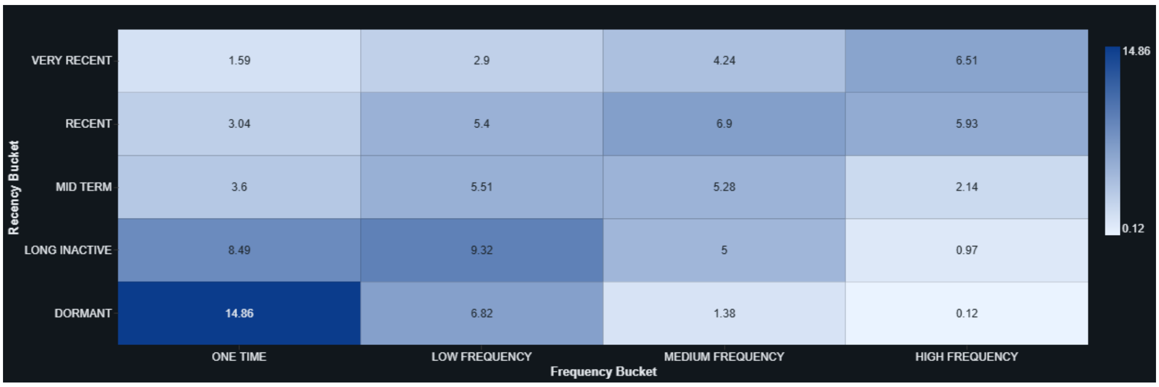

Figure 3.13: OTC CUSTOMER Engagement Concentration

OTC CUSTOMER shows the broadest dispersion across recency and frequency combinations. The segment includes highly engaged customers, occasional purchasers, and customers with extended inactivity. This heterogeneity makes OTC CUSTOMER particularly sensitive to oversimplified inactivity thresholds.

Internal Recency-Frequency Structure of the FREE Only CUSTOMER Segment

The FREE Only CUSTOMER segment is concentrated in low-engagement and inactive states, indicating structurally weak and discontinuous interaction behavior.

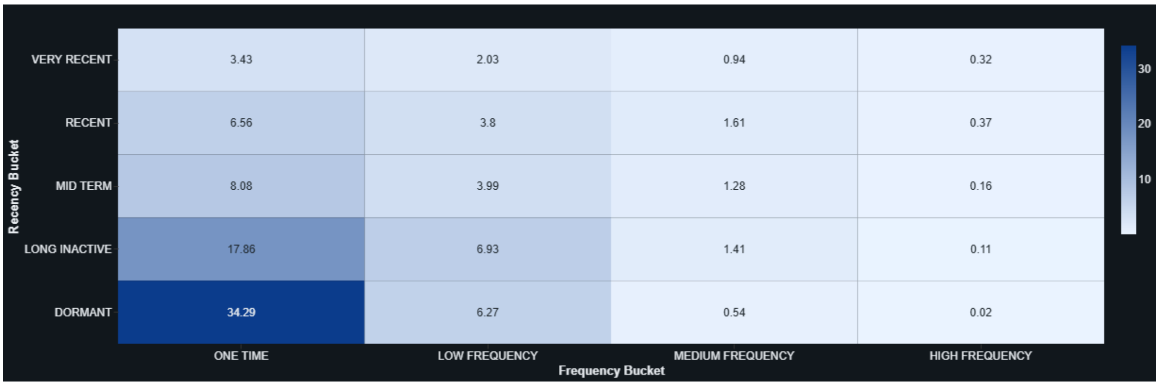

Figure 3.14: FREE CUSTOMER Engagement Concentration

FREE Only CUSTOMER is concentrated in low-frequency and dormant zones, with a strong presence of one-time and highly inactive customers. Sustained engagement is limited, indicating a structurally different retention context from RX-driven behavior.

Cross-Segment Behavioral Interpretation

The recency–frequency analysis shows that customer segments are internally heterogeneous and cannot be described through average engagement metrics alone. RX CUSTOMER contains the strongest active core but also includes customers moving toward lower activity states. OTC CUSTOMER is the most dispersed segment, while FREE Only CUSTOMER is concentrated in low-engagement and dormant zones.

These findings create an important constraint for churn interpretation: inactivity must be evaluated together with purchasing intensity and segment context. A single inactivity threshold applied uniformly across the customer base would ignore this internal behavioral structure.

Engagement-Based Refinement Within the RX CUSTOMER Segment

Because RX CUSTOMER is commercially important but not behaviorally uniform, the segment is further divided into RX Only and RX + OTC. This refinement assesses whether prescription-only customers and customers combining prescription and non-prescription activity exhibit materially different engagement patterns.

Table 3.9: Internal Behavioral Comparison Within the RX CUSTOMER Segment

RX Subtype	Customers	Customer Share (%)	Avg Orders per Customer	Avg Recency (Days)
RX + OTC	1,171,951	77.10	8.75	111.58
RX Only	348,028	22.90	2.11	251.27

The comparison reveals a substantial divergence within RX CUSTOMER. RX + OTC customers exhibit stronger engagement, combining higher purchase frequency with lower recency. This suggests a broader and more continuous relationship with the platform beyond isolated prescription fulfillment.

RX Only customers display lower purchasing frequency and higher recency, indicating a more episodic interaction structure. This confirms that even the RX CUSTOMER segment contains important internal variation, requiring additional behavioral context for churn interpretation.

3.4.6 Return Probability by Segment - Empirical Time-to-Next-Order Analysis

Following the recency – frequency analysis, the next step is to examine how quickly customers return after a completed purchase across the main behavioral segments (Schmittlein, Morrison, & Colombo, 1987). This analysis measures empirical return probability across selected time horizons using observed inter-order transitions.

The focus is not to define churn thresholds directly, but to estimate the natural return rhythm of each segment. If return dynamics differ materially across customer groups, the same inactivity period cannot carry the same meaning across the population.

Table 3.10: Segment-Level Return Probability and Time-to-Next-Order Statistics

Customer Segment	Observed Order Transitions (M)	Return Within 30d (%)	Return Within 60d (%)	Return Within 90d (%)	Return Within 180d (%)	Return Within 365d (%)	Avg Days to Next Order	Median Days to Next Order
OTC CUSTOMER	23.6	32.90	56.54	70.58	88.56	97.82	81.55	51
RX CUSTOMER	9.5	47.40	70.89	82.52	94.81	99.30	54.58	33
FREE Only CUSTOMER	1.3	21.47	39.32	53.19	76.75	93.96	123.81	83

The segment-level return statistics reveal clear and structurally different temporal engagement patterns across the customer base. RX CUSTOMER exhibits the fastest and most concentrated return behavior, with nearly half of all observed transitions followed by another purchase within 30 days and more than 80% occurring within 90 days. The relatively short median return interval confirms that recurring engagement is structurally embedded within RX-driven behavior.

OTC CUSTOMER follows a more moderate return trajectory. Return accumulation is gradual, and inactivity periods are longer compared to RX CUSTOMER, suggesting a more flexible and demand-driven interaction cycle.

FREE Only CUSTOMER displays the slowest return structure. Return probability accumulates gradually, and both average and median inter-order intervals remain elevated, indicating limited repeat engagement and structurally lower purchasing continuity.

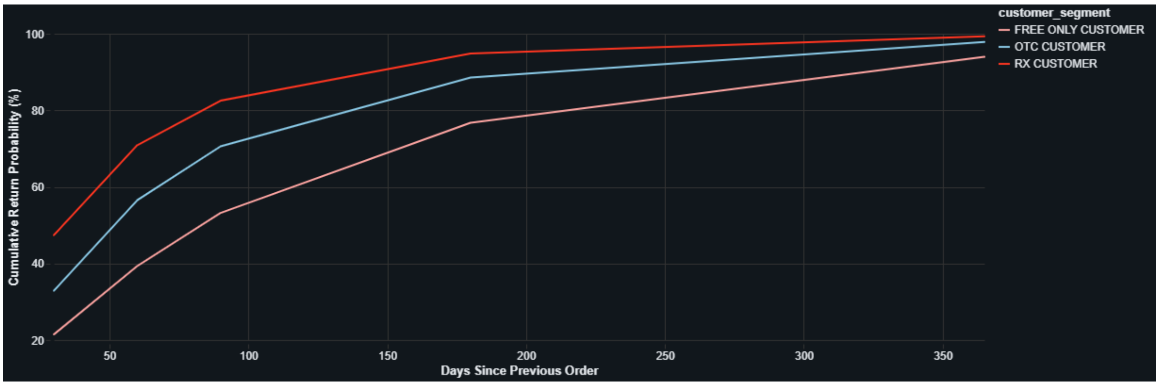

Figure 3.15: Return Behavior Varies Structurally Across Customer Segments

The cumulative return curves reinforce these differences. RX CUSTOMER demonstrates a steep early return trajectory, OTC CUSTOMER follows a more gradual accumulation pattern, and FREE Only CUSTOMER exhibits a flatter and delayed return structure.

These differences are central for churn interpretation. The same inactivity duration may signal early disengagement for RX CUSTOMER while remaining closer to the expected return rhythm of FREE Only CUSTOMER or parts of the OTC CUSTOMER population. Churn thresholds must therefore be calibrated relative to segment-specific return dynamics.

Return-Dynamics Refinement Within the RX CUSTOMER Segment

Although RX CUSTOMER exhibits the strongest overall return behavior, the segment may still contain different temporal engagement patterns. To evaluate this, RX CUSTOMER is refined into RX Only and RX + OTC subgroups.

Table 3.11: Return Probability Comparison Within the RX CUSTOMER Segment

RX Subtype	Observed Order Transitions (M)	Return Within 30d (%)	Return Within 60d (%)	Return Within 90d (%)	Return Within 180d (%)	Avg Days to Next Order	Median Days to Next Order
RX + OTC	9.1	48.05	71.69	83.22	95.17	53.12	32
RX Only	0.4	32.10	51.91	65.87	86.51	88.70	57

The refinement reveals a clear divergence in return dynamics. RX + OTC customers return faster and more consistently, with nearly half of observed transitions followed by another purchase within 30 days and more than 80% within 90 days.

RX Only customers return more slowly across all observed horizons, with higher average and median inter-order intervals. This suggests a more episodic interaction structure driven primarily by prescription-related need cycles.

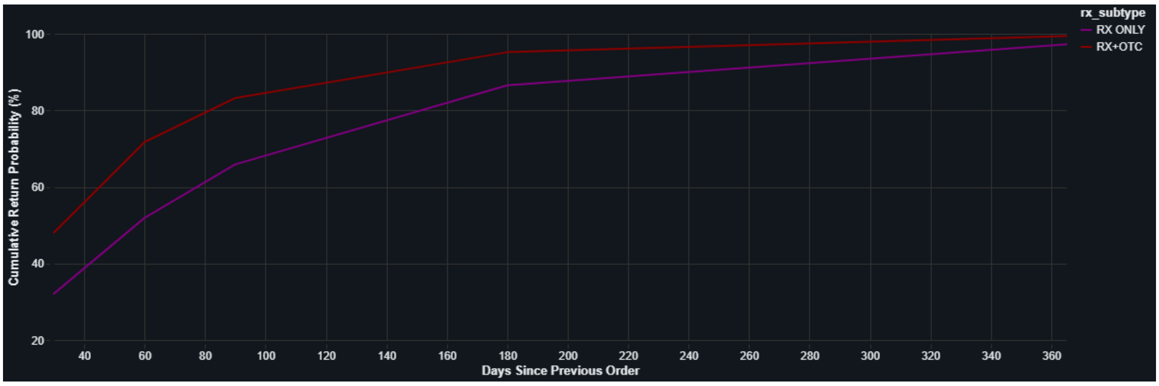

Figure 3.16: Cumulative Return Probability Within the RX CUSTOMER Segment

The cumulative return curves make this divergence visible. RX + OTC customers exhibit faster return accumulation, while RX Only customers follow a flatter and delayed trajectory. This confirms that churn interpretation must remain sensitive not only across primary customer segments, but also within structurally distinct RX subgroups.

3.5 Behavioral Synthesis and Implications for Churn Definition

The analyses conducted throughout this chapter show that customer behavior does not follow a single uniform engagement structure. Activity differs across purchasing frequency, return timing, engagement intensity, product interaction, and behavioral consistency.

RX CUSTOMER behavior is characterized by faster and more concentrated return activity, OTC CUSTOMER behavior shows broader and more heterogeneous engagement patterns, and FREE Only CUSTOMER behavior is dominated by low-frequency interaction and prolonged inactivity. Internal variation is also visible within segments, particularly in the distinction between RX Only and RX + OTC customers.

These findings establish the behavioral basis for churn interpretation. A globally applied inactivity threshold would ignore the observed structure of customer behavior and increase the risk of systematic misclassification. Churn must therefore be interpreted relative to expected engagement rhythm, segment membership, return behavior, and customer position within the broader behavioral structure.

Building on this foundation, the following chapter introduces a structured segmentation framework that translates these observed behavioral dynamics into operationally interpretable customer groups.

4. Behavioral Segmentation Framework

4.1 From Behavioral Understanding to Segmentation Design

The behavioral analysis in Chapter 3 showed that customer activity does not follow a single uniform engagement pattern. Purchasing frequency, return timing, inactivity duration, product interaction, and behavioral consistency differ materially across the customer base. These differences indicate that customer behavior cannot be interpreted through a single common norm (Blattberg, Kim, & Neslin, 2008).

However, the observed behavioral complexity is difficult to use directly in downstream churn logic. Without a structured interpretation layer, differences in product interaction, recency, frequency, and return behavior remain analytically fragmented and difficult to translate into consistent business rules.

This chapter addresses that gap by transforming the behavioral patterns identified in Chapter 3 into a structured segmentation framework. The purpose of segmentation is not to create descriptive marketing personas, but to organize customer behavior into stable, explainable, and reusable analytical groups that can support churn definition, predictive modeling, and operational decision-making.

4.2 Segmentation Design Principles

The segmentation framework is designed to balance analytical rigor with practical usability. Rather than increasing segmentation complexity, the focus is on building a stable behavioral structure that supports churn interpretation, predictive modeling, and informed business decision-making.

The first design principle is behavioral grounding. Customer groups are derived from observed purchasing behavior rather than demographic assumptions, externally imposed personas, or pre-defined marketing categories. This ensures that segmentation remains linked to measurable customer interaction patterns and preserves consistency across downstream analytical layers (Wedel & Kamakura, 2000).

The second principle is interpretability. Each segment must reflect a recognizable behavioral profile that can be understood by both technical and business stakeholders. A segmentation framework that is analytically sophisticated but difficult to explain would have limited practical value in an enterprise setting.

The third principle is controlled abstraction. The proposed approach avoids unnecessary micro-segmentation and instead focuses on the distinctions that materially affect customer interpretation and churn evaluation. This preserves usability while still capturing the major behavioral structures observed in the data.

Finally, the framework remains extensible. Where meaningful heterogeneity persists within a segment, additional refinement layers can be introduced without disrupting the overall architecture. Together, these principles position segmentation as a behavioral interpretation layer rather than a purely descriptive categorization exercise.

4.3 Primary Behavioral Segmentation Based on Product Interaction

The first segmentation layer is based on dominant product interaction, reflecting the role of FREE, OTC, and RX products in shaping customer behavior. Chapter 3 showed that these product interactions are associated with different engagement structures, return dynamics, and levels of purchasing continuity.

RX interaction is treated as behaviorally dominant because it is associated with more structured and recurring engagement. In the absence of RX activity, OTC interaction defines the customer’s primary behavioral segment, while FREE Only interaction represents the lowest structural engagement category.

To capture these differences in a stable and operationally interpretable way, customers are assigned to a single primary segment using a deterministic hierarchical classification logic. RX interaction is treated as behaviorally dominant due to its structurally recurring and commercially significant nature. In the absence of RX activity, customers are classified based on OTC interaction, while FREE Only interaction represents the lowest structural engagement category. This results in three primary customer segments:

- RX CUSTOMER
- OTC CUSTOMER
- FREE Only CUSTOMER

This deterministic hierarchy creates a business-aligned abstraction layer that translates complex product interaction patterns into stable customer groups. Rather than representing every product combination as a separate segment, the framework prioritizes the distinctions that are most relevant for churn interpretation and downstream decision logic.

The primary segmentation layer does not assume that all customers within a segment behave identically. It establishes the structural starting point for behavioral interpretation, while allowing additional refinement where the evidence shows meaningful internal heterogeneity.

4.4 Intra-Segment Behavioral Refinement of the RX CUSTOMER Segment

The primary segmentation layer captures the main structural differences across customer groups, but it does not fully eliminate internal heterogeneity. This is particularly important within RX CUSTOMER, where Chapter 3 revealed a clear distinction between customers interacting exclusively through prescription products and customers combining prescription and non-prescription activity.

RX Only customers exhibit a more episodic and prescription-specific interaction pattern, with lower purchase frequency and longer inactivity periods. In contrast, RX + OTC customers show stronger engagement continuity, higher order frequency, and faster return behavior, indicating a broader relationship with the platform.

To preserve this behavioral distinction, RX CUSTOMER is refined into two subgroups:

- RX Only

- RX + OTC

This refinement is introduced as an interpretive extension of the primary segmentation layer. Its purpose is not to increase segmentation complexity, but to preserve behavioral fidelity where the empirical evidence shows materially different engagement patterns.

The refinement demonstrates a key principle of the framework: segmentation should remain simple enough to be usable, but flexible enough to capture heterogeneity when it affects churn interpretation.

4.5 Recency-Frequency Behavioral Layer

Product-based segmentation defines the structural nature of customer interaction, but it does not describe the customer's current behavioral position. Customers within the same segment may differ substantially in how recently and how frequently they interact with the platform.

To capture this variation, the framework introduces a recency–frequency behavioral layer. Recency represents the time elapsed since the customer's most recent completed order, while frequency reflects the level of purchasing activity observed during the analysis window. Interpreted together, these dimensions provide a more complete view of customer engagement than either measure alone.

This layer is not used to create additional fixed customer segments. Instead, it contextualizes each customer's current position within the product-based segmentation structure. Its role is interpretive: it helps distinguish customers who share the same structural segment but occupy different engagement positions, such as active, declining, inactive, or dormant behavioral states.

This distinction is important for the following chapters because churn states are not assigned directly from recency–frequency categories. They are later defined through segment-specific return thresholds and personalized behavioral baselines.

4.6 Integrated Behavioral Segmentation Framework

The final segmentation framework integrates the behavioral dimensions established throughout this chapter into a layered architecture. Rather than relying on a single segmentation axis, the framework combines structural product interaction, RX-specific refinement, and recency–frequency positioning.

At the first level, customers are grouped according to dominant product interaction. Where meaningful internal heterogeneity remains, refinement layers are introduced, as shown in the distinction between RX Only and RX + OTC. Finally, the recency–frequency layer provides a temporal view of the customer's current engagement position.

The integrated segmentation framework can therefore be summarized as follows:

Table 4.1: Integrated Behavioral Segmentation Framework and Interpretive Role of Each Layer

Framework Layer	Behavioral Logic	Output	Role in Behavioral Interpretation
Primary Product-Based Segmentation	Hierarchical interaction logic (RX > OTC > FREE)	RX CUSTOMER / OTC CUSTOMER / FREE Only CUSTOMER	Defines the structural customer context
RX Behavioral Refinement	Distinguishes prescription-only from mixed RX engagement	RX Only / RX + OTC	Captures behavioral heterogeneity within the RX segment
Recency-Frequency Behavioral Layer	Combines inactivity timing and engagement intensity	Current engagement position	Contextualizes current engagement within the product-based segment

This layered architecture keeps the segmentation framework explainable and reusable while preserving the main behavioral differences observed in the customer base. Each layer contributes a distinct interpretive role: product-based segmentation defines the structural customer context, RX refinement captures meaningful heterogeneity within prescription-related behavior, and the recency–frequency layer adds the customer’s current engagement position.

Because the framework is grounded in observable transactional behavior, it can be implemented within the existing analytical infrastructure without requiring opaque modeling assumptions or externally imposed personas. It therefore provides a stable basis for the churn definition logic developed in the following chapter.

4.7 Segmentation Readiness for Churn Definition

The segmentation framework developed in this chapter establishes the interpretive context required for churn definition. Product-based segmentation defines the structural nature of customer interaction, RX refinement captures meaningful heterogeneity within prescription-related behavior, and the recency–frequency layer describes the customer’s current engagement position.

Together, these layers make it possible to interpret inactivity relative to customer context rather than through a single global threshold. This is the necessary transition from behavioral observation to churn-state definition.

The following chapter builds on this segmentation framework by formalizing churn definitions that are explicitly aligned with the observed behavioral differences across customer groups.

5. General Segmented Churn Definition

5.1 Customer-Level Churn Base and Behavioral Overview

The analysis now moves from behavioral segmentation toward the first operational layer of churn interpretation. The objective is to evaluate whether customer inactivity can be interpreted consistently across behaviorally different customer groups.

A reusable customer-level churn base is established for this purpose. This layer consolidates the behavioral attributes required for churn interpretation, including customer-level order frequency, last observed order date, and recency relative to a fixed analytical reference point. Each customer is also mapped to the segmentation framework introduced in Chapter 4, including the RX Only and RX + OTC refinement within RX CUSTOMER.

Table 5.1 provides a customer-level overview of segment distribution, engagement frequency, and recency characteristics. This overview establishes the analytical basis for evaluating whether inactivity can realistically be interpreted through a universal rule.

Table 5.1: Behavioral Comparison Across Customer Segments

Customer Segment	RX Subtype	Customers (M)	Customer Share (%)	Avg Orders per Customer	Avg Recency Days	Median Recency Days
RX CUSTOMER	RX + OTC	1.2	10.78	8.75	111.58	55
RX CUSTOMER	RX Only	0.3	3.20	2.11	251.27	226
OTC CUSTOMER	–	7.2	66.21	4.28	223.41	161
FREE Only CUSTOMER	–	2.2	19.81	1.61	324.52	305

Several important behavioral differences emerge immediately. RX + OTC customers represent the most engaged population, combining the highest average order frequency with the lowest median recency. In contrast, RX Only customers show materially lower purchasing frequency and longer inactivity periods despite belonging to the same primary RX CUSTOMER segment.

The divergence is also visible across OTC CUSTOMER and FREE Only CUSTOMER. OTC CUSTOMER shows moderate recurring engagement, while FREE Only CUSTOMER combines the lowest order frequency with the longest inactivity horizon. The median recency of more than 300 days suggests that prolonged inactivity may be structurally normal for a large part of this segment rather than an immediate churn signal.

The gap between average and median recency across segments further indicates right-skewed inactivity distributions influenced by highly inactive customer subsets. These differences establish the need to test whether a unified inactivity threshold can produce coherent customer-state interpretation.

The reusable SQL views supporting the churn foundation and behavioral state assignment logic are provided in Appendix C.

5.2 Why a Unified Churn Definition Fails: Initial Stress Test

To evaluate the limitations of a simple churn definition, a unified 90-day inactivity threshold is applied across the full customer population. This approach reflects a common operational shortcut: customers are classified only according to the number of days since their last observed interaction.

The rule classifies customers with recency less than or equal to 90 days as ACTIVE and customers exceeding 90 days as CHURNED. No segment-specific context, return expectation, or purchasing-frequency difference is incorporated. The purpose is not to establish the final churn definition, but to stress-test whether a one-size-fits-all rule can represent structurally different customer groups.

Table 5.2: Unified Threshold Produces Imbalanced Customer-State Distribution Across Segments

Customer Segment	RX Subtype	ACTIVE (%)	CHURNED (%)
FREE Only CUSTOMER	–	19.06	80.94
OTC CUSTOMER	–	36.51	63.49
RX CUSTOMER	RX + OTC	63.26	36.74
RX CUSTOMER	RX Only	28.88	71.12

Table 5.2 shows that the unified threshold produces materially different customer-state distributions across segments. Rather than creating a stable interpretation of inactivity, the same 90-day rule behaves very differently depending on the underlying engagement structure.

The distortion is most visible in FREE Only CUSTOMER, where more than 80% of customers are classified as CHURNED. Given the low-frequency and delayed return behavior observed earlier, this result likely reflects premature churn classification rather than widespread customer abandonment. A similar pattern appears in OTC CUSTOMER, where nearly two-thirds of customers are classified as CHURNED despite the segment's naturally broader purchasing rhythm.

The RX CUSTOMER segment initially appears more balanced, but the RX refinement reveals a different picture. RX + OTC customers remain predominantly ACTIVE, while RX Only customers are mostly classified as CHURNED. This confirms that even within the same primary segment, a fixed inactivity threshold can conceal important behavioral differences.

Operationally, such distortions would affect retention prioritization and intervention timing. Low-frequency customers could be over-targeted, while faster-returning customer groups may require earlier detection logic. Figure 5.1 visualizes the same classification results using normalized segment-level distributions.

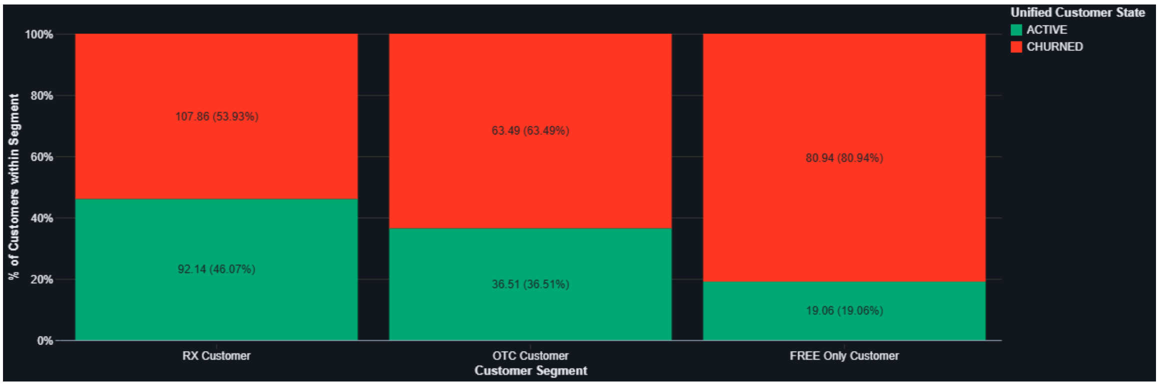

Figure 5.1: Customer-State Distribution Under a Unified 90-Day Inactivity Threshold

The stress test confirms that inactivity cannot be interpreted independently of behavioral context. A globally applied threshold fails to produce meaningful interpretation of inactivity across customer segments and supports the use of segment-specific churn definitions.

5.3 Empirical Return Milestones as the Basis for Churn Definition

Having shown the limitations of a unified threshold, the next step is to derive churn reference points from observed return behavior. Instead of relying on arbitrary calendar-based rules, this section uses empirical return milestones to identify when inactivity remains within normal segment behavior and when it becomes increasingly atypical.

Three milestones are extracted from each segment’s inter-order gap distribution. P50 represents the typical return horizon, P80 captures the upper boundary within which most customers still return, and P90 marks the point beyond which inactivity becomes increasingly unusual for the segment. These thresholds are derived from observed customer behavior rather than imposed externally.

Table 5.3: Segment-Level Empirical Return Milestones (P50, P80, P90)

Customer Segment	P50 Days	P80 Days	P90 Days
RX CUSTOMER	33	83	127
OTC CUSTOMER	51	124	194
FREE Only CUSTOMER	83	201	299

Table 5.3 shows clear differences in return horizons across segments. RX CUSTOMER has the fastest and most concentrated return behavior, with a median return time of 33 days and a P90 threshold of 127 days. This indicates that extended inactivity becomes unusual relatively quickly within this segment.

OTC CUSTOMER follows a broader return structure, with a median of 51 days and a P90 threshold of 194 days. Longer inactivity periods therefore remain behaviorally plausible for a larger share of this segment.

FREE Only CUSTOMER shows the slowest return rhythm, with a median return time of 83 days and a P90 threshold approaching 300 days. This confirms that extended inactivity is structurally embedded in the segment's normal behavior and should not be interpreted through the same timing logic used for RX CUSTOMER.

The widening distance between P50, P80, and P90 across segments also indicates increasing variability in return behavior. These milestones provide the behavioral basis for defining segment-specific customer states in the next section.

5.4 Segment-Specific Behavioral State Definition

The empirical return milestones provide the foundation for translating inactivity into segment-specific behavioral states. These states are designed to reflect actual return behavior while remaining consistent and practical for downstream business applications.

A key design decision concerns the boundary between normal inactivity and early churn risk. Using P50 as the ACTIVE boundary would make the framework highly sensitive, but it would also classify many customers as at risk simply because they return later than the segment median. This would increase false-positive interpretation, especially in segments with broader return distributions.

For this reason, the framework adopts a more conservative P80/P90 design. Customers remain ACTIVE while their recency stays within the segment-specific P80 return milestone. Customers between P80 and P90 are classified as AT RISK, representing meaningful deviation but still within a potentially recoverable intervention window. Customers beyond P90 are classified as CHURNED, as their inactivity exceeds the late-return horizon observed for the large majority of the segment.

This design balances sensitivity with false-positive control. It avoids labeling customers too early while still identifying a targeted intervention window before inactivity becomes structurally atypical. This is particularly important in an e-commerce pharmacy environment, where purchasing rhythms vary across prescription-related, OTC, and FREE Only interactions.

Table 5.4: Segment-Specific Behavioral State Definition Logic

Behavioral State	Threshold Logic	Interpretation	Business Meaning
ACTIVE	Recency ≤ P80	Customer remains within the expected return horizon of the segment	No churn signal; normal monitoring
AT RISK	P80 < Recency ≤ P90	Customer has moved beyond typical return behavior but remains within the late-return range	Actionable intervention window
CHURNED	Recency > P90	Customer is beyond the segment-specific late-return threshold	Reactivation / win-back logic

The resulting state model supports differentiated business action. ACTIVE customers remain within their expected segment-level return horizon. AT RISK customers have moved beyond typical return behavior but remain within an actionable intervention window. CHURNED customers have exceeded the segment-specific late-return boundary and are better suited for reactivation or win-back logic.

At this stage, states are intentionally defined at the level of the primary behavioral segments. This preserves clarity before introducing customer-level personalization in Chapter 6.

5.5 Application of Segment-Specific Churn States

The segment-specific P80/P90 logic is then applied across the customer base to evaluate how customers distribute across ACTIVE, AT RISK, and CHURNED states. This step does not introduce new rules; it applies the thresholds defined in Section 5.4 to each customer based on their segment.

The goal is to evaluate whether segment-specific thresholds provide a more realistic interpretation of customer inactivity than the unified 90-day rule used earlier.

Table 5.5: Customer-State Distribution Under Segment-Specific Threshold Logic

Customer Segment	ACTIVE Customers	ACTIVE (%)	AT RISK Customers	AT RISK (%)	CHURNED Customers	CHURNED (%)
RX CUSTOMER	811,240	53.37	147,418	9.70	561,321	36.93
OTC CUSTOMER	3,153,285	43.82	816,892	11.35	3,226,312	44.83

FREE Only CUSTOMER	763,747	35.48	293,787	13.65	1,095,212	50.88

The goal is to evaluate whether segment-specific thresholds provide a more realistic interpretation of customer inactivity than the unified 90-day rule used earlier.

Table 5.5 shows that segment-specific thresholds produce a more balanced and behaviorally aligned customer-state distribution than the unified threshold. RX CUSTOMER retains the strongest ACTIVE base, consistent with its shorter and more concentrated return horizon.

OTC CUSTOMER shows a more balanced distribution across ACTIVE, AT RISK, and CHURNED states, reflecting its broader and more flexible return behavior. FREE Only CUSTOMER remains the least continuously engaged segment, but the framework no longer classifies the overwhelming majority of the segment as churned solely because of extended inactivity.

Figure 5.2 visualizes the resulting proportional customer-state distribution across segments.

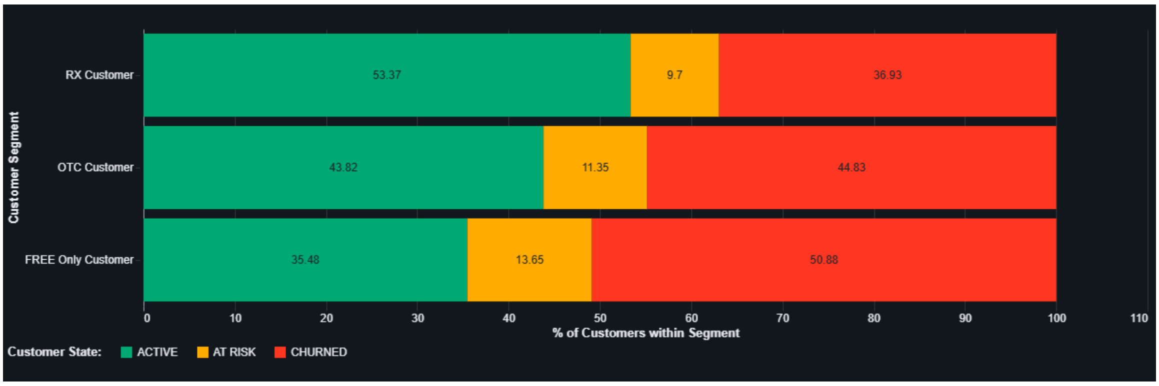

Figure 5.2: Customer-State Distribution Under Segment-Specific Thresholds

Under the segment-specific framework, inactivity is no longer treated as an absolute number of elapsed days. Customer states are evaluated relative to the expected return structure of the corresponding segment, creating a more context-aware interpretation of churn risk.

However, the results also show the remaining limitation of segment-level logic. Customers within the same segment may still differ in ordering rhythm, engagement intensity, and return expectations. This motivates the move toward personalized churn interpretation in Chapter 6.

5.6 Cross-Segment Interpretation of Inactivity

The practical meaning of segment-specific churn logic becomes clear when identical inactivity durations are compared across customer segments. This section does not introduce additional churn rules; it illustrates how the same elapsed time without an order is interpreted differently depending on segment-specific return expectations.

Table 5.6: Cross-Segment Interpretation of Identical Inactivity Durations

Inactivity Days	RX CUSTOMER	OTC CUSTOMER	FREE Only CUSTOMER
60	ACTIVE	ACTIVE	ACTIVE
100	AT RISK	ACTIVE	ACTIVE
150	CHURNED	AT RISK	ACTIVE
220	CHURNED	CHURNED	AT RISK
300	CHURNED	CHURNED	CHURNED

Table 5.6 shows that inactivity is a context-dependent signal rather than an absolute indicator. At 100 inactivity days, RX CUSTOMER has already entered the AT RISK range, while OTC CUSTOMER and FREE Only CUSTOMER remain ACTIVE. At 150 days, RX CUSTOMER is classified as CHURNED, OTC CUSTOMER as AT RISK, and FREE Only CUSTOMER remains ACTIVE.

The interpretation gradually converges at higher inactivity durations. By 300 days, all segments are classified as CHURNED, but the path toward this state differs substantially by segment. RX CUSTOMER reaches meaningful inactivity much earlier, while FREE Only CUSTOMER transitions more gradually due to its slower return rhythm.

This illustrates the central value of the segment-specific framework: churn states are assigned relative to observed return behavior rather than through a uniform calendar rule. At the same time, the example also exposes the limitation of group-level interpretation, since individual customers within the same segment may still have different personal return rhythms.

5.7 Business Interpretation and Actionability

The segment-specific churn framework translates behavioral states into differentiated business actions. Once inactivity is interpreted relative to segment-specific return dynamics, customer states can support intervention timing, retention prioritization, and engagement strategy design (Buckinx & Van den Poel, 2005).

The states should be understood as levels of behavioral deviation rather than static labels. ACTIVE customers remain within expected segment behavior, AT RISK customers represent a meaningful but potentially recoverable deviation, and CHURNED customers have moved beyond the expected return horizon.

Table 5.7: Mapping of Behavioral States to Business Actions

Customer State	Behavioral Meaning	Business Objective	Typical Actions
ACTIVE	Customer remains within expected return behavior	Maintain engagement and prevent gradual drop-off	Routine communication, lightweight personalization
AT RISK	Customer shows meaningful deviation from expected return timing	Prevent churn through timely intervention	Targeted reminders, replenishment nudges, incentive-based retention
CHURNED	Customer has moved beyond expected return horizon	Reactivate or re-acquire customer	Win-back campaigns, stronger incentives, reactivation strategies

The states should be understood as levels of behavioral deviation rather than static labels. ACTIVE customers remain within expected segment behavior, AT RISK customers represent a meaningful but potentially recoverable deviation, and CHURNED customers have moved beyond the expected return horizon.

Table 5.7 links each behavioral state to a corresponding business objective. The proposed approach does not assume that every inactive customer requires the same intervention intensity. Instead, treatment can be aligned with the behavioral meaning of inactivity within each segment.

The AT RISK state is the most strategically important intervention window. It identifies customers whose behavior has moved beyond the expected return range but has not yet reached the late-return boundary. This creates a more usable balance between intervening too early and reacting too late.

From a business perspective, the value of the framework lies in prioritization. It helps distinguish normal behavioral variability from meaningful disengagement, supporting more focused retention actions and reducing unnecessary campaign noise.

The proposed approach remains a segment-level baseline and improves interpretability compared to unified thresholds, but it does not yet capture individual customer return expectations. This limitation leads directly to the personalized churn logic.

5.8 Limitations of Segment-Level Definitions and Transition to Personalization

The segment-specific churn framework represents a substantial improvement over unified inactivity threshold logic because it aligns customer-state interpretation with observed segment-level return behavior. However, it remains an aggregated interpretation layer.

Customers within the same segment may still differ in purchasing cadence, behavioral consistency, and expected return rhythm. As a result, two customers with the same segment and the same inactivity duration may not necessarily carry the same churn signal if their historical interaction patterns differ materially.

This limitation motivates the transition toward a more personalized interpretation of churn. Rather than replacing the segment-specific framework, the next chapter builds on it by introducing customer-level behavioral expectations derived from each customer’s historical ordering behavior.

The analysis therefore shifts from segment-aware churn interpretation to personalized behavioral state definition, where inactivity is evaluated in relation to each customer’s expected purchasing rhythm and behavioral consistency profile.

6. Personalized Churn Definition

6.1 From Segment-Level to Customer-Level Interpretation

Chapter 5 established that segment-specific churn definitions provide a more coherent interpretation of inactivity than a single global threshold. However, segment-level logic still relies on aggregated behavioral expectations and cannot fully capture variation within each customer group.

Customers within the same segment may follow different ordering rhythms, interaction frequencies, and consistency patterns. A highly regular customer may show meaningful disengagement after a short delay, while a naturally irregular customer may remain within expected behavior despite a longer inactivity period.

The objective of personalization is therefore to extend segment-level churn interpretation with customer-level behavioral expectations derived from each customer's own historical ordering pattern. Segmentation provides the structural context; personalization refines the interpretation by evaluating whether the customer has deviated from their own expected rhythm.

The reusable SQL views and personalization layers supporting this chapter are provided in Appendix C.

6.2 Defining Individual Behavioral Baselines

The first step in personalized churn interpretation is to define a customer-level behavioral baseline. While segment-level thresholds capture broad return patterns, they cannot fully represent the personal ordering rhythm of each customer.

The proposed approach derives this baseline from historical inter-order gaps, where each gap represents the number of days between two completed orders for the same customer. The median inter-order gap is selected as the primary baseline metric because it provides a stable representation of typical return timing while limiting the influence of unusually short or long gaps.

Personalization is applied only where sufficient behavioral history exists. Customers with fewer than two observed inter-order gaps do not provide enough evidence to infer a reliable return rhythm and therefore remain within the segment-level fallback logic introduced in Chapter 5.

Table 6.1 summarizes the resulting personalization eligibility structure across the customer base.

Table 6.1: Baseline Eligibility and Personalization Coverage

Baseline Eligibility	Customers (M)	% of Customers	Avg Gap Count	Median Gap Count	Avg Median Gap Days	Median of Median Gap Days
Fallback to Segment Level	5.9	54.21	0.31	0	189.05	145
Eligible for Personalization	5.0	45.79	6.53	4	67.34	54

Table 6.1 shows that approximately 45.8% of customers have sufficient repeat behavior to support personalized interpretation, while 54.2% remain under segment-level fallback logic. This reflects the structure of a real customer population, where many customers interact only occasionally or have not yet developed observable repeat behavior within the analysis window.

Eligible customers exhibit materially richer behavioral histories, with higher gap counts and shorter median return cycles. Fallback customers provide limited recurring evidence, making individualized interpretation unreliable by design. The baseline eligibility layer therefore protects the framework from artificial precision.

6.3 Baseline Reliability and Behavioral Consistency

Defining an individual baseline improves inactivity interpretation, but the baseline alone does not indicate how reliable that interpretation is. Two customers may have the same median return interval while following very different ordering patterns: one stable and predictable, the other highly irregular.

To capture this distinction, the framework introduces a behavioral consistency layer based on the coefficient of variation of inter-order gaps. Lower values indicate more stable ordering behavior, while higher values reflect greater fluctuation around the customer's historical rhythm.

Customers eligible for personalization are grouped into Consistent, Moderately Variable, and Highly Variable categories. These categories do not replace the personalized baseline or directly change churn-state assignment. Instead, they provide a reliability layer indicating how confidently deviations should be interpreted.

Table 6.2: Behavioral Consistency Distribution Among Personalized Customers

Behavioral Consistency	Customers (M)	% of Eligible Customers	Median Gap Count	Median Return Days	Median Coefficient of Variation
Consistent	1.1	21.86	3	92	0.3442

Moderately Variable	2.7	54.47	5	56	0.7404
Highly Variable	1.2	23.65	4	28	1.1931
Undefined Consistency	0.0	0.02	2	0	–

Table 6.2 shows that most eligible customers fall into the Moderately Variable category, while smaller but meaningful shares are classified as Consistent or Highly Variable. This confirms that sufficient history does not automatically imply stable behavior.

Consistency becomes important during churn interpretation because deviation from expected rhythm is more meaningful for historically stable customers than for customers whose ordering behavior has always fluctuated. The consistency layer therefore acts as a confidence signal around personalized churn states rather than as a separate classification mechanism.

6.4 Customer-Level Expected Return Window

The customer-level baseline is then translated into an expected return window. For customers eligible for personalization, this window is defined by the median number of days between historical orders.

Current inactivity is evaluated through the deviation ratio, calculated as current recency divided by the expected return window. A value close to 1.0 indicates that the customer remains near their expected timing, while higher values indicate increasing deviation from historical behavior.

This normalization is important because identical inactivity durations may carry different meanings depending on the customer’s own return rhythm.

Table 6.3 summarizes the expected return structure and deviation behavior across customers eligible for personalization.

Table 6.3: Expected Return and Deviation Summary Among Personalized Customers

Customers (M)	Avg Expected Return Days	Median Expected Return Days	Avg Recency Days	Median Recency Days	Avg Deviation Ratio	Median Deviation Ratio
4.9	67.75	54	121.40	71	5.2455	1.2143

The results show that the median expected return window among personalized customers is 54 days, while median recency is 71 days. The median deviation ratio of 1.2143 indicates that the typical personalized customer is slightly delayed relative to their expected return rhythm, although the higher average deviation ratio suggests a right-skewed distribution influenced by strongly delayed customers.

To contextualize the severity of behavioral deviation, customers are grouped into broader deviation buckets representing increasing levels of inactivity relative to expected behavior.

Table 6.4: Deviation Ratio Distribution Among Personalized Customers

Deviation Bucket	Customers (M)	% of Personalized Customers	Median Expected Return Days	Median Recency Days	Median Deviation Ratio
Within Expected Rhythm	2.2	44.58	70	28	0.4400
Slightly Delayed	0.6	11.61	59	73	1.2297
Clearly Delayed	0.4	7.79	56	97	1.7353
Strongly Delayed	1.8	36.02	37	220	4.7162

Table 6.4 shows clear separation across deviation levels. Approximately 44.6% of personalized customers remain within their expected rhythm, while 36.0% are strongly delayed. The latter group has a median expected return window of 37 days but a median recency of 220 days, indicating substantial deviation from historical behavior.

These results establish the analytical basis for defining personalized churn states in the following section.

6.5 Personalized Behavioral State Definition

The expected return framework provides a customer-level reference point for interpreting inactivity. The next step is to translate the deviation ratio into operational churn states that can support monitoring, prioritization, and intervention.

The personalized framework defines three behavioral states: ACTIVE, AT RISK, and CHURNED. Customers whose inactivity remains within their expected return rhythm are classified as ACTIVE. Customers with moderate deviation enter the AT RISK state, while customers whose inactivity materially exceeds their historical return pattern are classified as CHURNED.

The thresholds are intentionally simple and interpretable. A deviation ratio of 1.0 separates customers operating within their expected rhythm from those beginning to show behavioral delay. A deviation ratio above 1.5 indicates a 50% delay beyond the expected return rhythm and is treated as a materially abnormal deviation. This threshold is an operational design choice rather than a statistically optimized cutoff; future validation through intervention experiments or rolling-window sensitivity analysis would be required before production calibration.

Table 6.5: Personalized Behavioral State Definition Based on Deviation Ratio

Behavioral State	Threshold Logic	Interpretation	Business Meaning
ACTIVE	Deviation Ratio ≤ 1.0	Customer remains within expected return rhythm	No churn signal; normal monitoring
AT RISK	1.0 < Deviation Ratio ≤ 1.5	Customer has moved beyond expected behavior but remains within early deviation range	Actionable intervention window
CHURNED	Deviation Ratio > 1.5	Customer is materially beyond expected return rhythm	Reactivation / win-back logic

Table 6.5 formalizes the personalized churn-state logic by interpreting inactivity relative to each customer's historical behavior. Two customers with identical recency values may therefore receive different churn states depending on their expected return rhythm.

This approach intentionally avoids excessive state fragmentation. Maintaining the same ACTIVE, AT RISK, and CHURNED structure used in the segment-level framework preserves interpretability while allowing the underlying logic to become customer-specific.

Personalized churn states should also be interpreted together with baseline reliability. Customers differ not only in expected return rhythm, but also in how consistently that rhythm has historically been followed. For this reason, the framework combines churn states with behavioral consistency profiles.

Table 6.6: Reliability-Aware Interpretation of Personalized Behavioral States

Behavioral State	Consistency Level	Interpretation	Business Action
AT RISK	Consistent	High-confidence early deviation from expected behavior	Immediate targeted intervention
AT RISK	Moderately Variable	Moderate-confidence signal	Monitor and apply soft activation
AT RISK	Highly Variable	Lower-confidence signal	Cautious monitoring; lower-intensity engagement
CHURNED	Consistent	Strong and reliable churn signal	Aggressive win-back strategy

CHURNED	Moderately Variable	High likelihood of disengagement	Structured reactivation approach
CHURNED	Highly Variable	Lower-confidence churn signal due to irregular behavior	Gradual reactivation; avoid over-triggering

Table 6.6 shows that behavioral consistency affects interpretation confidence rather than churn-state assignment itself. A customer deviating from a historically stable rhythm produces a stronger churn signal than a customer whose interaction pattern has always fluctuated. This reliability layer supports more cautious prioritization and reduces the risk of overreacting to naturally volatile behavior.

6.6 Application of Personalized Churn States

The personalized state logic is then applied to the eligible customer population by assigning each customer to ACTIVE, AT RISK, or CHURNED based on their deviation ratio.

Table 6.7: Personalized Churn State Distribution Among Personalized Customers

Personalized State	Customers (M)	% of Personalized Customers	Median Expected Return Days	Median Recency Days	Median Deviation Ratio
ACTIVE	2.2	44.58	70	28	0.4400
AT RISK	0.6	11.61	59	73	1.2297
CHURNED	2.2	43.80	40	191	3.7879

Table 6.7 shows clear separation across personalized churn states. Approximately 44.6% of customers remain within their expected return rhythm and are classified as ACTIVE, while 43.8% are classified as CHURNED due to substantial deviation from historical behavior. The AT RISK group represents 11.6% of personalized customers and forms the main intervention window between normal variation and prolonged disengagement.

The underlying metrics reinforce this separation. ACTIVE customers have a median recency of 28 days against a median expected return window of 70 days, while CHURNED customers show a median recency of 191 days against an expected return window of only 40 days.

Figure 6.1 visualizes how personalized churn states are distributed across the three primary customer segments. FREE Only CUSTOMER exhibits the weakest behavioral profile, with the highest CHURNED concentration, while RX CUSTOMER shows the strongest overall profile. OTC CUSTOMER occupies a more balanced middle position.

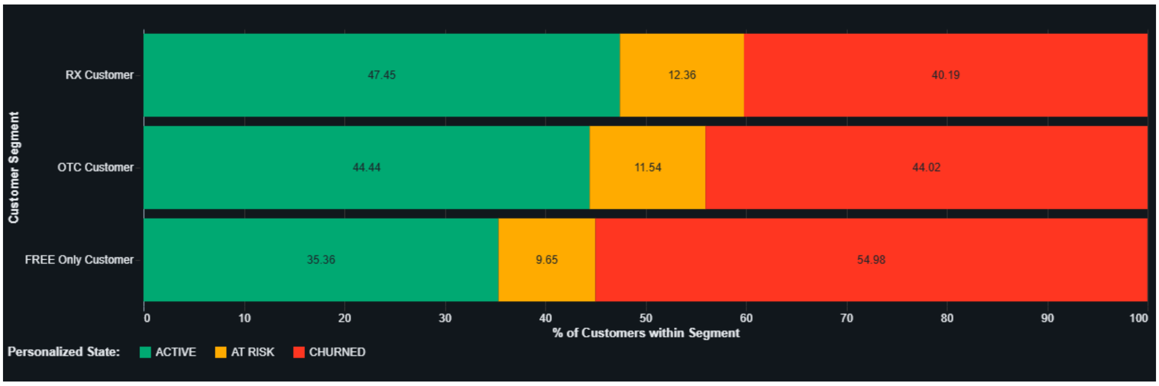

Figure 6.1: Personalized Churn States Reveal Behavioral Differences Across Segments

Figure 6.2 further examines RX subtypes. RX + OTC customers display a healthier engagement profile than RX Only customers, with a higher ACTIVE share and lower CHURNED proportion. This supports the earlier finding that broader interaction with the platform is associated with more stable long-term engagement.

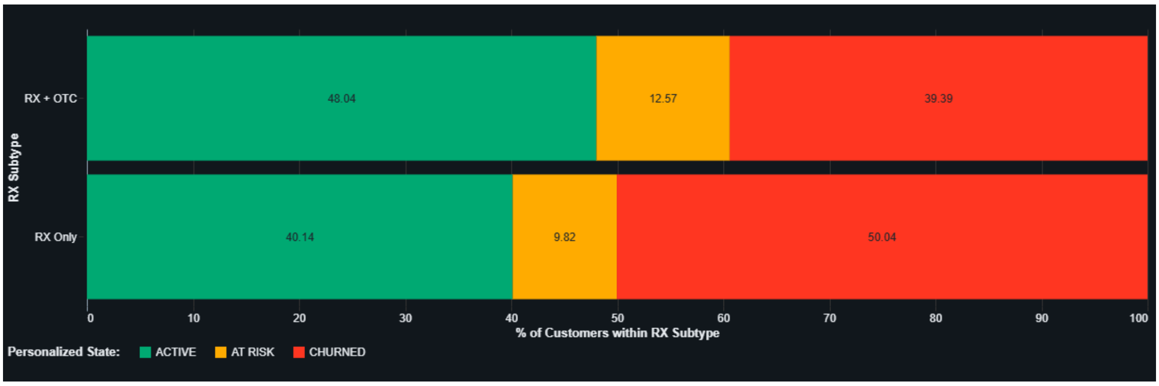

Figure 6.2: RX + OTC Customers Exhibit Stronger Behavioral Stability Than RX Only Customers

Finally, Figure 6.3 adds the reliability-aware perspective by showing how behavioral consistency differs across personalized churn states. CHURNED customers with historically consistent behavior represent stronger churn signals than customers whose interaction patterns have always been highly variable.

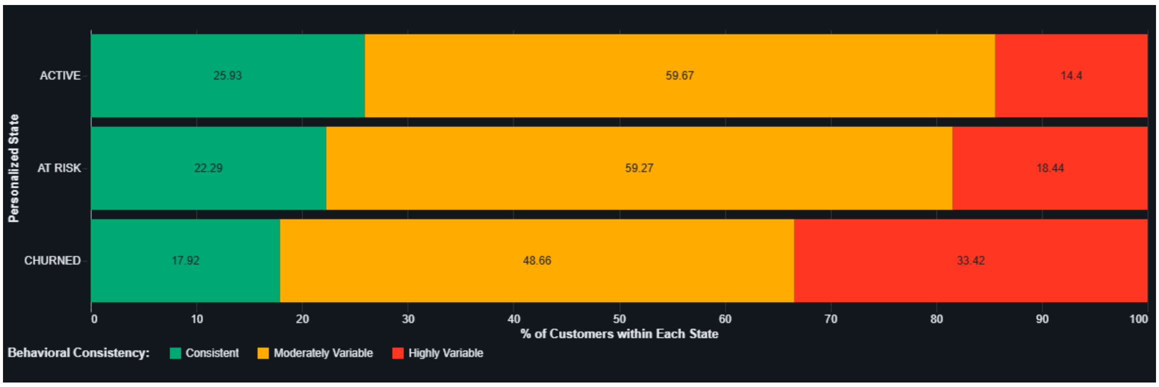

Figure 6.3: Churn Signal Reliability Varies with Behavioral Consistency

6.7 Comparison: Segment-Level vs Personalized Churn Definition

The next step is to compare personalized churn interpretation with the segment-level definition introduced in Chapter 5. The focus is not to determine which framework is universally correct, but to identify where segment-level logic is too broad to capture customer-specific behavioral deviation.

Table 6.8 compares the segment-level state assignment with the personalized churn state for the same customer population and quantifies how customers move between classifications when inactivity is interpreted relative to individual behavioral baselines.

Table 6.8: Reclassification Between Segment-Level and Personalized Churn States

Segment-Level State	Personalized State	Customers (M)	% of Personalized Customers	% within Segment-Level State	Median Deviation Ratio
ACTIVE	ACTIVE	2.1	43.07	66.59	0.4247
ACTIVE	AT RISK	0.4	8.88	13.73	1.2184
ACTIVE	CHURNED	0.6	12.73	19.68	2.3000
AT RISK	ACTIVE	0.1	1.37	11.35	0.8122
AT RISK	AT RISK	0.1	1.95	16.19	1.2523
AT RISK	CHURNED	0.4	8.74	72.46	3.0652
CHURNED	ACTIVE	0.0	0.15	0.63	0.8926
CHURNED	AT RISK	0.0	0.78	3.37	1.3012
CHURNED	CHURNED	1.1	22.34	95.99	6.1905

The off-diagonal transitions reveal where personalization changes churn interpretation:

• ACTIVE → AT RISK highlights hidden early-risk customers whose behavior already deviates from their individual rhythm.

• ACTIVE → CHURNED represents missed churn signals that remain invisible under segment-level interpretation.

• AT RISK → CHURNED indicates late detection, where segment-level logic underestimates the severity of customer-level deviation.

• Reverse transitions remain limited, suggesting that personalization primarily refines churn timing rather than introducing instability.

Table 6.8 shows that personalization materially changes customer-state interpretation, particularly within the ACTIVE and AT RISK populations. Only 66.6% of customers classified as ACTIVE at segment level remain ACTIVE after personalization, while a meaningful share transitions into AT RISK or CHURNED states. Similarly, most customers classified as AT RISK at segment level transition directly into CHURNED under personalized logic, indicating that segment-level interpretation may often identify disengagement relatively late.

By contrast, customers classified as CHURNED at segment level remain highly stable under personalized interpretation. This suggests that segment-level logic performs reasonably well once inactivity becomes strongly prolonged, while personalization adds the greatest value in detecting hidden early risk and improving intervention timing.

Figure 6.4 visualizes these reclassification patterns. The concentration of customers outside the diagonal highlights hidden-risk and late-detection effects, especially within the ACTIVE and AT RISK groups. Overall, the comparison shows that segment-level churn definitions provide a strong structural baseline, while personalized interpretation improves sensitivity to customer-specific behavioral deviation.

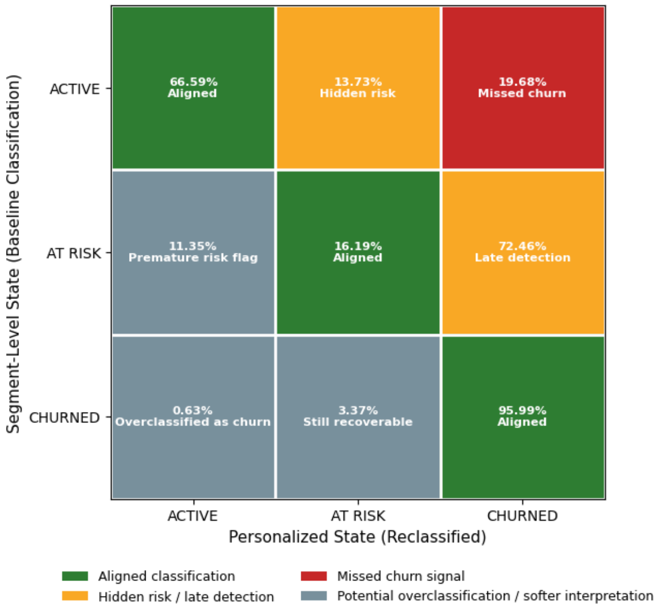

Figure 6.4: Personalization Reveals Hidden Risk and Late Detection Patterns

6.8 Business Implications of Personalized Churn

The personalized churn framework improves the business relevance of churn interpretation by evaluating inactivity relative to each customer's expected return behavior. This shifts churn from a generic elapsed-time rule to a customer-relative behavioral signal.

The main operational benefit is intervention timing. AT RISK customers are identified when their behavior has started to deviate from their normal return rhythm but before inactivity becomes strongly prolonged. This creates a more actionable window for CRM communication, replenishment reminders, and re-engagement actions.

Personalization also improves prioritization. Customers with stable historical behavior and strong deviation signals can be prioritized more confidently, while highly variable customers may require softer activation or additional monitoring. The reliability layer therefore helps distinguish strong behavioral breakpoints from normal volatility.

Within the RX segment, personalization makes the difference between RX + OTC and RX Only customers more operationally visible. RX + OTC customers generally show broader and more continuous engagement, while RX Only customers appear more episodic and may require more cautious reactivation logic.

Overall, the personalized framework provides a stronger analytical basis for retention prioritization by asking not only how long a customer has been inactive, but whether that inactivity is unusual for that specific customer.

6.9 Limitations and Operational Considerations

While the personalized churn framework improves the contextual interpretation of inactivity, it also introduces operational boundaries that must be acknowledged. These limitations do not invalidate the framework; they define the conditions under which personalized churn interpretation remains reliable and explainable.

The framework relies exclusively on observable transactional behavior, including order frequency, inter-order timing, and recency. This preserves explainability but means that external drivers such as health events, seasonality, prescription cycles, or market conditions are not explicitly modeled and may influence inactivity independently of disengagement.

A second limitation concerns customers with insufficient behavioral history. Personalized baseline construction requires at least two observed inter-order gaps. Customers with limited history therefore fall back to the segment-level logic introduced in Chapter 5. This avoids artificial precision but also means that personalization naturally concentrates on customers with richer repeat behavior.

The deviation ratio also requires careful interpretation for customers with historically short return cycles followed by prolonged inactivity. These customers can produce very high deviation values because moderate absolute delays may represent large relative breaks from their expected rhythm. Such values should be interpreted through distributional summaries and reliability context rather than isolated extremes.

From an implementation perspective, the framework should be viewed as a behavioral decision layer rather than a complete activation system. It does not yet incorporate channel constraints, consent management, campaign logic, or downstream CRM execution rules. These components would be required before operational deployment.

Overall, these limitations reflect deliberate design trade-offs. The approach prioritizes interpretability, behavioral transparency, and robustness while creating a foundation that can later be extended through predictive modeling and operational activation layers.

6.10 From Personalized Definition to Evaluation

Chapter 6 extended segment-level churn interpretation by introducing customer-level behavioral expectations. Individual baselines, behavioral consistency, and the deviation ratio allow inactivity to be evaluated relative to each customer's own historical return rhythm rather than only against segment-level thresholds.

The comparison with segment-level definitions showed that personalization materially changes customer-state interpretation, especially by revealing hidden risk among customers classified as ACTIVE and late detection among customers classified as AT RISK. At the same time, the framework remains explainable and controlled by applying personalization only where sufficient behavioral history exists and retaining segment-level fallback logic otherwise.

The next chapter examines which segments are most affected by reclassification, where personalization provides the greatest incremental business value, and how reliability-aware interpretation can support prioritization.

7. Evaluating the Impact of Personalized Churn Definitions Across Behavioral Segments

7.1 Evaluation Objective and Analytical Scope

Following the definition of personalized churn states in Chapter 6, the analysis now evaluates their impact. The objective is to measure how customer-state interpretation changes when inactivity is evaluated relative to individual behavioral baselines rather than segment-level thresholds.

The chapter examines four evaluation dimensions: overall reclassification impact, segment-level concentration of reclassification, reliability of the resulting signals, and operational prioritization logic. These dimensions are summarized in Table 7.1.

Table 7.1: Evaluation Dimensions and Analytical Focus

Evaluation Layer	Analytical Objective	Business Relevance
Overall reclassification impact	Measure how frequently customer states change under personalized interpretation	Quantify the practical impact of personalization
Segment-level reclassification impact	Identify where reclassification is concentrated across behavioral segments	Detect high-impact retention opportunities
Reliability-aware evaluation	Assess the behavioral stability of reclassification signals	Improve intervention confidence and prioritization
Operational interpretation	Translate reclassification patterns into decision-support logic	Support retention allocation and customer lifecycle management

The chapter does not introduce a new churn definition. Instead, it evaluates the consequences of applying the personalized framework across the comparable customer population. Predictive modeling is intentionally excluded at this stage; the focus remains on assessing the behavioral and operational relevance of personalized churn interpretation before moving to future-state prediction in Chapter 8. The reusable analytical layers supporting this evaluation are provided in Appendix C.

7.2 Population-Level Reclassification Impact

The first evaluation step measures the overall magnitude of reclassification introduced by personalization. The analysis compares segment-level and personalized churn states for the same population of customers with sufficient behavioral history.

Customers are grouped into two categories: those whose state remains unchanged and those whose state changes after customer-level behavioral context is introduced.

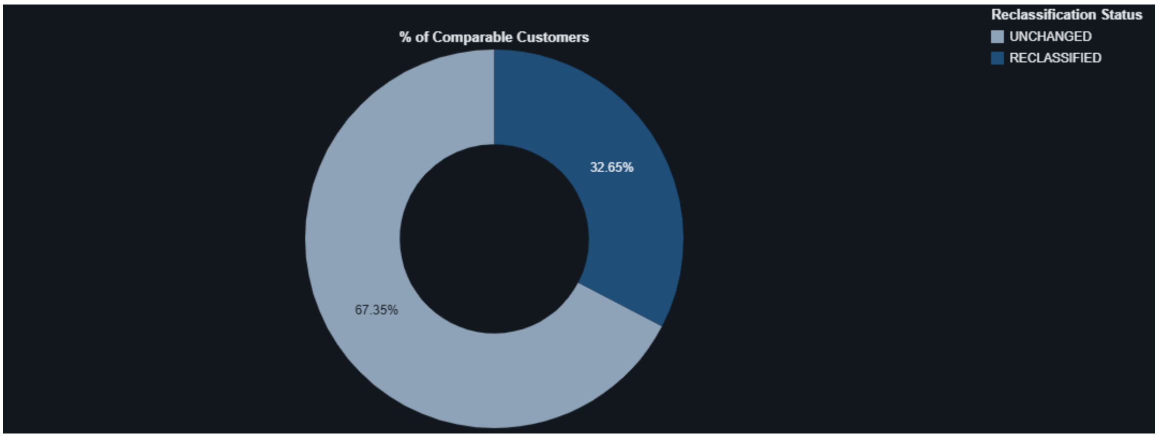

Figure 7.1: Personalization Reclassifies a Significant Share of Customers

As illustrated in Figure 7.1, approximately one-third of comparable customers are reclassified once customer-level behavioral interpretation is introduced. The detailed population breakdown is presented in Table 7.2.

Table 7.2: Personalization Introduces Material Reclassification at Population Level

Reclassification Status	Customers (M)	% of Comparable Customers
Unchanged	3.3	67.35%
Reclassified	1.6	32.65%

Table 7.2 shows that 32.65% of comparable customers are reclassified under the personalized framework, while 67.35% retain the same churn state. This means that personalization affects approximately one in three customers with sufficient behavioral history.

This is a material shift rather than a marginal refinement. At the same time, the majority of customers remain aligned under both approaches, indicating that personalization does not destabilize the segment-level framework. Instead, it selectively refines interpretation where aggregate thresholds are insufficient.

The analysis therefore shifts from whether personalization changes outcomes toward where those changes are concentrated and how they should influence retention prioritization.

7.3 Reclassification Impact Across Behavioral Segments

The population-level result does not show where reclassification is concentrated. In practice, personalization does not affect all customer groups equally, and its operational value depends on which segments experience the most meaningful changes in risk interpretation.

The analysis therefore evaluates reclassification across four behavioral groups: FREE Only, OTC Only, RX Only, and RX + OTC.

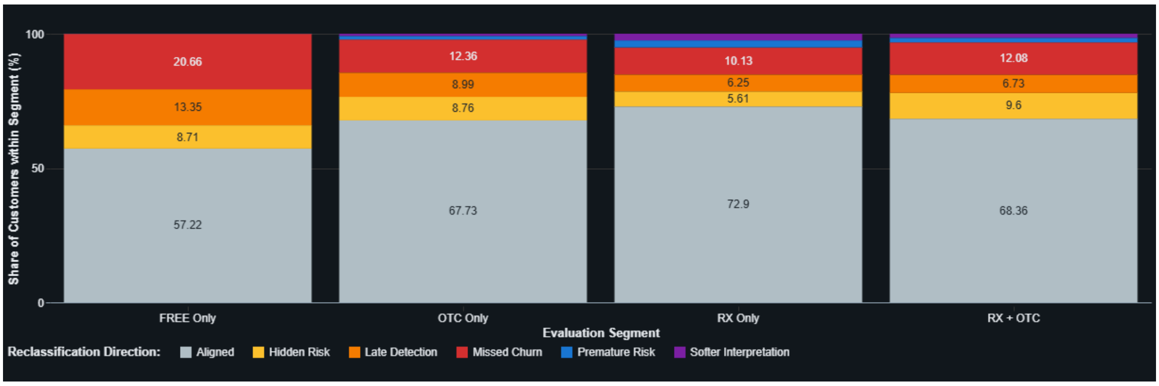

Figure 7.2: Reclassification Impact Varies Substantially Across Behavioral Segments

Table 7.3: Segment-Level Reclassification Impact and Business Interpretation

Segment	Reclassified %	Main Pattern	Business Priority
FREE Only	42.78%	High missed churn / late detection; weak engagement	Low-cost monitoring
OTC Only	32.27%	Mixed hidden risk and late detection patterns	Scalable retention optimization
RX Only	27.10%	Moderate missed churn; relatively stable behavior	Refinement layer
RX + OTC	31.64%	Missed risk in high-value engaged customers	Highest-priority intervention

Figure 7.2 and Table 7.3 show that personalization impact varies both in magnitude and business relevance. FREE Only customers show the highest reclassification rate, but this mainly reflects weak and irregular engagement, making the segment better suited for broad, low-cost monitoring rather than intensive retention investment.

OTC Only customers represent the largest scalable opportunity. Their reclassification patterns are more balanced and behaviorally interpretable, making personalization useful for improving retention prioritization across a large customer base.

The RX groups show a different pattern. RX Only has a lower reclassification rate, suggesting that segment-level interpretation already captures much of its inactivity structure. RX + OTC, however, remains strategically important because even moderate reclassification levels can carry higher business relevance due to stronger engagement and commercial value.

These findings show that personalization does not create uniform value across the customer base. Its impact depends on the intersection between behavioral structure, segment scale, and business relevance.

7.4 Operational Interpretation of Reclassification Patterns

Reclassification patterns are not only analytical label changes. Each transition between segment-level and personalized states changes how customer risk should be interpreted operationally.

Table 7.4: Business Interpretation of Reclassification Directions

Reclassification Direction	Operational Meaning	Business Implication
ACTIVE → AT RISK	Hidden early risk	Earlier intervention opportunity
ACTIVE → CHURNED	Missed churn signal	Absence of timely retention action
AT RISK → CHURNED	Late detection	Reduced recoverability
AT RISK → ACTIVE	Premature risk escalation	Unnecessary intervention pressure
CHURNED → AT RISK / ACTIVE	Softer personalized interpretation	Improved targeting precision

As summarized in Table 7.4, the most important transitions are those where risk is underestimated under segment-level logic. ACTIVE → AT RISK reveals hidden early risk, ACTIVE → CHURNED indicates missed churn signals, and AT RISK → CHURNED reflects late detection. These transitions point to cases where intervention timing may need to move earlier.

Reverse transitions indicate cases where segment-level logic may overestimate risk relative to the customer's own rhythm. Operationally, this can reduce unnecessary intervention pressure and improve targeting precision.

Overall, personalization improves the proportionality of business response by distinguishing missed risk from over-escalated risk.

7.5 Reliability-Aware Reclassification

Reclassification direction alone does not determine intervention priority. The reliability of the signal also depends on how stable the customer's historical behavior has been. For this reason, reclassification patterns are evaluated together with behavioral consistency.

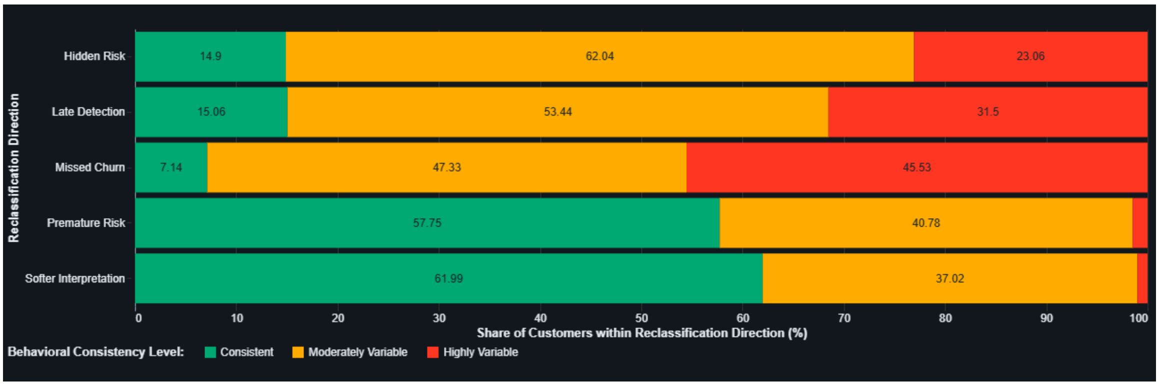

Figure 7.3: Reclassification Signals Differ in Reliability Across Behavioral Profiles

Figure 7.3 shows that elevated-risk reclassifications, particularly missed churn and late detection, are concentrated among moderately variable and highly variable customers. This suggests that not all elevated churn signals should trigger the same intervention intensity, since the reliability of the underlying behavior differs across customers.

By contrast, softer personalized interpretations and premature risk corrections are more concentrated among behaviorally consistent customers. In these cases, personalization reliably reduces overestimated risk because the customer's historical pattern provides a stable reference point.

The reliability layer therefore strengthens prioritization by combining the severity of reclassification with confidence in the underlying behavioral signal.

7.6 Strategic Retention Prioritization Across Behavioral Segments

The operational value of personalization is segment-dependent. Reclassification impact must therefore be interpreted together with behavioral consistency, customer value, and the expected feasibility of intervention. (Rust, Lemon, & Zeithaml, 2004).

Table 7.5: Strategic Retention Prioritization Framework

Segment	Strategic Role	Retention Prioritization Logic
FREE Only	Low-value monitoring population	Broad, low-cost retention; avoid high-intensity intervention.
OTC Only	Scalable retention and conversion opportunity	Prioritize controlled targeting and retention experimentation.
RX Only	Stable high-engagement population	Use personalization as refinement rather than primary trigger.
RX + OTC	Highest-value retention priority	Prioritize early, targeted intervention for missed-risk signals.

Table 7.5 shows that personalization should not be used uniformly across all customer groups. FREE Only customers show high reclassification intensity, but their weak and irregular engagement limits the value of intensive intervention. OTC Only customers represent a scalable opportunity because of their size and interpretable risk patterns.

The strongest strategic priority remains RX + OTC. Even when reclassification rates are not the highest, missed-risk signals in this group carry greater business relevance due to stronger engagement continuity and higher commercial importance.

The main value of personalization is therefore not simply higher churn sensitivity. Its value lies in improving where retention effort is focused, where intervention intensity should be moderated, and where customer-specific risk signals justify earlier action.

7.7 Behavioral Foundations for Predictive Modeling

The evaluation confirms that personalized churn states provide a more behaviorally specific representation of customer inactivity than segment-level thresholds alone. This has direct implications for predictive modeling.

First, personalized churn states create a more informative target structure by reducing noise from overly broad inactivity assumptions. Second, the evaluation identifies important feature families for future prediction, including segment membership, recency, frequency, deviation from expected rhythm, and behavioral consistency.

Predictive modeling therefore builds on the behavioral framework rather than replacing it. The churn states defined and evaluated in Chapters 5 - 7 become the foundation for forecasting future customer-state evolution in Chapter 8.

7.8 From Behavioral Evaluation to Predictive Intelligence

This chapter evaluated the impact of moving from segment-level to personalized churn interpretation. The results showed that personalization reclassifies a material share of customers, but that its value is not evenly distributed across the customer base.

The strongest contribution lies in improving intervention timing and prioritization. Personalization helps reveal hidden early risk, identify late detection, and reduce over-escalation where customer behavior remains aligned with individual expectations.

Having established the behavioral and operational relevance of personalized churn states, the next step is to predict future customer-state evolution. Chapter 8 therefore extends the framework into leakage-safe predictive modeling using behaviorally grounded churn states as the target outcome.

8. Predictive Modeling of Future Customer Churn States Using Behavioral Traits

8.1 Predictive Modeling Objective and Scope

Following the evaluation of personalized churn interpretation in Chapter 7, the analysis now moves from retrospective behavioral assessment to future customer-state prediction. The objective is to examine whether behaviorally grounded churn states can be predicted using leakage-safe transactional features constructed from historical customer activity.

This chapter does not aim to replace the behavioral framework developed throughout the thesis or to optimize a standalone ML benchmark. Instead, predictive modeling is used as an extension of the behavioral decision architecture established across Chapters 3 - 7. The modeling design therefore prioritizes temporal validity, interpretability, scalability, and operational relevance over exhaustive algorithmic experimentation (Sculley, et al., 2015).

The predictive task is formulated as a multiclass classification problem with three future customer states: ACTIVE, AT RISK, and CHURNED. This structure preserves the intervention logic developed earlier in the thesis by distinguishing stable engagement, emerging behavioral deterioration, and prolonged disengagement rather than collapsing all outcomes into a binary churn label.

To prevent information leakage, all predictive features are constructed exclusively from historical behavioral data available within the observation window, while customer states are evaluated in a future prediction horizon. This temporal separation ensures that the model reflects realistic deployment conditions, where future behavior is not available at prediction time.

Table 8.1: Predictive Modeling Scope

Modeling Dimension	Predictive Scope	Operational Objective
Prediction Target	ACTIVE / AT RISK / CHURNED	Enable differentiated retention intervention
Prediction Logic	Behavioral multiclass classification	Detect varying levels of disengagement severity
Feature Foundation	Behavioral transaction traits	Preserve explainability and operational interpretability
Temporal Design	Observation window → future prediction window	Prevent information leakage
Modeling Objective	Early identification of behavioral deterioration	Support proactive retention decision-making

As summarized in Table 8.1, the predictive framework is designed to evaluate whether explainable behavioral traits can support future-state classification within a large-scale deployment (Ribeiro, Singh, & Guestrin, 2016). The reusable Spark ML preprocessing pipeline, predictive dataset construction logic, model-training workflows, and evaluation implementation details are provided in Appendix D.

8.2 Temporal Modeling Architecture

The predictive framework uses a temporally separated modeling architecture. Behavioral features are constructed from an 18-month observation window, while future customer states are evaluated within a subsequent 6-month prediction horizon. This design preserves a clear separation between information available at prediction time and outcomes observed later.

Figure 8.1 illustrates the temporal modeling architecture. Historical behavioral traits, including recency, ordering frequency, inter-order dynamics, behavioral consistency, transaction value, and segment-level activity patterns, are extracted from the observation window and used to predict ACTIVE, AT RISK, or CHURNED states in the future prediction window.

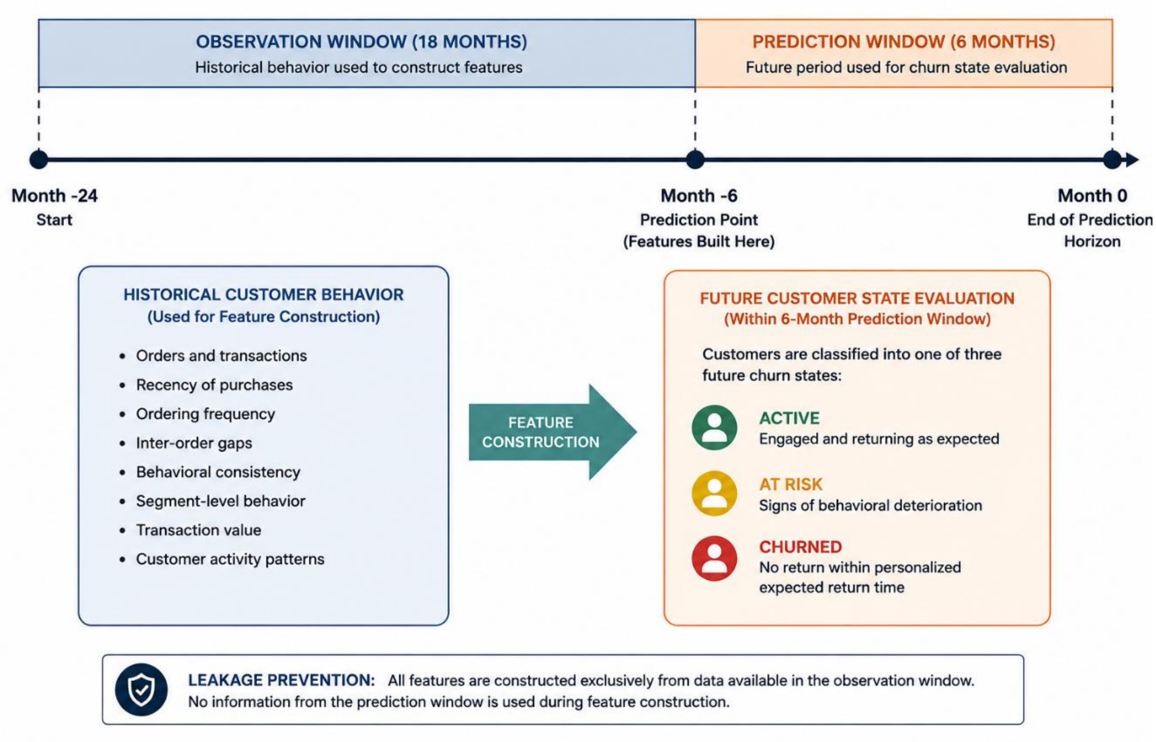

Figure 8.1: Temporal Modeling Architecture for Leakage-Safe Churn Prediction

The 18-month observation window was selected to provide sufficient behavioral depth for estimating customer-level patterns across recurring RX activity, irregular OTC engagement, and variable return rhythms. Shorter windows would increase behavioral volatility, while substantially longer windows could dilute recent behavioral changes.

The 6-month prediction horizon balances operational relevance with predictive stability. Shorter horizons may capture temporary inactivity fluctuations, while longer horizons can reduce the usefulness of retention-intervention timing. The selected horizon therefore provides a realistic forward-looking window for proactive customer-retention analysis.

The use of a single observation–prediction split is an intentional scope decision. It provides a clear and interpretable temporal separation between feature construction and future-state evaluation, which is aligned with the applied enterprise scope of the thesis. Rolling-window validation could further assess temporal stability across multiple cutoff points, but is treated as a future extension rather than part of the current modeling scope.

Future customer states are derived from forward-looking inactivity behavior measured relative to each customer’s last observed order within the observation window. No behavioral information from the prediction window is used during feature construction, ensuring that predictive performance reflects genuine forward-looking inference rather than retrospective leakage.

8.3 Behavioral Feature Engineering Framework

The predictive framework uses customer-level behavioral traits derived exclusively from transactional activity observed within the 18-month observation window. No demographic enrichment, externally inferred attributes, or pre-derived business labels are used. This preserves interpretability and ensures that the model remains aligned with the behavioral scope of the thesis.

Feature engineering is organized around the behavioral dimensions developed in earlier chapters: recency, frequency, inter-order dynamics, behavioral variability, transaction value, segment context, and reliability indicators. These features are designed to help the models distinguish stable engagement, transitional deterioration, and prolonged disengagement.

Segment context is included because product interaction patterns influence expected ordering behavior. Behavioral groups such as FREE Only, OTC Only, RX Only, FREE + OTC, RX + OTC, FREE + RX, and FREE + RX + OTC capture product-level interaction structure, while RX CUSTOMER, OTC CUSTOMER, and FREE Only CUSTOMER provide higher-level customer context.

Reliability-related traits are also included because customers with irregular ordering patterns may show inactivity that is less straightforward to interpret. Gap count, coefficient variation, and behavioral consistency indicators allow the models to account for differences between stable deterioration and naturally volatile purchasing behavior.

Table 8.2 summarizes the primary behavioral feature categories incorporated into the predictive modeling framework.

Table 8.2: Behavioral Feature Categories

Behavioral Dimension	Representative Behavioral Traits	Predictive Purpose
Recency	Days since last order, personalized inactivity deviation	Detect delayed customer return behavior
Frequency	Order count, average ordering cadence	Capture customer engagement intensity
Inter-Order Dynamics	Median inter-order gap, gap progression patterns	Model customer return rhythm
Behavioral Variability	Gap variability, coefficient of variation	Distinguish stable vs irregular purchasing behavior
Transaction Value	Average transaction value, cumulative customer value	Capture commercial engagement magnitude
Segment-Level Behavioral Context	FREE Only, OTC Only, RX Only, FREE + OTC, RX + OTC, FREE + RX, FREE + RX + OTC	Preserve structural behavioral differences
Customer-Level Behavioral Segments	RX CUSTOMER, OTC CUSTOMER, FREE Only CUSTOMER	Provide higher-level behavioral context
Reliability Indicators	Gap count, behavioral consistency measures	Support confidence-aware interpretation

The overall behavioral feature architecture is illustrated in Figure 8.2.

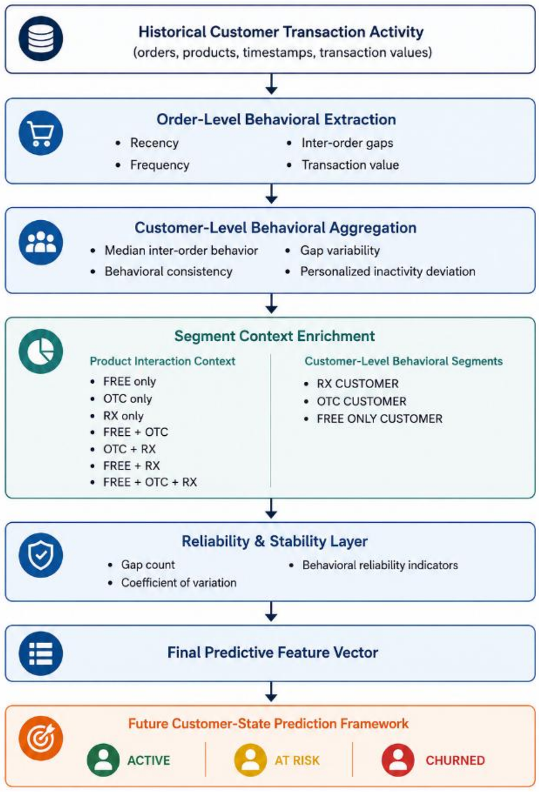

Figure 8.2: Behavioral Feature Engineering Architecture

Figure 8.2 illustrates how raw transactional activity is transformed into a final predictive feature vector. The process combines order-level behavioral extraction, customer-level aggregation, segment-context enrichment, and reliability indicators before constructing the model-ready feature base.

This feature architecture is intentionally constrained. More complex representation-learning approaches may improve raw predictive performance, but the selected behavioral-trait design preserves transparency, business interpretability, and governance suitability within an organizational retention setting (Molnar, 2022).

8.4 Construction of the Predictive Modeling Dataset

The predictive modeling dataset integrates the behavioral feature base with future personalized customer-state outcomes at the ordering-customer level. The dataset is structured to remain stable and scalable while maintaining the temporal separation necessary to avoid data leakage (Kaufman, Rosset, Perlich, & Stitelman, 2012).

All features are derived exclusively from the observation window, while the prediction target is assigned from the subsequent prediction horizon. The final target consists of the three future behavioral states: ACTIVE, AT RISK, and CHURNED.

Table 8.3: Predictive Target Class Distribution

Customer State	Customers	% of Dataset
ACTIVE	911,230	16.22%
AT RISK	689,855	12.28%
CHURNED	4,026,575	71.50%

Table 8.3 shows substantial class imbalance, with CHURNED representing 71.50% of the modeling population. This imbalance is expected given the inactivity-heavy structure of the customer base and reinforces the need for per-class evaluation rather than reliance on accuracy alone. The AT RISK class represents a transitional behavioral condition and is therefore expected to be more difficult to classify consistently.

Before model training, the dataset was split into training and testing subsets using an 80/20 split with a fixed random seed for reproducibility. The test set was used only for evaluation on unseen customers.

A Spark ML preprocessing pipeline was then used to prepare the dataset for distributed model training (Zaharia, et al., 2016). Categorical behavioral attributes were transformed through indexing and one-hot encoding, missing numerical values were handled through imputation, and all processed variables were assembled into a unified feature vector.

The preprocessing design was intentionally conservative and preserves interpretability, scalability, and reproducibility within a large-scale deployment.

Table 8.4 summarizes the main preparation steps used to transform the scoped behavioral dataset into a model-ready Spark ML input structure.

Table 8.4: Predictive Modeling Dataset Preparation Workflow

Step	Purpose	Operational Role
Train/Test Split	Separate training and evaluation data	Prevent evaluation bias and ensure generalizable performance
Categorical Encoding	Convert behavioral categories into machine-readable representations	Enable model processing of segment-related features
Missing Value Imputation	Handle incomplete numerical observations	Preserve dataset stability and customer coverage
Feature Vector Assembly	Consolidate all behavioral inputs into a unified structure	Standardize model-ready input generation

8.5 Predictive Modeling Strategy

The modeling strategy was structured to compare representative ML approaches using a consistent behavioral feature and target framework. Rather than pursuing extensive hyperparameter tuning, the focus was placed on evaluating how different model families balance interpretability, non-linear learning capacity, and predictive adaptability.

The implementation followed a SQL-first behavioral engineering approach combined with distributed Spark-based model training. Initial exploratory processing was performed using Databricks Serverless compute, while the final modeling workflow was executed on a dedicated Databricks ML compute environment running Databricks Runtime 17.3 LTS ML. This supported stable Spark ML and XGBoost execution, scalable training, and persistence of prediction and evaluation artifacts.

Three models were evaluated: Logistic Regression, Random Forest, and XGBoost. The models were selected to represent a progression from interpretable linear baseline to non-linear ensemble learning and boosted tree-based classification. No explicit resampling or synthetic balancing techniques were applied, as the objective was to evaluate predictive behavior under the natural class distribution observed in the behavioral population.

Logistic Regression was selected as an interpretable baseline. Its transparent structure makes it useful for evaluating whether the engineered behavioral features carry predictive signal under a simple linear decision framework (Hosmer, Lemeshow, & Sturdivant, 2013).

Random Forest was selected as a non-linear ensemble baseline capable of capturing interactions between recency, ordering consistency, purchasing intensity, and product context while remaining relatively robust and interpretable through feature-importance analysis (Breiman, 2001).

XGBoost was selected as the most advanced model in the comparison because of its strong performance on structured tabular data and its ability to capture complex behavioral boundaries through gradient boosting (Chen & Guestrin, 2016).

Model configuration was kept controlled and comparable across algorithms. The idea was to evaluate whether behaviorally grounded features support future churn-state prediction across increasingly flexible model families.

Table 8.5 summarizes the modeling role, strengths, limitations, and relevance of each selected model. This comparison supports both predictive-performance evaluation and interpretation of the tradeoff between model complexity and operational transparency.

Table 8.5: Predictive Model Comparison Strategy

Model	Modeling Role	Main Strength	Main Limitation	Enterprise Relevance
Logistic Regression	Interpretable behavioral baseline	High explainability and operational transparency	Limited ability to capture non-linear interactions	Strong suitability for governance-oriented environments
Random Forest	Non-linear ensemble baseline	Robust handling of behavioral interaction patterns	Lower interpretability compared to linear models	Suitable for scalable behavioral classification
XGBoost	Advanced boosted ensemble model	Strong predictive capability for complex behavioral structures	Higher model complexity and lower transparency	High enterprise applicability for production-scale predictive analytics

8.6 Predictive Performance Evaluation

This section evaluates predictive performance across ACTIVE, AT RISK, and CHURNED states. Because the target distribution is imbalanced, evaluation extends beyond accuracy and includes weighted metrics, macro-level performance, per-class behavior, confusion patterns, and AT RISK detection capability.

The evaluation focuses on four complementary perspectives:

1. Overall predictive performance
2. Per-class predictive balance
3. Misclassification structure
4. AT RISK detection capability

8.6.1 Overall Predictive Performance

Table 8.6 presents the aggregated predictive performance across the evaluated models. The comparison includes overall classification accuracy together with weighted and macro-level F1 indicators in order to evaluate not only aggregate predictive performance, but also the consistency of classification behavior across all customer states.

Table 8.6: Overall Predictive Performance

Model	Accuracy	Weighted Precision	Weighted Recall	Weighted F1	Macro F1
XGBoost	0.8117	0.7907	0.8117	0.7861	0.6185
Random Forest	0.7996	0.7829	0.7996	0.7589	0.5697
Logistic Regression	0.7987	0.7721	0.7988	0.7602	0.5588

Accuracy alone would provide an incomplete view of model quality because the dataset is dominated by the CHURNED class. Weighted metrics capture overall performance while accounting for class support, whereas Macro F1 provides a stricter view of balance across all states.

All three models achieved strong overall performance, indicating that the behavioral feature architecture contains meaningful predictive signal. XGBoost achieved the highest Accuracy, Weighted F1, and Macro F1, suggesting the strongest overall ability to separate future customer states.

The performance differences between models are meaningful but not extreme. This suggests that predictive quality is driven not only by model complexity, but also by the behavioral relevance and temporal consistency of the engineered features.

8.6.2 Per-Class Predictive Performance

Aggregate metrics hide important differences between customer states. Table 8.7 therefore reports Precision, Recall, and F1 Score for ACTIVE, AT RISK, and CHURNED across all models.

Table 8.7: Per-Class Predictive Performance by Model

Customer State	Metric	Logistic Regression	Random Forest	XGBoost
ACTIVE	Precision	0.6356	**0.7865**	0.6750
	Recall	**0.5789**	0.4259	0.5618
	F1 Score	0.6059	0.5526	**0.6132**
AT RISK	Precision	0.5980	**0.6384**	0.6322
	Recall	0.1027	0.1684	**0.2334**
	F1 Score	0.1754	0.2665	**0.3409**
CHURNED	Precision	0.8328	0.8068	**0.8440**

	Recall	0.9677	**0.9922**	0.9672
	F1 Score	0.8952	0.8899	**0.9014**

F1 Score was selected as the primary visualization metric because it balances precision and recall while providing a more representative performance signal under multiclass imbalance conditions. Figure 8.3 visualizes the F1-score distribution across the three behavioral states for all evaluated models.

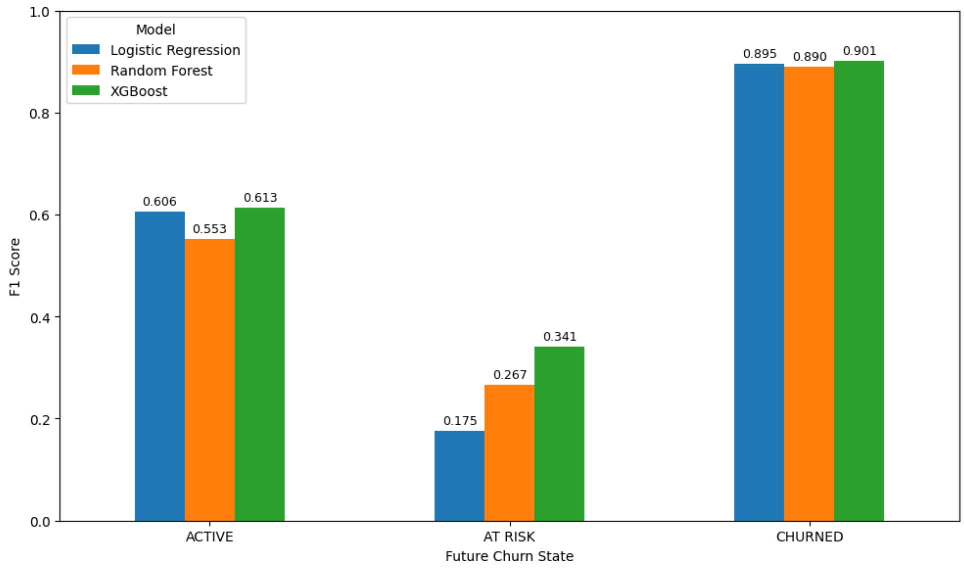

Figure 8.3: Per-Class F1 Score Comparison Across Customer States

The per-class results show that ACTIVE and CHURNED are predicted more reliably than AT RISK across all models. This pattern is behaviorally expected. ACTIVE customers typically show recent engagement, while CHURNED customers exhibit prolonged inactivity and clearer disengagement signals.

AT RISK is expected to be the most difficult state to predict because it represents a transitional behavioral condition rather than a stable endpoint. Unlike ACTIVE and CHURNED, which are more clearly separated by recent engagement or prolonged deviation, AT RISK sits between normal delay and material disengagement.

Logistic Regression performs reasonably for ACTIVE and CHURNED, but struggles with AT RISK recall, suggesting that early deterioration is difficult to capture through linear decision boundaries. XGBoost achieves the strongest AT RISK F1 Score and recall, indicating better sensitivity to gradual behavioral deterioration, although the class remains challenging overall.

8.6.3 Misclassification Structure Analysis

To examine model behavior beyond aggregate metrics, the XGBoost confusion matrix was analyzed. This view shows where the best-performing model separates customer states effectively and where classification errors are concentrated.

Figure 8.4 presents the confusion matrix generated by the XGBoost multiclass classification model.

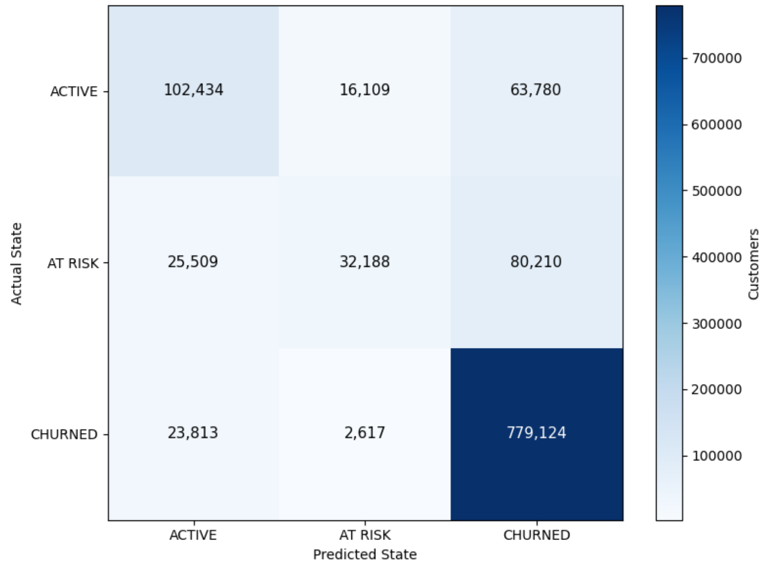

Figure 8.4: XGBoost Confusion Matrix for Multiclass Behavioral Churn Prediction

The confusion matrix confirms strong classification performance for CHURNED customers, where most observations are correctly assigned to the CHURNED state. ACTIVE customers are more frequently confused with CHURNED than with AT RISK, indicating that some customers with apparent engagement signals still resemble future disengagement patterns.

The AT RISK class shows the most dispersed error structure. A substantial share of AT RISK customers is predicted as CHURNED, while another portion is predicted as ACTIVE. This reinforces the interpretation that AT RISK represents a transitional and behaviorally ambiguous state rather than a sharply separated class.

8.6.4 AT RISK Detection Capability

Since AT RISK represents the most actionable intervention stage, it is evaluated separately through Precision and Recall. Precision indicates how reliable AT RISK predictions are, while Recall indicates how many actual AT RISK customers are captured by the model. Figure 8.5 compares Precision and Recall performance across the evaluated models specifically for the AT RISK customer state.

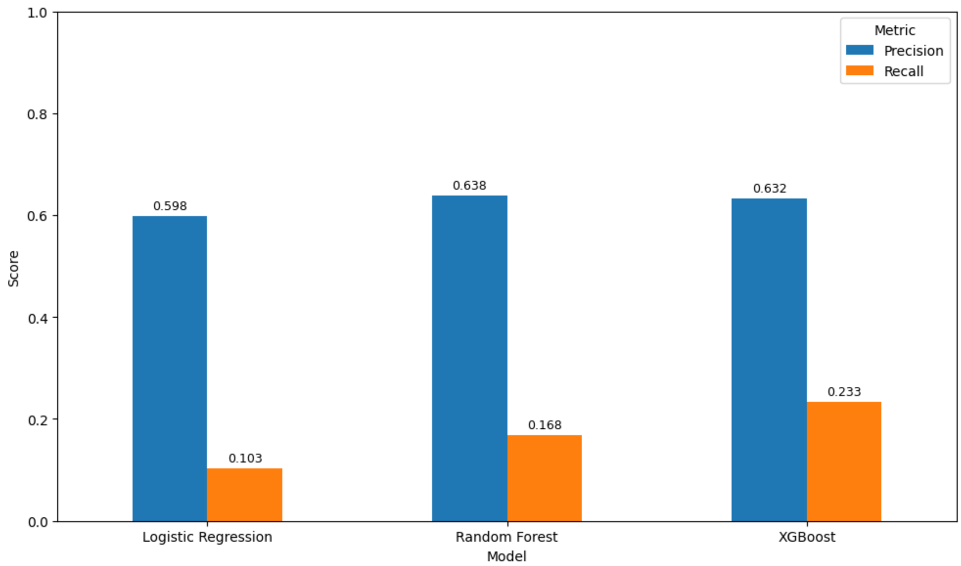

Figure 8.5: Precision–Recall Tradeoff for AT RISK Customer Detection

Figure 8.5 highlights the tradeoff between intervention reliability and early-risk coverage. Higher precision reduces unnecessary targeting, while higher recall improves the ability to identify deteriorating customers before full churn materializes.

XGBoost provides the strongest AT RISK recall among the evaluated models, but absolute recall remains modest. This should not be interpreted only as model weakness. It reflects the structural ambiguity of AT RISK as a transitional state with overlapping behavioral characteristics.

Operationally, AT RISK predictions should therefore be used as prioritization signals rather than deterministic automation triggers. Customers predicted as AT RISK may require differentiated treatment depending on segment context, behavioral consistency, and intervention cost.

Although the AT RISK state remains operationally valuable as an early-warning signal, its comparatively lower recall indicates that transitional behavioral deterioration remains difficult to separate cleanly from both ACTIVE and CHURNED trajectories. This limitation reflects both class overlap and the inherently gradual nature of behavioral disengagement.

8.6.5 Behavioral Feature Importance Analysis

Feature-importance analysis based on the XGBoost model was used to examine which behavioral patterns most strongly influenced prediction outcomes. Instead of concentrating on case-by-case prediction explanations, the analysis focused on understanding the broader decision patterns and behavioral relationships captured by the model.

Figure 8.6 presents the most influential behavioral features identified during the XGBoost modeling process using gain-based feature importance evaluation.

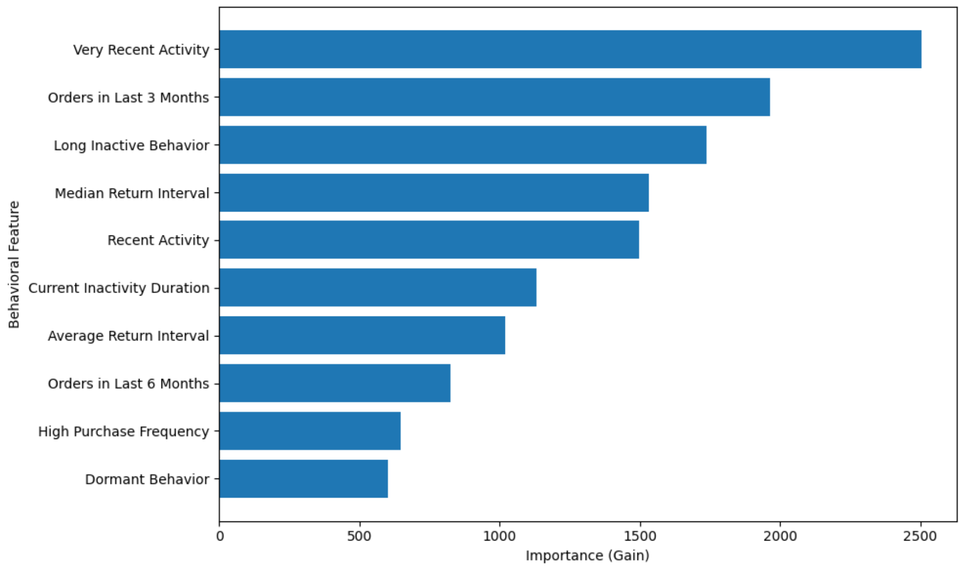

Figure 8.6: Gain-Based Feature Importance in the XGBoost Model

The most influential features relate to recency, recent purchasing activity, inactivity duration, and return-rhythm characteristics. This confirms that the model primarily relies on behavioral signals aligned with the churn framework developed earlier in the thesis.

Features such as median return interval, average return interval, and long inactive behavior further indicate that prediction depends not only on current inactivity, but also on the customer's historical return structure. This is consistent with the personalized churn logic introduced in Chapters 5 and 6.

The analysis remains a global interpretability layer. Local explainability techniques such as SHAP or LIME could provide more detailed explanations for individual predictions and would be valuable in a production validation setting, but they are outside the scope of the current thesis implementation.

8.7 Operational Implications of Predictive Churn Intelligence

The predictive results show that customer-state prediction depends on both model capability and the behavioral separability of the target states. CHURNED is the most reliably predicted class because prolonged inactivity and strong deviation from expected return behavior create clearer predictive boundaries. ACTIVE also shows stronger separability due to recent engagement signals.

AT RISK remains the most difficult class because it represents a transitional condition between normal inactivity and full disengagement. This has important operational implications: AT RISK predictions should support prioritization and monitoring rather than automatic intervention without additional business context.

The results also confirm that personalized churn interpretation improves the behavioral quality of the prediction target. By defining future states relative to customer-level return expectations, the framework reduces label noise caused by applying uniform inactivity assumptions across structurally different purchasing rhythms.

From a business perspective, predictive performance should therefore be interpreted together with behavioral reliability, segment context, and intervention cost. The model does not replace business decision logic; it provides a scalable signal that can support more targeted retention prioritization.

8.8 From Predictive Modeling to Operational Decision Systems

Chapter 8 extended the behavioral churn framework into leakage-safe predictive customer-state evaluation. The results show that behaviorally grounded features can support multiclass prediction of ACTIVE, AT RISK, and CHURNED states, with XGBoost achieving the strongest overall performance among the evaluated models.

The predictive layer should be interpreted as a controlled proof of concept within a behaviorally grounded modeling framework, not as a production-calibrated system. The modeling design intentionally prioritized temporal separation, interpretability, scalability, and operational relevance over exhaustive algorithmic optimization.

Before operational deployment, additional validation would be required. Rolling temporal evaluation could assess performance stability across multiple cutoff points, probability calibration could improve the reliability of predicted risk scores, and local explainability methods such as SHAP or LIME could support individual-level transparency. In addition, live intervention testing would be necessary to confirm whether model-driven prioritization improves retention outcomes in practice.

Overall, the chapter demonstrates that predictive performance is strongest when the target definition, feature engineering, and evaluation logic remain aligned with customer-behavior understanding. Chapter 9 builds on this foundation by examining how predictive churn behavioral insight can be translated into enterprise decision systems, activation workflows, and scalable retention operations.

9. Operationalizing Behavioral Churn Prediction in Enterprise Decision Systems

9.1 Why Prediction Alone Is Not Enough

Chapter 8 established the predictive layer of the proposed framework by transforming behaviorally grounded churn logic into a forward-looking ML approach. The models estimate whether a customer is likely to remain ACTIVE, move into an AT RISK state, or become CHURNED within a future prediction window.

However, prediction alone does not create business value. A predicted churn state or risk probability indicates what may happen, but it does not define what the organization should do next. In operational environments, retention decisions also depend on customer value, segment context, behavioral reliability, intervention timing, operational capacity, and the expected relevance of a possible action. Without this translation layer, predictive models risk remaining disconnected from practical business execution.

Chapter 9 completes the framework by translating predictive behavioral insight into structured decision support. The objective is not to introduce a new model or redefine churn, but to operationalize the behavioral and predictive layers developed throughout the thesis in a realistic, explainable, and scalable way.

In practice, the same predicted state should not automatically lead to the same action. An RX + OTC customer predicted as AT RISK may require earlier attention because the relationship is broader and more commercially relevant. A FREE Only customer with the same predicted state may be better suited for monitoring or low-cost engagement, especially when the behavioral signal is weak or unstable. The predicted state is therefore treated as one input in the decision process, not as a standalone trigger.

9.2 Decision Inputs and Behavioral Intelligence Signals

The proposed decision framework combines multiple layers of behavioral and predictive intelligence developed throughout the thesis. This is important because retention decisions are rarely determined by risk alone. A predicted churn state may indicate future disengagement, but the appropriate response depends on customer context, signal reliability, and expected business impact.

The first input is customer segment context and the approach distinguishes behavioral structures such as FREE Only, OTC Only, RX Only, and RX + OTC. These categories reflect different levels of engagement depth, relationship continuity, and commercial relevance.

The second input is the predicted future churn state. Customers are classified as ACTIVE, AT RISK, or CHURNED within the future prediction window. This output provides the forward-looking behavioral risk signal used for prioritization.

The third input is behavioral reliability. Customers with stable historical ordering behavior generate stronger and more interpretable signals when they deviate from expected patterns, while customers with sparse or highly variable histories require more cautious interpretation.

The fourth input consists of the behavioral traits underlying the prediction itself, including recency, frequency, return rhythm, ordering intensity, consistency, and value-related signals. These traits explain why a customer is predicted to belong to a certain future state and provide interpretability for business users.

Table 9.1: Decision Input Categories

Input Layer	Role in Decision Support	Business Interpretation
Customer Segment	Provides business and engagement context	Indicates whether the prediction concerns a FREE Only, OTC Only, RX Only, or RX + OTC customer relationship
Predicted Future State	Provides the forward-looking churn signal	Indicates whether the customer is expected to be ACTIVE, AT RISK, or CHURNED
Behavioral Reliability	Provides confidence in the signal	Helps determine whether the prediction should trigger intervention, monitoring, or cautious interpretation
Core Behavioral Traits	Provide explanatory behavioral context	Clarify the behavioral evidence behind the prediction through recency, frequency, return rhythm, consistency, and value-related signals

Taken together, these inputs make the decision logic more usable for business teams, because risk is interpreted together with value, reliability, and observable customer behavior.

9.3 Decision Framework Design

The decision framework translates predictive behavioral insight into structured business logic. Its purpose is to support consistent and explainable prioritization rather than automate customer intervention blindly.

The framework evaluates customer signals across three dimensions: predicted risk, business value, and behavioral reliability. Predicted risk reflects the expected future customer state. Business value is represented through segment context and engagement depth. Behavioral reliability reflects how confidently the signal can be interpreted based on historical customer behavior.

This distinction is operationally important because the same predicted state does not necessarily imply the same business response. An RX + OTC customer predicted as AT RISK with a stable historical return pattern represents a stronger and more usable signal than a FREE Only customer with limited engagement history and highly irregular behavior.

In practice, this means that identical churn predictions should not automatically trigger identical campaign actions.

Table 9.2: Example Decision Logic Framework

Segment	Predicted State	Reliability	Decision Priority	Example Action
RX + OTC	AT RISK	Consistent	High	Early retention intervention
OTC Only	AT RISK	Moderate	Medium	Targeted engagement campaign
FREE Only	AT RISK	Variable	Low	Monitoring or low-cost engagement
RX Only	CHURNED	Consistent	Medium	Selective reactivation

Table 9.2 is not intended to prescribe fixed campaign rules. Instead, it demonstrates how the framework prevents uniform intervention logic by interpreting churn predictions within broader behavioral and business context.

The goal is not to maximize intervention volume, but to improve intervention efficiency by prioritizing customers where action is timely, justified, and likely to create measurable business impact.

9.4 Prioritization and Resource Allocation

In real enterprise environments, not every predicted churn signal can or should trigger immediate intervention. Retention activities are constrained by campaign capacity, communication limits, budget, operational resources, and the risk of customer over-contacting (Blattberg, Kim, & Neslin, 2008).

For this reason, the framework prioritizes customers based on the combined interpretation of predicted risk, behavioral reliability, business relevance, and expected intervention impact. Customers with strong behavioral signals, high-value segment profiles, and stable historical behavior should receive greater operational attention than customers with weak engagement or structurally ambiguous behavioral patterns.

Prioritization should therefore be understood not only as a classification problem, but also as a resource-allocation problem. Occasional low-impact targeting inaccuracies may be operationally acceptable, while systematically targeting low-value or unreliable cases may reduce campaign efficiency and increase unnecessary intervention costs.

Table 9.3: Example Prioritization Tiers

Priority Tier	Customer Characteristics	Operational Strategy
Tier 1	High business relevance, strong predicted risk, and reliable behavioral signal	Immediate or early retention intervention
Tier 2	Moderate business relevance or moderate signal confidence	Selective engagement
Tier 3	Low confidence, weak behavioral history, or limited business relevance	Monitoring or low-cost engagement

This structure makes the prioritization logic more realistic, because intervention intensity is linked to both signal confidence and expected business value.

9.5 Operationalization in Business Context

The decision framework only creates business value when its outputs can be translated into operational processes. Rather than proposing a complete technical CRM implementation, this section illustrates how predictive churn intelligence can enable timely, transparent, and structured customer interventions.

In practice, the framework could support trigger-based activation after integration with consent, campaign governance, CRM execution rules, and intervention measurement. A predicted state does not directly determine the final action, but it can initiate a structured business trigger after being interpreted through the decision logic. Customers predicted as AT RISK may become candidates for retention campaigns. Customers predicted as CHURNED may enter selective reactivation flows. Customers predicted as ACTIVE should generally not receive churn-prevention interventions, unless a separate growth or engagement logic applies.

This distinction matters because business teams need more than a risk label. Marketing, CRM, and lifecycle teams need to understand why a customer is prioritized, how urgent the signal is, and what type of response is justified (Payne & Frow, 2005). This supports structured business decision-making rather than autonomous decision replacement.

Table 9.4: Example Trigger-Based Activation Logic

Predicted State	Decision Interpretation	Example Business Trigger	Operational Note
ACTIVE	Customer is expected to remain engaged	No churn intervention	Avoid unnecessary churn-related communication

AT RISK	Customer shows early signs of future disengagement	Retention campaign	Main intervention window where action may still influence behavior
CHURNED	Customer is expected to move beyond normal return behavior	Reactivation flow	Action should be selective and based on value, reliability, and recoverability

The AT RISK state remains the most operationally important category because it represents the stage where intervention may still influence future customer behavior before full disengagement materializes.

A key strength of the framework is that it preserves explainability for business users. Decisions remain interpretable through customer segment, predicted future state, behavioral reliability, and supporting behavioral traits rather than through opaque model output alone.

9.6 Behavioral Expansion Opportunities Beyond Retention

9.6.1 Strategic Extension Beyond Churn Prevention

Although the thesis focuses mainly on churn prevention, the same behavioral foundation can support broader customer lifecycle decisions. Once customer behavior is structured consistently, the same layer can also support retention, prioritization, lifecycle monitoring, and customer value development.

Within the context of Redcare Pharmacy, one relevant extension is the identification of OTC Only customers whose behavior increasingly resembles stronger mixed engagement patterns. These customers may demonstrate high ordering frequency, stable return behavior, low volatility, and recent repeat engagement. Such patterns do not imply medical need or prescription eligibility. Instead, they indicate stronger platform adoption and deeper customer engagement behavior.

This distinction is essential. The framework does not attempt to predict medical conditions, prescription need, or healthcare outcomes (European Commission, 2019). Such an approach would introduce substantial ethical, regulatory, and governance concerns. The extension remains strictly behavioral: it identifies customers who may be suitable for broader engagement journeys based on observed interaction patterns.

9.6.2 Behavioral Interpretation of Expansion Potential

The previous chapters showed that RX + OTC customers represent the strongest engagement benchmark within the thesis scope. Building on this observation, the framework can be extended to identify OTC Only customers who show behavioral characteristics associated with stronger lifecycle potential.

Relevant behavioral indicators may include frequent ordering, stable inter-order timing, recent repeat engagement, limited inactivity gaps, and consistent multi-order behavior. These signals suggest stronger customer commitment to the platform, not medical intent.

Table 9.5: Behavioral Indicators of Expansion Potential

Segment	Behavioral Pattern	Potential Opportunity
OTC Only	High frequency and stable ordering behavior	Educational or engagement journey
OTC Only	Recent repeat engagement	Broader value expansion opportunity
OTC Only	Strong recency with limited inactivity gaps	Long-term lifecycle nurturing
OTC Only	Consistent multi-order engagement	Personalized retention and expansion campaigns
RX + OTC	High-value mixed behavior	Benchmark engagement profile

The same behavioral insight foundation can help the organization understand not only where relationships may deteriorate, but also where stronger customer engagement may be developed.

9.6.3 Strategic Implications for Enterprise AI Adoption

This extensibility increases the long-term practical value of the framework. The approach is not limited to a single churn-prediction use case. Instead, it creates a reusable behavioral intelligence foundation that can gradually evolve across different customer lifecycle applications over time.

Because the framework remains grounded in observable customer behavior, its outputs are easier to understand, validate, govern, and operationalize within real organizational settings. Rather than relying on opaque automation, the architecture supports interpretable and decision-oriented AI integration that can be introduced progressively into existing business processes.

In practice, an organization may initially apply the framework to churn prevention and later extend it toward engagement optimization, lifecycle orchestration, controlled experimentation, and broader personalization initiatives. This creates a more realistic AI adoption path: begin with a focused and explainable use case, establish organizational trust, and gradually expand the behavioral intelligence layer over time.

9.7 System-Level Perspective and Scalability

9.7.1 Enterprise-Level Architectural Perspective

The long-term value of the approach does not depend only on model performance. It also depends on whether the logic can be maintained, explained, extended, and connected to business processes. For this reason, the proposed approach is better understood as a modular decision-support architecture rather than as a standalone ML model.

The architecture separates the system into distinct layers: behavioral data construction, segmentation logic, personalized churn interpretation, predictive modeling, prioritization, and business activation. This separation improves maintainability because each layer can evolve without requiring a full redesign of the framework.

For example, new behavioral features may be introduced without changing the decision logic. New customer segments may be added without rebuilding the predictive pipeline. Intervention strategies may evolve without retraining the model. Predictive models may also be upgraded while preserving the surrounding business interpretation layer.

This modularity represents one of the primary strengths of the proposed architecture.

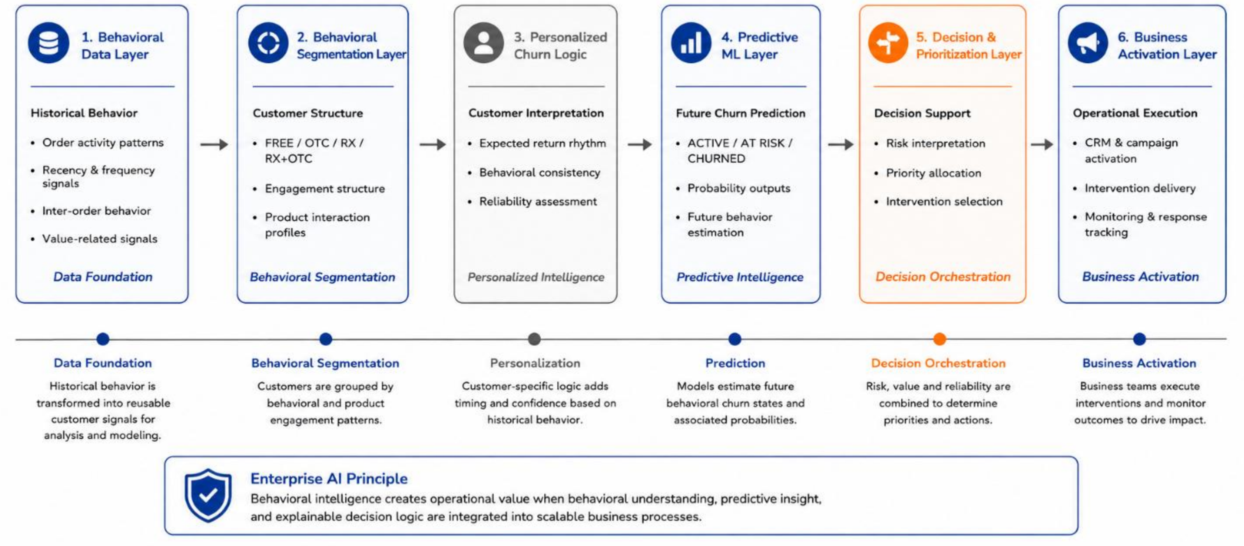

Figure 9.1: Enterprise Behavioral Decision Architecture

Figure 9.1 summarizes the full architecture developed across the thesis. Behavioral data is transformed into segmentation logic, personalized churn interpretation, predictive intelligence, decision prioritization, and business activation. The figure illustrates how behavioral understanding is progressively transformed into predictive behavioral insight and operational decision support.

9.7.2 Scalability and Extensibility

The architecture is scalable because it separates behavioral understanding from predictive execution and operational action.

First, additional behavioral traits can be added as data maturity improves. Second, new customer segments may be introduced as the business evolves. Third, the predictive layer remains replaceable; stronger future models can be introduced without changing the full decision framework. Fourth, intervention logic can evolve independently as CRM strategies, business priorities, or campaign capabilities change.

This separation is particularly important in operational environments, where business decision processes often evolve faster than technical modeling infrastructures. In turn, the framework supports gradual AI maturity rather than requiring a single large-scale transformation.

9.7.3 Explainability and Governance Advantages

In healthcare commerce environments, customer-facing AI systems must remain explainable and operationally transparent (Gartner, 2024). The proposed framework supports explainability across all layers.

Segmentation logic remains interpretable, personalized churn states are grounded in observable customer behavior, predictive outputs are evaluated within business context, and prioritization follows structured decision logic.

This design reduces the risk of opaque model-driven action. Predictions are not treated as instructions. They are interpreted through behavioral evidence before business action is considered. This improves stakeholder trust, governance alignment, auditability, and operational adoption.

9.8 From Predictive Intelligence to Enterprise Decision Support

Chapter 9 completed the transition from predictive modeling to operational decision support. While Chapter 8 demonstrated how historical behavioral traits can predict future customer states, this chapter addressed the next enterprise-critical question: how those predictions can be translated into structured, explainable, and actionable business decisions.

Customer segment, predicted future state, behavioral reliability, and core behavioral traits are combined to interpret churn risk in context. This prevents all predictions from being treated as equally usable and allows the business response to reflect both risk severity and signal quality.

A central contribution of the chapter is the distinction between intervention volume and intervention efficiency. In a real business setting, retention resources, campaign capacity, communication frequency, and operational attention are limited. The proposed framework therefore prioritizes action where the combination of risk, value, reliability, and behavioral evidence suggests the highest expected impact.

The chapter also showed that the same behavioral intelligence foundation can support opportunities beyond churn prevention. Within clear governance boundaries, behavioral signals can help identify customers suitable for broader engagement and lifecycle development. This extension remains strictly behavior-based and does not infer medical need, prescription eligibility, or clinical intent.

Finally, the system-level perspective demonstrated that the proposed approach is not a standalone churn model, but a modular decision-support architecture. Behavioral data, segmentation, personalized churn logic, predictive modeling, prioritization, and activation are separated into interpretable layers that can evolve independently.

Overall, Chapter 9 shows how the predictive layer can be used in practice without treating the model output as an automatic business decision. Customer behavior, segment context, personalized churn logic, predictive signals, and prioritization rules are connected into a decision-support structure that remains explainable and operationally usable. Rather than treating churn prediction as an isolated modeling task, the thesis demonstrates how behavioral intelligence can be translated into structured and operationally usable customer lifecycle decision support.

10. Conclusion

10.1 Research Objective and Thesis Scope

Customer churn prediction is a common application of ML in customer analytics. However, many approaches remain too model-centric: they focus on predictive performance without giving enough attention to how churn is defined, interpreted, and used in business decisions. In practice, value is created not by prediction alone, but by turning predictive signals into reliable and actionable decision support.

This thesis addressed that challenge by developing an explainable behavioral AI framework for customer churn interpretation and operational decision support within the context of Redcare Pharmacy. Rather than treating churn prediction as an isolated modeling task, the work focused on the broader behavioral lifecycle of customer engagement.

The proposed framework integrates behavioral segmentation, personalized expected-return baselines, reliability-aware churn interpretation, predictive modeling within a distributed environment, and operational decision-support logic into a unified architecture. Throughout the thesis, emphasis was placed on explainability, enterprise realism, and practical applicability, aiming to demonstrate how behavioral data can be transformed into actionable decision support.

More broadly, the thesis follows an applied enterprise-AI research orientation emphasizing operational realism, explainability, scalability, governance awareness, and decision-support usability over purely theoretical optimization or maximum predictive complexity. The objective was not to design an isolated high-performance predictive model, but to develop a behavior-aware framework capable of supporting realistic enterprise decision environments.

10.2 Summary of Thesis Contributions

The contribution of this thesis lies in connecting behavioral churn interpretation, leakage-safe predictive modeling, and operational decision support within one coherent framework.

From a behavioral analytics perspective, the work demonstrated that customer churn cannot be interpreted as a universal phenomenon with identical meaning across all customer groups. Customer behavior differs significantly according to engagement structure, product interaction patterns, ordering rhythm, and behavioral consistency. The thesis therefore introduced a layered behavioral interpretation framework that progressively moved from population-level behavioral analysis toward segment-aware and personalized churn interpretation. This included the development of behavioral segmentation logic, personalized expected-return baselines, behavioral reliability assessment, and customer-level churn contextualization.

From a predictive modeling perspective, the thesis developed a scalable and leakage-safe ML framework using Spark ML within a distributed enterprise environment. Historical customer behavior was transformed into reusable behavioral features covering recency, frequency, return dynamics, behavioral consistency, engagement structure, and value-related customer signals. The predictive framework supported multiclass future-state prediction through the classification of customers into ACTIVE, AT RISK, and CHURNED behavioral states.

Importantly, the modeling architecture emphasized explainability, temporal integrity, and operational applicability rather than maximizing predictive complexity alone.

Beyond prediction itself, the thesis contributed a structured operational decision-support layer that translated predictive outputs into business-relevant prioritization logic. Instead of treating all predicted risk equally, the framework integrated predicted future state, business value context, behavioral reliability, and behavioral traits into explainable intervention prioritization. This enabled the transition from predictive intelligence toward operational decision orchestration within realistic constraints.

Finally, the thesis extended the framework beyond defensive churn prevention by introducing the concept of behavioral expansion potential. This demonstrated how the same behavioral insight foundation can support not only retention-oriented use cases, but also broader customer engagement and lifecycle development opportunities. Importantly, this extension remained governance-aware and behavior-centered without attempting to infer medical need or prescription eligibility.

10.3 Enterprise Implications and Operational Relevance

One of the central goals of this thesis was to maintain enterprise realism throughout the analytical and modeling process. In real organizational environments, predictive systems operate within practical constraints related to explainability, operational scalability, governance requirements, intervention capacity, and business adoption. As a result, the value of AI systems depends not only on predictive performance, but also on whether their outputs can be trusted, interpreted, prioritized, and integrated into operational workflows.

The proposed framework addresses these requirements through modular and interpretable architectural separation. Behavioral understanding, segmentation, personalized churn interpretation, predictive modeling, and decision orchestration remain logically distinct while still functioning as part of a connected decision-support architecture. This modular structure improves maintainability, scalability, organizational transparency, and long-term extensibility without requiring redesign of the overall framework.

The thesis also demonstrated the importance of explainability in operational AI systems. Business teams require understandable reasoning behind customer prioritization decisions, particularly when interventions involve customer communication, campaign targeting, or retention resource allocation (Payne & Frow, 2005). The approach therefore avoids black-box operational logic and instead emphasizes interpretable behavioral signals, contextualized prediction, and transparent prioritization criteria.

Another important practical implication concerns intervention efficiency. Enterprise organizations cannot realistically intervene on all customers simultaneously. Budget limitations, communication fatigue, campaign capacity, and operational prioritization constraints require more selective and behavior-aware decision support. The proposed framework therefore prioritizes expected impact rather than intervention volume, enabling organizations to focus retention efforts where behavioral evidence, business value, and predicted risk suggest the strongest operational opportunity.

Overall, the framework shows how predictive modeling can become more useful when it is embedded in explainable behavioral logic and realistic operational decision rules.

10.4 Research Boundaries and Operational Limitations

While the proposed framework demonstrated strong behavioral interpretability and operational applicability, several methodological and operational boundaries intentionally shaped the scope of the research. These boundaries reflect conscious trade-offs between predictive complexity, explainability, enterprise realism, governance considerations, and practical operational usability.

First, the framework intentionally relied exclusively on observable behavioral and transactional customer signals. Demographic information, clinical context, and external enrichment data were deliberately excluded in order to preserve a clean, explainable, and governance-aware behavioral modeling scope. This decision was partially influenced by enterprise data-governance realities and partially by the core objective of the research itself: to evaluate how much predictive and interpretive intelligence could emerge directly from customer behavior patterns. While this approach strengthened interpretability and behavioral transparency, it also reduced the range of potentially available explanatory variables (European Commission, 2019).

In addition, the thesis intentionally prioritized practical behavioral relevance and operational interpretability over formal statistical significance testing. Given the scale of the dataset, even relatively small behavioral differences would likely achieve statistical significance without necessarily contributing meaningful operational value. As a result, the analysis focused primarily on large-scale behavioral structure, segmentation logic, and enterprise decision applicability rather than inferential statistical optimization.

Second, the proposed framework focused on behavioral prediction and decision support rather than causal inference. The models estimate the likelihood of future churn-related behavior, but they do not establish causal explanations for why individual customers disengage. As a result, the framework supports prioritization and intervention guidance, but not causal attribution. Although survival-analysis approaches could provide probabilistic time-to-event estimation, the thesis intentionally prioritized interpretable behavioral segmentation and operationally explainable inactivity interpretation over statistically optimized hazard modeling.

Third, the thesis did not evaluate real intervention outcomes within a live production environment. Although the framework demonstrates operational applicability conceptually, no real-world retention campaigns or intervention experiments were executed within the scope of the study. Consequently, the actual business uplift generated by the framework was not empirically validated through production deployment or controlled experimentation.

Another important boundary concerns operational deployment maturity. The framework was designed around structured historical behavioral windows and periodic predictive execution rather than continuous real-time event orchestration. While this design remains appropriate for the current enterprise scope and organizational maturity level, future production environments could extend the framework toward near-real-time behavioral scoring and adaptive intervention triggering capabilities.

The predictive target preserves the personalized deviation-ratio logic by comparing future inactivity directly with each customer’s expected return rhythm. No minimum baseline floor was applied to expected_return_days, because the framework intentionally defines deviation as inactivity relative to the customer’s own observed return behavior. In a production deployment, additional calibration such as minimum baseline floors or score capping could be considered to improve operational score stability.

The predictive evaluation is based on a temporally constructed modeling dataset combined with a stratified train–test split. While this approach preserves temporal integrity within the target construction process, future work should extend the evaluation through rolling-window validation to assess model robustness across different time periods and behavioral conditions. Similarly, future production-oriented deployment could further strengthen operational robustness through probability calibration analysis, adaptive threshold optimization, and cost-sensitive evaluation strategies.

In addition, the AT RISK state proved structurally more difficult to predict than the ACTIVE and CHURNED states. This outcome reflects the inherently transitional nature of behavioral deterioration, where customer disengagement often develops gradually rather than through abrupt binary shifts. Importantly, most predictive confusion occurred between adjacent behavioral states rather than between completely opposite states, supporting the behavioral realism of the proposed multiclass framework.

Finally, the thesis was developed within the business and operational context of the German e-commerce pharmacy market and specifically within the organizational environment of Redcare Pharmacy. Behavioral structures, product interaction patterns, regulatory considerations, and operational priorities may therefore differ across industries or geographic markets.

10.5 Future Evolution Opportunities

The proposed framework creates multiple opportunities for future extension and operational evolution.

One realistic short-term direction involves the integration of intervention outcome feedback into the decision framework. This would allow organizations to evaluate which intervention strategies generate the strongest behavioral response across different customer segments and behavioral profiles. Such feedback loops could improve prioritization quality and support more adaptive customer lifecycle management.

Another important extension involves real-time or near-real-time behavioral scoring architectures. Future systems could continuously update customer behavioral states based on streaming behavioral events, enabling faster operational reaction to behavioral changes and improving intervention timing.

The framework could also evolve through the integration of customer lifetime value estimation and profitability-aware prioritization. While the current framework emphasizes behavioral engagement and value-related behavioral signals, future versions could incorporate more advanced financial optimization logic within the prioritization process.

From a more advanced decision-optimization perspective, future research could explore reinforcement-learning approaches for intervention sequencing and prioritization optimization, as well as causal modeling techniques for understanding intervention effectiveness and customer behavioral response dynamics.

Finally, the proposed architecture could gradually evolve into a broader omnichannel customer lifecycle framework integrating behavioral insight across additional customer interaction channels, engagement touchpoints, and operational systems.

10.6 Final Thesis Conclusion

This thesis demonstrated how customer behavioral data can be transformed from transactional records into explainable behavioral intelligence and, ultimately, into operational decision support. By progressively integrating behavioral understanding, segmentation, personalized churn interpretation, predictive modeling, and prioritization logic, the work established a behavior-aware AI framework designed for realistic enterprise application.

The results show that improving churn prediction alone is not sufficient to create business value. Predictive output needs to be interpreted together with customer context, behavioral reliability, and operational priorities. By connecting behavioral analysis, personalized churn interpretation, predictive modeling, and prioritization logic, the thesis demonstrates how customer lifecycle management can become more accurate, transparent, and actionable.

Overall, the thesis demonstrated that customer churn should not be treated as a static inactivity problem or as an isolated predictive exercise. Instead, churn emerges as a contextual behavioral signal that gains meaning only when interpreted through customer behavior patterns, engagement structure, reliability, and operational business context. By progressively connecting behavioral understanding, personalized interpretation, predictive intelligence, and operational prioritization, the proposed framework transforms churn analytics from a purely predictive exercise into an explainable enterprise decision-support capability.

References

Amazon. (2025, January 1). *Amazon Prime Overview*. Retrieved from https://www.amazon.com/prime

Blattberg, R. C., Kim, B.-D., & Neslin, S. A. (2008). *Database Marketing: Analyzing and Managing Customers.* New York: Springer.

Breiman, L. (2001). Random forests. *Machine Learning, 45*(1), 5-32. doi:10.1023/A:1010933404324

Buckinx, W., & Van den Poel, D. (2005). Customer Base Analysis: Partial Defection of Behaviorally Loyal Clients in a Non-Contractual FMCG Retail Setting. *European Journal of Operational Research, 164*(1), 252–268. doi:10.1016/j.ejor.2003.12.010

Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 785-794). Association for Computing Machinery. doi:10.1145/2939672.2939785

Davenport, T. H., & Ronanki, R. (2018). Artificial Intelligence for the Real World. *Harvard Business Review, 96*(1), 108-116.

dm-drogerie markt. (2025, January 1). *PAYBACK Loyalty Program and Customer Engagement Services*. Retrieved from https://www.dm.de

DocMorris AG. (2025, January 1). *Digital Healthcare and Customer Services Overview*. Retrieved from https://www.docmorris.com

Doshi-Velez, F., & Kim, B. (2017). *Towards a Rigorous Science of Interpretable Machine Learning.* arXiv. Retrieved 2026, from https://arxiv.org/abs/1702.08608

European Commission. (2019). *Ethics Guidelines for Trustworthy AI.* European Commission. Retrieved 2026, from https://digital-strategy.ec.europa.eu/en/library/ethics-guidelines-trustworthy-ai

Fader, P. S., Hardie, B. G., & Lee, K. L. (2005). Counting your customers the easy way: An alternative to the Pareto/NBD model. *Marketing Science, 24*(2), 275-284. doi:10.1287/mksc.1040.0098

Gartner. (2024). *Enhance AI Decision Making Through Enterprise Governance of AI*. Retrieved 2026, from https://www.gartner.com/en/documents/5813315

Gupta, S., & Lehmann, D. R. (2003). Customers as Assets. *Journal of Interactive Marketing, 17*(1), 9–24. doi:10.1002/dir.10045

Hadden, J., Tiwari, A., Roy, R., & Ruta, D. (2007). Computer assisted customer churn management: State-of-the-art and future trends. *Computers & Operations Research, 34*(10), 2902-2917. doi:10.1016/j.cor.2005.11.007

Hosmer, D. W., Lemeshow, S., & Sturdivant, R. X. (2013). *Applied Logistic Regression.* (3rd, Ed.) Hoboken, NJ: Wiley.

Kaufman, S., Rosset, S., Perlich, C., & Stitelman, O. (2012). Leakage in data mining: Formulation, detection, and avoidance. *ACM Transactions on Knowledge Discovery from Data, 6*(4), 1-21. doi:10.1145/2382577.2382579

Kumar, V., & Reinartz, W. (2018). *Customer Relationship Management* (3rd ed.). Springer.

Molnar, C. (2022). *Interpretable Machine Learning.* (2nd, Ed.) Munich: Christoph Molnar. Retrieved 2026, from https://christophm.github.io/interpretable-ml-book/

Payne, A., & Frow, P. (2005). A Strategic Framework for Customer Relationship Management. *Journal of Marketing, 69*(4), 167–176. doi:10.1509/jmkg.2005.69.4.167

RedCare Pharmacy. (2025). *Internal Global Sales Data Model Documentation and Field Definitions*. Retrieved from https://www.redcare-pharmacy.com

Reichheld, F. F., & Sasser, W. E. (1990). Zero defections: Quality comes to services. *Harvard Business Review, 68*(5), 105-111.

Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). "Why Should I Trust You?": Explaining the Predictions of Any Classifier. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 1135-1144). San Francisco: ACM. doi:10.1145/2939672.2939778

Rust, R. T., Lemon, K. N., & Zeithaml, V. A. (2004). Return on Marketing: Using Customer Equity to Focus Marketing Strategy. *Journal of Marketing, 68*(1), 109–127. doi:10.1509/jmkg.68.1.109.24030

Schmittlein, D. C., Morrison, D. G., & Colombo, R. (1987). Counting Your Customers: Who Are They and What Will They Do Next? *Management Science, 33*(1), 1–24. doi:10.1287/mnsc.33.1.1

Sculley, D., Holt, G., Golovin, D., Davydov, E., Phillips, T., Ebner, D., . . . Young, M. (2015). Hidden Technical Debt in Machine Learning Systems. *Advances in Neural Information Processing Systems 28 (NeurIPS 2015)* (pp. 2503–2511). Curran Associates.

Tam, K. Y., & Ho, S. Y. (2006). Understanding the Impact of Web Personalization on User Information Processing and Decision Outcomes. *MIS Quarterly, 30*(4), 865–890. doi:10.2307/25148757

Verbeke, W., Dejaeger, K., Martens, D., Hur, J., & Baesens, B. (2012). New insights into churn prediction in the telecommunication sector: A profit driven data mining approach. *European Journal of Operational Research, 218*(1), 211-229. doi:10.1016/j.ejor.2011.09.031

Wedel, M., & Kamakura, W. A. (2000). *Market Segmentation: Conceptual and Methodological Foundations* (2nd ed.). Boston: Springer.

Zaharia, M., Xin, R. S., Wendell, P., Das, T., Armbrust, M., Dave, A., . . . Stoica, I. (2016). Apache Spark: A Unified Engine for Big Data Processing. *Communications of the ACM, 59*(11), 56-65. doi:10.1145/2934664

Appendices

The appendices contain selected SQL and implementation artifacts supporting the behavioral, predictive, and operational framework presented throughout the thesis. The included implementations were selected to demonstrate the core analytical logic, reusable architectural layers, and business-oriented design decisions underlying the proposed approach.

Appendix A - Behavioral Data Foundation

Purpose

This appendix establishes the governed transactional foundation used throughout the thesis. The following SQL implementation validates the completeness, structural consistency, transactional scope, and behavioral integrity of the final analytical dataset before any segmentation, churn interpretation, personalization, or predictive modeling logic is applied.

Rather than functioning as a predictive or segmentation layer, this step ensures that all downstream behavioral analysis is grounded in stable, operationally meaningful, and behaviorally interpretable customer transactions.

Supports: Chapter 2

Framework Relevance

The dataset validation step confirms that the analytical population used in the thesis is complete, consistently scoped, and suitable for behavioral churn analysis. It verifies the 24-month observation horizon, customer and order coverage, transaction validity, product subtype scope, monetary completeness, and exclusion of structurally irrelevant records.

This step does not create a reusable behavioral view. Its role is to document and validate the final transactional scope before the reusable behavioral layers are constructed in Appendix B - Behavioral Understanding Framework.

Source Foundation

The implementation is based on the enterprise transactional sales table used as the primary behavioral source throughout the thesis:

rdc_udp_globalsales_globalsales_dev.default.fact_sales_trans

The query applies the final thesis scope: German own-stock transactions, completed order lines, FREE / OTC / RX product subtypes, exclusion of bill-of-material components, and non-null transaction values.

Included Behavioral Logic

The validation query reports dataset scale, order and customer coverage, observation-window completeness, average basket structure, customer activity intensity, total transaction value, identifier completeness, and product subtype consistency.

SQL Implementation - Behavioral Dataset Validation Layer

```
WITH base AS (
  SELECT
    f.order_date,
    f.main_order_id_bi,
    f.super_customer_id_order,
    f.item_sub_type_name,
    f.sales_price_net_rep_cur
  FROM rdc_udp_globalsales_globalsales_dev.default.fact_sales_trans f
  WHERE
    -- 24-month observation window
    f.order_date BETWEEN DATE('2023-12-31') AND DATE('2025-12-31')

    -- Business scope
    AND f.sales_domain = 'os'
    AND f.sub_company_id = 'Shop-COM'

    -- Valid transactions only
    AND f.order_line_status = 'Processed'

    -- Exclude BOM components
    AND (f.bom_type_name IS NULL OR f.bom_type_name <> 'Stücklistenposition')

    -- Core product subtypes
    AND f.item_sub_type_name IN ('FREE', 'OTC', 'RX')

    -- Ensure valid monetary values
    AND f.sales_price_net_rep_cur IS NOT NULL
)

SELECT
  -- Dataset size
  COUNT(*) AS total_order_lines,

  -- Structural counts
  COUNT(DISTINCT main_order_id_bi) AS total_orders,
  COUNT(DISTINCT super_customer_id_order) AS total_customers,

  -- Time coverage
  MIN(order_date) AS min_order_date,
  MAX(order_date) AS max_order_date,

  -- Basket structure
  COUNT(*) * 1.0 / COUNT(DISTINCT main_order_id_bi) AS avg_lines_per_order,

  -- Customer activity
  COUNT(DISTINCT main_order_id_bi) * 1.0 / COUNT(DISTINCT super_customer_id_order) AS avg_orders_per_customer,

  -- Revenue overview
  SUM(sales_price_net_rep_cur) AS total_transaction_value,

  -- Data quality checks
  SUM(CASE WHEN main_order_id_bi IS NULL THEN 1 ELSE 0 END) AS null_order_id,
  SUM(CASE WHEN super_customer_id_order IS NULL THEN 1 ELSE 0 END) AS null_customer_id,
  SUM(CASE WHEN item_sub_type_name IS NULL THEN 1 ELSE 0 END) AS null_subtype,
  SUM(CASE WHEN sales_price_net_rep_cur IS NULL THEN 1 ELSE 0 END) AS null_transaction_value,

  -- Subtype sanity check
  SUM(CASE WHEN item_sub_type_name = 'FREE' THEN 1 ELSE 0 END) AS free_lines,
  SUM(CASE WHEN item_sub_type_name = 'OTC' THEN 1 ELSE 0 END) AS otc_lines,
  SUM(CASE WHEN item_sub_type_name = 'RX' THEN 1 ELSE 0 END) AS rx_lines
```

Appendix B - Behavioral Understanding Framework

Purpose

This appendix presents the reusable behavioral SQL views that establish the governed analytical foundation used throughout the behavioral understanding framework of the thesis. These reusable layers reconstruct customer purchasing behavior across order-line, order-level, and temporal interaction dimensions and serve as the core analytical infrastructure supporting the behavioral analyses developed in Chapters 3 and 4.

Rather than functioning as isolated exploratory queries, the following views progressively transform raw transactional activity into structured behavioral representations that can later support segmentation, churn interpretation, personalization, and predictive modeling. The idea is to create reusable and behaviorally interpretable analytical layers capable of consistently reconstructing customer interaction patterns across the entire framework.

Supports: Chapters 3 - 4

B.1 Design of the Reusable Order-Line-Level Behavioral Base

Purpose

This view creates the governed order-line-level behavioral base used throughout the thesis. It preserves the most behaviorally relevant transactional information required to understand customer product interaction while excluding records that do not represent valid customer purchasing behavior.

The resulting structure establishes the lowest governed analytical layer used within the behavioral framework and functions as the primary source for reconstructing product-level customer interaction patterns.

Framework Relevance

The order-line layer preserves the product interaction context required to distinguish FREE, OTC, and RX purchasing behavior. This is necessary because churn is later interpreted at customer level, but customer behavior cannot be understood without first knowing which product categories shaped the interaction.

This view provides the product-level input for order composition analysis, entry product mix analysis, customer segmentation, and downstream behavioral abstraction.

Source Foundation

The view is built directly on the enterprise transactional sales table:

rdc_udp_globalsales_globalsales_dev.default.fact_sales_trans

It applies the scoped 24-month thesis dataset defined in Chapter 2 and validated in Appendix A - Behavioral Data Foundation, including:

Included Behavioral Logic

The view retains order date, business-level order identifier, ordering customer identifier, product subtype, and net reported transaction value. It applies the fixed observation window, business scope, transaction-status filter, BOM exclusion, product subtype restriction, and monetary completeness rule.

SQL Implementation - Reusable Order-Line-Level Behavioral Base

```
CREATE OR REPLACE TEMP VIEW v_churn_base_lines_24m AS

WITH params AS (
   -- Define a fixed reference date so that the analysis window remains
   -- stable and fully reproducible across all downstream steps.
   SELECT DATE('2025-12-31') AS reference_date
)

SELECT
   -- Needed for time-window restriction and later temporal interpretation.
   f.order_date,

   -- Used to reconstruct complete customer orders across order lines.
   f.main_order_id_bi,

   -- This is the primary customer entity used throughout the churn framework.
   f.super_customer_id_order,

   -- Retained to distinguish between FREE, OTC, and RX interaction patterns.
   f.item_sub_type_name,

   -- Net reported transaction value in reporting currency.
   f.sales_price_net_rep_cur

FROM rdc_udp_globalsales_globalsales_dev.default.fact_sales_trans f
CROSS JOIN params p

WHERE
   -- Fixed 24-month behavioral observation window
   f.order_date BETWEEN ADD_MONTHS(p.reference_date, -24) AND p.reference_date

   -- Business scope restriction
   AND f.sales_domain = 'os'
   AND f.sub_company_id = 'Shop-COM'

   -- Transaction validity
   AND f.order_line_status = 'Processed'

   -- Structural exclusion of non-customer decision lines
   -- Exclude BOM component lines because they are system-structural
   -- records and do not reflect a direct, customer-visible purchase decision.
   AND (f.bom_type_name IS NULL OR f.bom_type_name <> 'Stücklistenposition')

   -- Product scope restriction
   AND f.item_sub_type_name IN ('FREE', 'OTC', 'RX')

   -- Monetary completeness
   AND f.sales_price_net_rep_cur IS NOT NULL
```

B.2 Design of the Reusable Order-Level Behavioral Base

Purpose

This view aggregates valid order-line records into completed customer orders. Its purpose is to create one stable and behaviorally interpretable row per completed customer order, enabling behavioral analysis at the correct temporal granularity.

The resulting structure transforms transactional line-level activity into customer purchasing events that can later support frequency analysis, recency evaluation, return-gap reconstruction, and churn interpretation.

Framework Relevance

Customer churn is not evaluated at order-line level. This view aggregates valid order lines into one row per completed customer order, preventing product-line granularity from inflating order counts or distorting frequency and recency metrics.

It provides the order-level base for purchase frequency, recency-frequency analysis, engagement intensity, and return-gap reconstruction.

Source Foundation

The implementation is built directly on:

v_churn_base_lines_24m

All order-level metrics therefore inherit the same transactional scope and filtering rules defined in the order-line base.

Included Behavioral Logic

The view groups records by ordering customer and business-level order identifier. It assigns a stable order date, calculates total order value, counts order lines, and captures the number of distinct product subtypes within each completed order.

SQL Implementation - Reusable Order-Level Behavioral Base

```
CREATE OR REPLACE TEMP VIEW v_churn_orders_24m AS

SELECT
   -- Primary customer entity used throughout the churn framework.
   -- Retained here so customer order sequences can be reconstructed.
   super_customer_id_order,

   -- Business-level order identifier.
   -- Each row in the resulting view represents one distinct completed order.
   main_order_id_bi,

   -- Order date at aggregated order level.
   -- MIN(order_date) is used defensively to ensure one stable order date
   -- in case multiple lines exist within the same order.
   MIN(order_date) AS order_date,

   -- Total net reported transaction value of the order.
   -- This preserves the full economic weight of the completed order
   -- across all included order lines.
   SUM(sales_price_net_rep_cur) AS order_net_revenue,

   -- Number of order lines included in the completed order.
   -- Retained as a lightweight structural descriptor of order complexity.
```

```
    COUNT(*) AS order_lines,

    -- Number of distinct product subtypes included in the order.
    -- Helps preserve minimal structural information about order composition
    -- without returning to the full order-line granularity.
    COUNT(DISTINCT item_sub_type_name) AS distinct_product_types

FROM v_churn_base_lines_24m

GROUP BY
    super_customer_id_order,
    main_order_id_bi
```

B.3 Design of the Reusable Customer Order Gap Base

Purpose

This view reconstructs the chronological order sequence of each customer and calculates the number of days between consecutive completed orders.

The resulting structure transforms completed purchasing activity into measurable temporal return behavior and establishes the core analytical layer used throughout the churn interpretation framework.

Framework Relevance

This view reconstructs customer return behavior by calculating the number of days between consecutive completed orders. These inter-order gaps are later used to analyze return probability, behavioral consistency, empirical churn thresholds, personalized expected return behavior, and predictive features.

Without this layer, inactivity would remain a static recency measure rather than a behaviorally interpretable return signal.

Source Foundation

The implementation builds directly on:

v_churn_orders_24m

As a result, the temporal reconstruction inherits the same governed order-level behavioral structure created from the scoped transactional foundation.

Included Behavioral Logic

The view orders each customer's completed purchases chronologically, identifies the next observed order using a window function, calculates the days between orders, and retains only valid forward-looking order transitions.

SQL Implementation - Reusable Customer Order Gap Base

```
CREATE OR REPLACE TEMP VIEW v_churn_order_gaps_24m AS

WITH ordered_orders AS (
  -- Step 1: Reconstruct the chronological order sequence
  SELECT
    -- Primary customer entity used throughout the churn framework.
    super_customer_id_order,

    -- Business-level order identifier of the current completed order.
    main_order_id_bi,

    -- Date of the current completed order.
    order_date,

    -- Net revenue of the current completed order.
    order_net_revenue,

    -- Date of the next completed order for the same customer.
    LEAD(order_date) OVER (
      PARTITION BY super_customer_id_order
      ORDER BY order_date
    ) AS next_order_date
```

```
    FROM v_churn_orders_24m
)

SELECT
    -- Primary customer entity.
    super_customer_id_order,

    -- Identifier of the current completed order.
    main_order_id_bi,

    -- Date of the current completed order.
    order_date,

    -- Net revenue of the current completed order.
    order_net_revenue,

    -- Date of the next completed order for the same customer.
    next_order_date,

    -- Number of days between the current order and the next order.
    DATEDIFF(next_order_date, order_date) AS days_to_next_order

FROM ordered_orders

WHERE
    -- Keep only valid forward-looking transitions.
    next_order_date IS NOT NULL
```

Appendix C - Churn Interpretation and Personalization Framework

Purpose

The views in this appendix translate the behavioral base into customer-level churn interpretation. They consolidate current inactivity, derive customer-specific return baselines, compare recency with expected return behavior, assign personalized states, and measure how personalization changes segment-level churn interpretation.

Together, these views support the movement from segment-level churn logic to personalized, reliability-aware churn evaluation.

Supports: Chapters 5 - 7

C.1 Design of the Reusable Customer-Level Churn Base View

Purpose
This view creates one reusable customer-level record per ordering customer at the fixed reference date. It consolidates customer order frequency, first and last observed order dates, current recency, product interaction flags, primary customer segment, and RX refinement.

The resulting structure provides the customer-level foundation required to evaluate inactivity consistently across the churn interpretation framework.

Framework Relevance

Churn state assignment requires one consistent customer-level record. This view consolidates order frequency, first and last observed order dates, current recency, product interaction flags, customer segment, and RX subtype.

It acts as the customer-level bridge between the behavioral reconstruction layers and the churn interpretation logic used in Chapters 5 - 7.

Source Foundation

The implementation builds on:

v_churn_orders_24m
v_churn_base_lines_24m

The order-level view provides customer frequency, first order date, last order date, and recency. The order-line-level view provides product interaction information required for segment construction.

Included Behavioral Logic

The view calculates total completed orders, first and last order date, recency in days, FREE / OTC / RX interaction flags, primary customer segment, and RX subtype.

The segmentation hierarchy is: RX CUSTOMER, OTC CUSTOMER, and FREE Only CUSTOMER. Within RX CUSTOMER, the view distinguishes RX + OTC from RX Only.

SQL Implementation - Reusable Customer-Level Churn Base View

```
CREATE OR REPLACE TEMP VIEW v_churn_customer_base_24m AS

WITH params AS (
  -- Step 0: Define fixed reference date
  SELECT DATE('2025-12-31') AS reference_date
),

customer_order_summary AS (
  -- Step 1: Compute customer-level order behavior
  SELECT
    o.super_customer_id_order,

    COUNT(DISTINCT o.main_order_id_bi) AS total_orders,

    MIN(o.order_date) AS first_order_date,

    MAX(o.order_date) AS last_order_date,

    DATEDIFF(p.reference_date, MAX(o.order_date)) AS recency_days

  FROM v_churn_orders_24m o
  CROSS JOIN params p

  GROUP BY
    o.super_customer_id_order,
    p.reference_date
),

order_type_flags AS (
  -- Step 2: Detect product-type presence at order level
  SELECT
    super_customer_id_order,
    main_order_id_bi,

    MAX(CASE WHEN item_sub_type_name = 'FREE' THEN 1 ELSE 0 END) AS has_free,
    MAX(CASE WHEN item_sub_type_name = 'OTC'  THEN 1 ELSE 0 END) AS has_otc,
    MAX(CASE WHEN item_sub_type_name = 'RX'   THEN 1 ELSE 0 END) AS has_rx

  FROM v_churn_base_lines_24m

  GROUP BY
    super_customer_id_order,
    main_order_id_bi
),

order_classification AS (
  -- Step 3: Assign simplified order type using product hierarchy
  SELECT
    super_customer_id_order,
    main_order_id_bi,

    CASE
      WHEN has_rx = 1 THEN 'RX Order'
      WHEN has_otc = 1 THEN 'OTC Order'
      ELSE 'FREE Order'
    END AS order_type,

    has_free,
    has_otc,
    has_rx

  FROM order_type_flags
),

customer_product_profile AS (
  -- Step 4: Collapse product interaction to customer level
  SELECT
    super_customer_id_order,

    MAX(CASE WHEN has_free = 1 THEN 1 ELSE 0 END) AS has_free,
    MAX(CASE WHEN has_otc  = 1 THEN 1 ELSE 0 END) AS has_otc,
    MAX(CASE WHEN has_rx   = 1 THEN 1 ELSE 0 END) AS has_rx,

    MAX(CASE WHEN order_type = 'RX Order'   THEN 1 ELSE 0 END) AS has_rx_order,
    MAX(CASE WHEN order_type = 'OTC Order'  THEN 1 ELSE 0 END) AS has_otc_order,
    MAX(CASE WHEN order_type = 'FREE Order' THEN 1 ELSE 0 END) AS has_free_order

  FROM order_classification
```

```
    GROUP BY super_customer_id_order
),

customer_segments AS (
    -- Step 5: Assign customer-level product segment and RX subtype
    SELECT
        super_customer_id_order,

        has_free,
        has_otc,
        has_rx,

        CASE
            WHEN has_rx = 1 THEN 'RX CUSTOMER'
            WHEN has_otc = 1 THEN 'OTC CUSTOMER'
            ELSE 'FREE Only CUSTOMER'
        END AS customer_segment,

        CASE
            WHEN has_rx = 1 AND has_otc = 1 THEN 'RX + OTC'
            WHEN has_rx = 1 AND has_otc = 0 THEN 'RX Only'
            ELSE NULL
        END AS rx_subtype

    FROM customer_product_profile
)
-- Final output:
-- One reusable customer-level row per customer
SELECT
    -- Primary customer entity used throughout the churn framework.
    s.super_customer_id_order,

    -- Customer-level behavioral frequency.
    o.total_orders,

    -- First observed completed order within the analysis window.
    o.first_order_date,

    -- Last observed completed order within the analysis window.
    o.last_order_date,

    -- Current inactivity at the reference date.
    o.recency_days,

    -- Product interaction flags retained for transparency and auditability.
    s.has_free,
    s.has_otc,
    s.has_rx,

    -- Primary behaviorally grounded customer segment.
    s.customer_segment,

    -- RX refinement layer, populated only for RX customers.
    s.rx_subtype

FROM customer_order_summary o
JOIN customer_segments s
    ON o.super_customer_id_order = s.super_customer_id_order
```

C.2 Design of the Reusable Customer-Level Behavioral Baseline View

Purpose

This view creates customer-level historical return baselines based on observed inter-order gaps. It summarizes each customer's individual ordering rhythm and determines whether the customer has sufficient behavioral history for personalized churn interpretation.

Framework Relevance

Personalized churn interpretation requires a customer-level return baseline. This view summarizes each customer's historical inter-order gaps and determines whether there is enough behavioral history for personalization.

Customers with sufficient history are eligible for personalized interpretation. Customers with limited history remain under segment-level fallback logic, avoiding artificial customer-specific precision.

Source Foundation

The implementation builds on:

v_churn_order_gaps_24m

This source provides the observed inter-order gaps between consecutive completed customer orders.

Included Behavioral Logic

The view calculates gap count, average gap, median gap, standard deviation, minimum and maximum gap, coefficient of variation, and baseline eligibility.

Customers with at least two observed gaps are classified as *ELIGIBLE_FOR_PERSONALIZATION*; all others are assigned to *FALLBACK_TO_SEGMENT_LEVEL.*

SQL Implementation - Reusable Customer-Level Behavioral Baseline View

```
CREATE OR REPLACE TEMP VIEW v_churn_customer_behavioral_baseline_24m AS
WITH customer_gap_metrics AS (

  -- Step 1: Aggregate historical inter-order gaps per customer

  SELECT
    -- Primary customer entity used throughout the churn framework
    super_customer_id_order,

    -- Number of observed inter-order gaps.
    -- A customer with 3 completed orders has 2 gaps.
    COUNT(*) AS gap_count,

    -- Average observed time between consecutive completed orders.
    ROUND(AVG(days_to_next_order), 2) AS avg_days_between_orders,

    -- Median observed time between consecutive completed orders.

    ROUND(PERCENTILE_APPROX(days_to_next_order, 0.5), 2) AS median_days_between_orders,

    -- Standard deviation of inter-order gaps.
    ROUND(STDDEV_SAMP(days_to_next_order), 2) AS stddev_days_between_orders,
```

```
    -- Minimum observed gap.
    MIN(days_to_next_order) AS min_days_between_orders,

    -- Maximum observed gap.
    MAX(days_to_next_order) AS max_days_between_orders

  FROM v_churn_order_gaps_24m

  WHERE
    -- Defensive filter:
    days_to_next_order IS NOT NULL
    AND days_to_next_order >= 0

  GROUP BY
    super_customer_id_order
),

baseline_enriched AS (

  -- Step 2: Add eligibility and relative variability indicators
  SELECT
    super_customer_id_order,
    gap_count,
    avg_days_between_orders,
    median_days_between_orders,
    stddev_days_between_orders,
    min_days_between_orders,
    max_days_between_orders,

    -- Relative variability of the customer's ordering rhythm.
    ROUND(
      CASE
        WHEN avg_days_between_orders > 0
          THEN stddev_days_between_orders / avg_days_between_orders
        ELSE NULL
      END,
      4
    ) AS coefficient_of_variation,

    -- Eligibility flag for personalized churn interpretation.
    CASE
      WHEN gap_count >= 2 THEN 1
      ELSE 0
    END AS baseline_eligible_flag,

    CASE
      WHEN gap_count >= 2 THEN 'ELIGIBLE_FOR_PERSONALIZATION'
      ELSE 'FALLBACK_TO_SEGMENT_LEVEL'
    END AS baseline_eligibility

  FROM customer_gap_metrics
)

SELECT
  *
FROM baseline_enriched
```

C.3 Design of the Reusable Customer-Level Expected Return View

Purpose

This view combines current customer inactivity with each customer's individual behavioral baseline in order to calculate an expected return window and deviation ratio.

The resulting structure provides the core personalized signal used to compare current inactivity against each customer's historical return rhythm.

Framework Relevance

This view compares current customer inactivity with the customer's own historical return rhythm. The resulting deviation ratio shows whether the customer is still within expected behavior or has started to move beyond their usual return pattern.

The view prepares the personalized churn signal but does not assign the final churn state.

Source Foundation

The implementation builds on:

v_churn_customer_base_24m
v_churn_customer_behavioral_baseline_24m

The first source provides customer-level recency and segment context. The second provides the individual historical return baseline.

Included Behavioral Logic

The view retains customer identity, order history, recency, segment, RX subtype, baseline metrics, eligibility flags, expected return days, and deviation ratio.

The deviation ratio is calculated as current recency divided by the customer's median historical inter-order gap.

SQL Implementation - Reusable Customer-Level Expected Return View

```
CREATE OR REPLACE TEMP VIEW v_churn_customer_expected_return_24m AS
SELECT
    -- Customer identity and current inactivity
    c.super_customer_id_order,

    -- Customer-level order frequency and activity window.
    c.total_orders,
    c.first_order_date,
    c.last_order_date,

    -- Current inactivity at the fixed reference date.
    c.recency_days,

    -- Segment context retained for downstream interpretation
    c.customer_segment,
    c.rx_subtype,

    -- Individual behavioral baseline metrics
    b.gap_count,
    b.avg_days_between_orders,
    b.median_days_between_orders,
    b.stddev_days_between_orders,
    b.min_days_between_orders,
    b.max_days_between_orders,
```

```
  b.coefficient_of_variation,
  b.baseline_eligible_flag,
  b.baseline_eligibility,

  -- Expected return window
  CASE
    WHEN b.baseline_eligible_flag = 1
       AND b.median_days_between_orders > 0
      THEN b.median_days_between_orders
    ELSE NULL
  END AS expected_return_days,

  -- Deviation ratio
  -- - 1.0 means the customer is around their expected return point
  -- - >1.0 means the customer is later than their normal rhythm
  -- - <1.0 means the customer is still earlier than their expected return
  ROUND(
    CASE
      WHEN b.baseline_eligible_flag = 1
         AND b.median_days_between_orders > 0
        THEN c.recency_days / b.median_days_between_orders
      ELSE NULL
    END,
    4
  ) AS deviation_ratio

FROM v_churn_customer_base_24m c

LEFT JOIN v_churn_customer_behavioral_baseline_24m b
  ON c.super_customer_id_order = b.super_customer_id_order
```

C.4 Design of the Reusable Personalized Churn State View

Purpose

This view assigns personalized behavioral states to customers based on their deviation from individual expected return behavior.

It transforms the expected return and deviation logic into an explainable customer-level churn state that reflects each customer's own historical rhythm.

Framework Relevance

This view assigns personalized churn states by comparing current inactivity with each customer's expected return rhythm. It keeps the same state structure used in the thesis — ACTIVE, AT RISK, and CHURNED — but makes the interpretation customer-specific where enough history exists.

Behavioral consistency is included as a reliability signal. It supports interpretation confidence but does not change the assigned churn state.

Source Foundation

The implementation builds on:

v_churn_customer_expected_return_24m

This source provides current inactivity, expected return days, deviation ratio, baseline eligibility, segment context, and RX subtype context.

Included Behavioral Logic

The view assigns behavioral consistency buckets based on the coefficient of variation and assigns personalized states using deviation ratio thresholds:

- ACTIVE: deviation ratio ≤ 1.00
- AT RISK: deviation ratio > 1.00 and ≤ 1.50
- CHURNED: deviation ratio > 1.50
- FALLBACK_TO_SEGMENT_LEVEL: insufficient history for personalized interpretation

SQL Implementation - Reusable Personalized Churn State View

```
CREATE OR REPLACE TEMP VIEW v_churn_personalized_states_24m AS
SELECT
  -- Customer identity and current customer-level behavior
  super_customer_id_order,
  total_orders,
  first_order_date,
  last_order_date,
  recency_days,

  -- Segment context retained for interpretation and downstream
  -- comparison
  customer_segment,
  rx_subtype,

  -- Individual baseline and expected-return information
  gap_count,
```

```
  avg_days_between_orders,
  median_days_between_orders,
  stddev_days_between_orders,
  min_days_between_orders,
  max_days_between_orders,
  coefficient_of_variation,
  baseline_eligible_flag,
  baseline_eligibility,
  expected_return_days,
  deviation_ratio,

  -- Behavioral consistency bucket
  CASE
    WHEN baseline_eligible_flag <> 1
         OR expected_return_days IS NULL
         OR deviation_ratio IS NULL
        THEN 'Not Eligible for Personalization'

    WHEN coefficient_of_variation IS NULL
        THEN 'Undefined Consistency'

    WHEN coefficient_of_variation <= 0.50
        THEN 'Consistent'

    WHEN coefficient_of_variation > 0.50
         AND coefficient_of_variation <= 1.00
        THEN 'Moderately Variable'

    WHEN coefficient_of_variation > 1.00
        THEN 'Highly Variable'

    ELSE 'Undefined Consistency'
  END AS behavioral_consistency_bucket,

  CASE
    WHEN baseline_eligible_flag <> 1
         OR expected_return_days IS NULL
         OR deviation_ratio IS NULL
        THEN 'FALLBACK_TO_SEGMENT_LEVEL'

    WHEN deviation_ratio <= 1.00
        THEN 'ACTIVE'

    WHEN deviation_ratio > 1.00
         AND deviation_ratio <= 1.50
        THEN 'AT RISK'

    WHEN deviation_ratio > 1.50
        THEN 'CHURNED'

    ELSE 'FALLBACK_TO_SEGMENT_LEVEL'
  END AS personalized_state,

  -- Explicit ordering for reporting
  CASE
    WHEN baseline_eligible_flag <> 1
         OR expected_return_days IS NULL
         OR deviation_ratio IS NULL
        THEN 0

    WHEN deviation_ratio <= 1.00
        THEN 1

    WHEN deviation_ratio > 1.00
         AND deviation_ratio <= 1.50
        THEN 2

    WHEN deviation_ratio > 1.50
        THEN 3

    ELSE 0
  END AS personalized_state_order
FROM v_churn_customer_expected_return_24m
```

C.5 Design of the Reusable Reclassification Impact View

Purpose

This view compares segment-level churn states with personalized churn states at customer level in order to measure how personalization changes customer interpretation.

It creates the reusable analytical base used to evaluate whether customer-state assignment remains aligned or changes when individual return behavior is considered.

Framework Relevance

This view compares segment-level churn states with personalized churn states. It measures where both approaches agree and where customer-level return behavior changes the interpretation.

The comparison supports the reclassification analysis in Chapter 7, including hidden risk, missed churn, late detection, premature risk assignment, and softer personalized interpretation.

Source Foundation

The implementation builds on:

v_churn_personalized_states_24m

It reconstructs the segment-level churn state logic from Chapter 5 using the segment-specific empirical P80 and P90 return thresholds.

Included Behavioral Logic

The view assigns segment-level states, compares them with personalized states, identifies whether each customer is unchanged or reclassified, and labels the business meaning of each reclassification direction.

SQL Implementation - Reusable Reclassification Impact View

```
CREATE OR REPLACE TEMP VIEW v_churn_reclassification_impact_24m AS

WITH segment_thresholds AS (

  -- Step 1: Define segment-level churn thresholds from Chapter 5
  -- - ACTIVE: recency_days <= P80
  -- - AT RISK: P80 < recency_days <= P90
  -- - CHURNED: recency_days > P90
  SELECT
    'RX CUSTOMER' AS customer_segment,
    83 AS p80_days,
    127 AS p90_days

  UNION ALL

  SELECT
    'OTC CUSTOMER' AS customer_segment,
    124 AS p80_days,
    194 AS p90_days

  UNION ALL

  SELECT
    'FREE Only CUSTOMER' AS customer_segment,
    201 AS p80_days,
```

```
    299 AS p90_days
),

comparison_base AS (

  -- Step 2: Build customer-level comparison base

  SELECT
    -- Customer identity
    p.super_customer_id_order,

    -- Segment context
    p.customer_segment,
    p.rx_subtype,

    -- Current inactivity and personalized behavioral signal
    p.recency_days,
    p.expected_return_days,
    p.deviation_ratio,

    -- Reliability context from Chapter 6
    p.behavioral_consistency_bucket,

    -- Personalized churn state from Chapter 6
    p.personalized_state,
    p.personalized_state_order,

    -- Segment-level thresholds from Chapter 5
    t.p80_days,
    t.p90_days,

    -- Segment-level churn state reconstructed from Chapter 5 logic
    CASE
      WHEN p.recency_days <= t.p80_days
        THEN 'ACTIVE'

      WHEN p.recency_days > t.p80_days
         AND p.recency_days <= t.p90_days
        THEN 'AT RISK'

      WHEN p.recency_days > t.p90_days
        THEN 'CHURNED'

      ELSE 'UNDEFINED'
    END AS segment_state,

    -- Explicit ordering for stable reporting and charts
    CASE
      WHEN p.recency_days <= t.p80_days
        THEN 1

      WHEN p.recency_days > t.p80_days
         AND p.recency_days <= t.p90_days
        THEN 2

      WHEN p.recency_days > t.p90_days
        THEN 3

      ELSE 0
    END AS segment_state_order

  FROM v_churn_personalized_states_24m p

  INNER JOIN segment_thresholds t
    ON p.customer_segment = t.customer_segment

  WHERE
    -- Keep only customers with valid personalized interpretation
    p.personalized_state IN ('ACTIVE', 'AT RISK', 'CHURNED')
),

reclassification_enriched AS (

  -- Step 3: Add reclassification status and business direction
  SELECT
    *,

    -- Changed vs unchanged classification
    CASE
      WHEN segment_state = personalized_state
        THEN 'UNCHANGED'
```

```
        ELSE 'RECLASSIFIED'
    END AS reclassification_status,

    -- Business-oriented reclassification direction
    CASE
        WHEN segment_state = personalized_state
            THEN 'Aligned'

        WHEN segment_state = 'ACTIVE'
             AND personalized_state = 'AT RISK'
            THEN 'Hidden Risk'

        WHEN segment_state = 'ACTIVE'
             AND personalized_state = 'CHURNED'
            THEN 'Missed Churn'

        WHEN segment_state = 'AT RISK'
             AND personalized_state = 'CHURNED'
            THEN 'Late Detection'

        WHEN segment_state = 'AT RISK'
             AND personalized_state = 'ACTIVE'
            THEN 'Premature Risk'

        WHEN segment_state = 'CHURNED'
             AND personalized_state IN ('ACTIVE', 'AT RISK')
            THEN 'Softer Interpretation'

        ELSE 'Other Reclassification'
    END AS reclassification_direction,

    -- Explicit ordering for business-oriented reporting
    CASE
        WHEN segment_state = personalized_state
            THEN 1

        WHEN segment_state = 'ACTIVE'
             AND personalized_state = 'AT RISK'
            THEN 2

        WHEN segment_state = 'ACTIVE'
             AND personalized_state = 'CHURNED'
            THEN 3

        WHEN segment_state = 'AT RISK'
             AND personalized_state = 'CHURNED'
            THEN 4

        WHEN segment_state = 'AT RISK'
             AND personalized_state = 'ACTIVE'
            THEN 5

        WHEN segment_state = 'CHURNED'
             AND personalized_state IN ('ACTIVE', 'AT RISK')
            THEN 6

        ELSE 7
    END AS reclassification_direction_order

  FROM comparison_base

  WHERE
    -- Defensive filter:
    -- Keep only valid segment-level states for comparison.
    segment_state IN ('ACTIVE', 'AT RISK', 'CHURNED')
)

SELECT
  *
FROM reclassification_enriched
```

Appendix D - Predictive Modeling: Future Churn-State Prediction

Purpose

This appendix documents the core implementation logic used to construct, train, evaluate, and persist the predictive modeling layer for future customer churn-state prediction. Building on the behavioral foundations developed in Appendices A–C, the predictive layer translates historical customer behavior into a leakage-safe supervised multiclass modeling dataset and evaluates future churn-state prediction across the classes ACTIVE, AT RISK, and CHURNED.

The appendix does not reproduce the entire Databricks notebook. The full notebook includes additional validation checks, exploratory outputs, chart-generation cells, and artifact reload steps. For readability and academic clarity, this appendix includes only the core implementation blocks required to understand and reproduce the predictive modeling architecture: temporal parameterization, feature-base construction, target assignment, Spark ML dataset preparation, distributed model training, evaluation, and artifact persistence.

Supports: Chapter 8

Technical Execution Environment

The predictive modeling implementation was developed in Python on Databricks, using SQL-based feature engineering and distributed Spark-based ML.

Exploratory validation was initially performed on Databricks Serverless compute. The final modeling workflow was executed on a dedicated Databricks ML environment using Databricks Runtime 17.3 LTS ML, in order to support Spark ML, XGBoost, distributed training, and persistent artifact storage.

The implementation uses PySpark, Spark MLlib, Spark-compatible XGBoost, Delta Lake, and DBFS-based storage.

Code Selection Principle

The full Databricks notebook contains additional validation checks, exploratory outputs, chart-generation steps, and artifact reload cells. The appendix includes only the core implementation blocks needed to document the predictive modeling workflow.

The selected code covers temporal parameterization, feature-base construction, target assignment, Spark ML preprocessing, model training, evaluation, and artifact persistence.

D.1 Design of the Reusable ML Feature Base

Purpose

This section defines the reusable customer-level ML feature base used for future churn-state prediction. The feature base is constructed exclusively from customer behavior observed before the cutoff date and therefore represents only the information available at the time of prediction.

Framework Relevance

This layer constructs the customer-level feature base using only behavior observed before the cutoff date. It separates historical feature construction from future target assignment, which is essential for leakage prevention.

The feature base combines recency, frequency, return behavior, consistency, customer segment, product interaction, and transaction-value signals.

Source Foundation

The implementation builds on the governed transactional scope used throughout the thesis, but introduces a stricter temporal split:

- observation window: 2023-12-31 to 2025-06-30,
- cutoff date: 2025-06-30,
- prediction window: 2025-07-01 to 2025-12-31.

Included Logic

The implementation defines temporal parameters, separates observation and prediction data, reconstructs observation-window orders and gaps, and integrates the final customer-level ML feature base. Intermediate feature views are not fully reproduced in the appendix to avoid unnecessary notebook duplication.

SQL Implementation - Temporal Modeling Parameters

```
CREATE OR REPLACE TEMP VIEW v_churn_ml_params AS
SELECT
   DATE('2023-12-31') AS observation_start_date,
   DATE('2025-06-30') AS cutoff_date,
   DATE('2025-07-01') AS prediction_start_date,
   DATE('2025-12-31') AS prediction_end_date
```

SQL Implementation - Scoped Transactional Base for ML

```
CREATE OR REPLACE TEMP VIEW v_churn_ml_base_lines_24m AS
WITH params AS (
   SELECT *
   FROM v_churn_ml_params
)
SELECT
   -- Date of the completed order line.
   -- Used to separate observation and prediction windows.
   f.order_date,

   -- Business-level order identifier.
   -- Used to reconstruct completed customer orders.
   f.main_order_id_bi,

   -- Ordering customer identifier.
   -- This remains the primary entity for churn prediction.
   f.super_customer_id_order,

   -- Product subtype used for behavioral segmentation.
   f.item_sub_type_name,

   -- Net reported transaction value.
   -- Used only as a behavioral economic activity proxy, not profit.
```

```
    f.sales_price_net_rep_cur

FROM rdc_udp_globalsales_globalsales_dev.default.fact_sales_trans f
CROSS JOIN params p
WHERE
    -- Full 24-month analytical window used for Chapter 8.
    f.order_date BETWEEN p.observation_start_date AND p.prediction_end_date

    -- Business scope restriction.
    AND f.sales_domain = 'os'
    AND f.sub_company_id = 'Shop-COM'

    -- Only completed customer transactions.
    AND f.order_line_status = 'Processed'

    -- Exclude system-structural BOM component lines.
    AND (f.bom_type_name IS NULL OR f.bom_type_name <> 'Stücklistenposition')

    -- Restrict to the governed product scope.
    AND f.item_sub_type_name IN ('FREE', 'OTC', 'RX')

    -- Ensure consistency for value-based behavioral features.
    AND f.sales_price_net_rep_cur IS NOT NULL
```

SQL Implementation - Observation and Prediction Window Separation

```
CREATE OR REPLACE TEMP VIEW v_churn_ml_observation_lines AS
WITH params AS (
    SELECT *
    FROM v_churn_ml_params
)
SELECT
    b.*
FROM v_churn_ml_base_lines_24m b
CROSS JOIN params p
WHERE
    b.order_date BETWEEN p.observation_start_date AND p.cutoff_date
```

```
CREATE OR REPLACE TEMP VIEW v_churn_ml_prediction_lines AS
WITH params AS (
    SELECT *
    FROM v_churn_ml_params
)
SELECT
    b.*
FROM v_churn_ml_base_lines_24m b
CROSS JOIN params p
WHERE
    b.order_date BETWEEN p.prediction_start_date AND p.prediction_end_date
```

SQL Implementation - Observation-Window Order Base

```
CREATE OR REPLACE TEMP VIEW v_churn_ml_observation_orders AS
SELECT
    -- Customer identifier (modeling entity)
    super_customer_id_order,

    -- Order identifier
    main_order_id_bi,

    -- Order date (used for recency & gap calculations)
    MIN(order_date) AS order_date,

    -- Total economic activity per order
    SUM(sales_price_net_rep_cur) AS order_value,

    -- Number of lines in the order (optional behavioral signal)
    COUNT(*) AS order_lines,

    -- Product-type composition flags (important for segmentation later)
    MAX(CASE WHEN item_sub_type_name = 'RX' THEN 1 ELSE 0 END) AS has_rx,
    MAX(CASE WHEN item_sub_type_name = 'OTC' THEN 1 ELSE 0 END) AS has_otc,
    MAX(CASE WHEN item_sub_type_name = 'FREE' THEN 1 ELSE 0 END) AS has_free
```

```
FROM v_churn_ml_observation_lines
GROUP BY
   super_customer_id_order,
   main_order_id_bi
```

SQL Implementation - Observation-Window Inter-Order Gaps

```
CREATE OR REPLACE TEMP VIEW v_churn_ml_observation_order_gaps AS
SELECT
   -- Customer identifier
   super_customer_id_order,

   -- Current order
   main_order_id_bi,
   order_date,

   -- Next order within observation window
   LEAD(order_date) OVER (
      PARTITION BY super_customer_id_order
      ORDER BY order_date
   ) AS next_order_date,

   -- Days between consecutive orders
   DATEDIFF(
      LEAD(order_date) OVER (
         PARTITION BY super_customer_id_order
         ORDER BY order_date
      ),
      order_date
   ) AS days_to_next_order

FROM v_churn_ml_observation_orders
```

SQL Implementation - Final ML Feature Base

```
CREATE OR REPLACE TEMP VIEW v_churn_ml_feature_base AS
SELECT
   rf.super_customer_id_order,

   -- Observation-window anchor date
   rf.last_order_date_observation,

   -- Recency & Frequency
   rf.total_orders_observation,
   rf.days_since_last_order,
   rf.orders_last_3m,
   rf.orders_last_6m,
   rf.orders_last_12m,
   rf.inactivity_bucket,
   rf.frequency_bucket,

   -- Return Behavior
   rb.gap_count,
   rb.median_gap_days,
   rb.avg_gap_days,
   rb.expected_return_days,

   -- Consistency
    c.gap_cv,
   c.consistency_bucket,

   -- Segment & Product
   s.customer_segment,
   s.rx_subtype,
   s.is_rx_plus_otc_customer,
   s.is_otc_only_customer,
   s.is_free_only_customer,

   -- Value
   v.total_transaction_value,
   v.avg_order_value,
   v.std_order_value

FROM v_churn_ml_recency_frequency_features rf
```

```
LEFT JOIN v_churn_ml_return_behavior_features rb
   ON rf.super_customer_id_order = rb.super_customer_id_order

LEFT JOIN v_churn_ml_consistency_features c
   ON rf.super_customer_id_order = c.super_customer_id_order

LEFT JOIN v_churn_ml_segment_product_features s
   ON rf.super_customer_id_order = s.super_customer_id_order

LEFT JOIN v_churn_ml_value_features v
   ON rf.super_customer_id_order = v.super_customer_id_order
```

D.2 Design of the Reusable Future Churn Target View

Purpose

This section defines the future churn-state target used for supervised multiclass prediction. The target is constructed from prediction-window behavior and compares each customer's future return behavior with their expected return baseline.

Supports: Chapter 8

Framework Relevance

The target layer assigns the future customer state after the feature base has already been constructed. Prediction-window behavior is used only for label creation, not for model features.

Customers without a usable expected return baseline are assigned to UNDEFINED and excluded from supervised training.

Source Foundation

The target layer builds on:

v_churn_ml_feature_base,

v_churn_ml_prediction_lines,

v_churn_ml_prediction_orders,

v_churn_ml_future_gap.

Included Logic

The implementation identifies the first future order after the cutoff date, calculates the future gap, compares that gap with the customer's expected return window, and assigns the future churn state.

SQL Implementation - Prediction-Window Orders

```
CREATE OR REPLACE TEMP VIEW v_churn_ml_prediction_orders AS
SELECT
   super_customer_id_order,
   MIN(order_date) AS next_order_date
FROM v_churn_ml_prediction_lines
GROUP BY
   super_customer_id_order
```

SQL Implementation - Future Gap Calculation

```
CREATE OR REPLACE TEMP VIEW v_churn_ml_future_gap AS

WITH params AS (
   SELECT *
   FROM v_churn_ml_params
)

SELECT
   f.super_customer_id_order,
```

```
  -- Last observed order before the cutoff date
  f.last_order_date_observation,

  -- Current inactivity at cutoff
  f.days_since_last_order,

  -- Personalized expected return baseline
  f.expected_return_days,

  -- First observed order during prediction window
  p.next_order_date,

  -- Total future behavioral delay
  CASE
    WHEN p.next_order_date IS NOT NULL THEN
      DATEDIFF(
        p.next_order_date,
        f.last_order_date_observation
      )

    ELSE
      DATEDIFF(
        params.prediction_end_date,
        f.last_order_date_observation
      )
  END AS future_gap_days

FROM v_churn_ml_feature_base f

CROSS JOIN params

LEFT JOIN v_churn_ml_prediction_orders p
  ON f.super_customer_id_order = p.super_customer_id_order
```

SQL Implementation - Future Churn State Target

```
CREATE OR REPLACE TEMP VIEW v_churn_ml_target AS

SELECT
  *,

  -- Future deviation ratio
  ROUND(
    CASE
      WHEN expected_return_days IS NOT NULL
           AND expected_return_days > 0
        THEN future_gap_days / expected_return_days
      ELSE NULL
    END,
    4
  ) AS future_deviation_ratio,

  -- Future churn state
  CASE
    WHEN expected_return_days IS NULL
         OR expected_return_days = 0
      THEN 'UNDEFINED'

    WHEN future_gap_days / expected_return_days <= 1.0
      THEN 'ACTIVE'

    WHEN future_gap_days / expected_return_days <= 1.5
      THEN 'AT RISK'

    ELSE 'CHURNED'
  END AS future_churn_state

FROM v_churn_ml_future_gap
```

D.3 Design of the Reusable Spark ML Modeling Pipeline

Purpose

This section documents the distributed Spark ML pipeline used to train and compare multiclass churn-state prediction models. The pipeline prepares the labeled customer-level modeling dataset, applies preprocessing, performs stratified train/test splitting, and trains Logistic Regression, Random Forest, and XGBoost models.

Supports: Chapter 8

Framework Relevance

This layer trains and compares three multiclass models using the same leakage-safe behavioral feature base and target definition. The goal is not exhaustive model optimization, but a controlled comparison between an interpretable baseline, a non-linear ensemble model, and a boosted tree model.

The implementation uses Spark-based distributed processing, which is appropriate for customer-level modeling across millions of records.

Source Foundation

The modeling pipeline builds on:

v_churn_ml_feature_base,

v_churn_ml_target,

and the labeled customer population where future_churn_state is ACTIVE, AT RISK, or CHURNED.

Included Logic

The pipeline constructs the modeling dataset, excludes UNDEFINED targets, defines categorical and numerical features, applies missing-value handling and encoding, creates a stratified train/test split, and trains Logistic Regression, Random Forest, and XGBoost models.

Python Implementation - Final Spark ML Modeling Dataset

```
from pyspark.sql import functions as F

modeling_df = spark.sql("""
SELECT
   f.*,
   t.future_churn_state
FROM v_churn_ml_feature_base f
JOIN v_churn_ml_target t
   ON f.super_customer_id_order = t.super_customer_id_order
WHERE t.future_churn_state IN ('ACTIVE', 'AT RISK', 'CHURNED')
""")

print("Modeling rows:", modeling_df.count())
modeling_df.groupBy("future_churn_state").count().orderBy("future_churn_state").show()
```

Python Implementation - Feature Columns and Target

```
target_col = "future_churn_state"
id_col = "super_customer_id_order"

categorical_cols = [
    "inactivity_bucket",
    "frequency_bucket",
    "consistency_bucket",
    "customer_segment",
    "rx_subtype"
]

numeric_cols = [
    "total_orders_observation",
    "days_since_last_order",
    "orders_last_3m",
    "orders_last_6m",
    "orders_last_12m",
    "gap_count",
    "median_gap_days",
    "avg_gap_days",
    "expected_return_days",
    "gap_cv",
    "is_rx_plus_otc_customer",
    "is_otc_only_customer",
    "is_free_only_customer",
    "total_transaction_value",
    "avg_order_value",
    "std_order_value"
]

modeling_df = modeling_df.fillna("UNDEFINED", subset=categorical_cols)

selected_cols = [id_col, target_col] + categorical_cols + numeric_cols
modeling_df = modeling_df.select(selected_cols)
```

Python Implementation - Stratified Train/Test Split

```
seed = 42
test_fraction = 0.20

fractions = {
    "ACTIVE": test_fraction,
    "AT RISK": test_fraction,
    "CHURNED": test_fraction
}

test_df = modeling_df.stat.sampleBy(
    col=target_col,
    fractions=fractions,
    seed=seed
)

train_df = modeling_df.join(
    test_df.select(id_col),
    on=id_col,
    how="left_anti"
)

print("Train rows:", train_df.count())
print("Test rows:", test_df.count())

print("Train distribution:")
train_df.groupBy(target_col).count().orderBy(target_col).show()

print("Test distribution:")
test_df.groupBy(target_col).count().orderBy(target_col).show()
```

Python Implementation - Model-Aware Spark ML Preprocessing

```
from pyspark.ml.feature import (
    StringIndexer,
    OneHotEncoder,
    VectorAssembler,
```

```
    StandardScaler,
    Imputer
)

label_indexer = StringIndexer(
    inputCol=target_col,
    outputCol="label",
    handleInvalid="keep"
)

categorical_indexers = [
    StringIndexer(
        inputCol=c,
        outputCol=f"{c}_idx",
        handleInvalid="keep"
    )
    for c in categorical_cols
]

encoder = OneHotEncoder(
    inputCols=[f"{c}_idx" for c in categorical_cols],
    outputCols=[f"{c}_ohe" for c in categorical_cols]
)

imputer = Imputer(
    inputCols=numeric_cols,
    outputCols=[f"{c}_imputed" for c in numeric_cols],
    strategy="median"
)

assembler_inputs = (
    [f"{c}_ohe" for c in categorical_cols] +
    [f"{c}_imputed" for c in numeric_cols]
)

tree_assembler = VectorAssembler(
    inputCols=assembler_inputs,
    outputCol="features_tree",
    handleInvalid="keep"
)

lr_assembler = VectorAssembler(
    inputCols=assembler_inputs,
    outputCol="features_unscaled",
    handleInvalid="keep"
)

lr_scaler = StandardScaler(
    inputCol="features_unscaled",
    outputCol="features_lr",
    withStd=True,
    withMean=False
)

print("Model-aware preprocessing objects created successfully.")
```

Python Implementation - Logistic Regression Model

```
from pyspark.ml import Pipeline
from pyspark.ml.classification import LogisticRegression

lr = LogisticRegression(
    featuresCol="features_lr",
    labelCol="label",
    predictionCol="prediction",
    probabilityCol="probability",
    maxIter=20,
    regParam=0.01,
    elasticNetParam=0.0,
    family="multinomial"
)

lr_pipeline = Pipeline(stages=[
    label_indexer,
    *categorical_indexers,
    encoder,
    imputer,
    lr_assembler,
```

```
    lr_scaler,
    lr
])

lr_model = lr_pipeline.fit(train_df)

lr_predictions = lr_model.transform(test_df)

lr_predictions.select(
    id_col,
    target_col,
    "label",
    "prediction",
    "probability"
).show(5, truncate=False)
```

Python Implementation - Random Forest Model

```
from pyspark.ml import Pipeline
from pyspark.ml.classification import RandomForestClassifier

rf = RandomForestClassifier(
    featuresCol="features_tree",
    labelCol="label",
    predictionCol="prediction",
    probabilityCol="probability",
    numTrees=50,
    maxDepth=8,
    minInstancesPerNode=100,
    seed=42
)

rf_pipeline = Pipeline(stages=[
    label_indexer,
    *categorical_indexers,
    encoder,
    imputer,
    tree_assembler,
    rf
])

rf_model = rf_pipeline.fit(train_df)

rf_predictions = rf_model.transform(test_df)

rf_predictions.select(
    id_col,
    target_col,
    "label",
    "prediction",
    "probability"
).show(5, truncate=False)
```

Python Implementation - XGBoost Model

```
from pyspark.ml import Pipeline
from xgboost.spark import SparkXGBClassifier

xgb = SparkXGBClassifier(
    features_col="features_tree",
    label_col="label",
    prediction_col="prediction",
    probability_col="probability",
    max_depth=6,
    eta=0.1,
    num_round=50,
    subsample=0.8,
    colsample_bytree=0.8,
    seed=42
)

xgb_pipeline = Pipeline(stages=[
    label_indexer,
    *categorical_indexers,
    encoder,
```

```
    imputer,
    tree_assembler,
    xgb
])

xgb_model = xgb_pipeline.fit(train_df)

xgb_predictions = xgb_model.transform(test_df)

xgb_predictions.select(
    id_col,
    target_col,
    "label",
    "prediction",
    "probability"
).show(5, truncate=False)
```

D.4 Design of the Reusable Prediction Evaluation and Persistence Layer

Purpose

This section documents the reusable evaluation and persistence layer used to compare model performance and preserve prediction artifacts and to ensure that evaluation outputs remain reproducible, reloadable, and available for later visualizations and operational decision-system logic.

Supports: Chapters 8 - 9

Framework Relevance

This layer evaluates model performance and persists outputs so that results can be reused after notebook or cluster restart. It supports reproducibility and makes the prediction artifacts available for later analysis.

The evaluation includes confusion matrices, per-class precision, recall and F1-score, aggregate model comparison, and AT RISK-focused metrics.

Source Foundation

This layer builds on the prediction outputs generated by:

- Logistic Regression,
- Random Forest,
- XGBoost.

Included Logic

The implementation extracts readable labels, builds confusion matrices, calculates precision, recall, and F1-score, creates aggregate comparison metrics, isolates AT RISK performance, and persists the reusable prediction and evaluation outputs.

Python Implementation - Label Mapping and Readable Predictions

```
from pyspark.sql import functions as F

def get_label_mapping(pipeline_model):
    label_indexer_model = pipeline_model.stages[0]
    labels = label_indexer_model.labels
    return {float(i): label for i, label in enumerate(labels)}

def add_readable_labels(predictions_df, label_mapping):
    actual_expr = None
    predicted_expr = None

    for index, label in label_mapping.items():
        if actual_expr is None:
            actual_expr = F.when(F.col("label") == index, F.lit(label))
            predicted_expr = F.when(F.col("prediction") == index, F.lit(label))
        else:
            actual_expr = actual_expr.when(F.col("label") == index, F.lit(label))
            predicted_expr = predicted_expr.when(F.col("prediction") == index, F.lit(label))

    return (
        predictions_df
        .withColumn("actual_state", actual_expr.otherwise(F.lit("UNKNOWN")))
        .withColumn("predicted_state", predicted_expr.otherwise(F.lit("UNKNOWN")))
```

```
    )

# Extract label mapping separately from each fitted pipeline
lr_label_mapping = get_label_mapping(lr_model)
rf_label_mapping = get_label_mapping(rf_model)
xgb_label_mapping = get_label_mapping(xgb_model)

print("Logistic Regression label mapping:")
for k, v in lr_label_mapping.items():
    print(f"{k} -> {v}")

print("Random Forest label mapping:")
for k, v in rf_label_mapping.items():
    print(f"{k} -> {v}")

print("XGBoost label mapping:")
for k, v in xgb_label_mapping.items():
    print(f"{k} -> {v}")

# Add readable labels using each model's own fitted label mapping
lr_eval = add_readable_labels(lr_predictions, lr_label_mapping)
rf_eval = add_readable_labels(rf_predictions, rf_label_mapping)
xgb_eval = add_readable_labels(xgb_predictions, xgb_label_mapping)

print("Label mapping:")
for k, v in xgb_label_mapping.items():
    print(f"{k} -> {v}")
```

Python Implementation - Confusion Matrices

```
def create_confusion_matrix(eval_df, model_name):
    return (
        eval_df
        .groupBy("actual_state", "predicted_state")
        .count()
        .withColumn("model", F.lit(model_name))
        .select("model", "actual_state", "predicted_state", "count")
        .orderBy("actual_state", "predicted_state")
    )

lr_cm = create_confusion_matrix(lr_eval, "Logistic Regression")
rf_cm = create_confusion_matrix(rf_eval, "Random Forest")
xgb_cm = create_confusion_matrix(xgb_eval, "XGBoost")

print("Logistic Regression Confusion Matrix")
lr_cm.show(50, truncate=False)

print("Random Forest Confusion Matrix")
rf_cm.show(50, truncate=False)

print("XGBoost Confusion Matrix")
xgb_cm.show(50, truncate=False)
```

Python Implementation - Per-Class Metrics

```
from pyspark.sql import functions as F

all_cm = (
    lr_cm
    .unionByName(rf_cm)
    .unionByName(xgb_cm)
    .cache()
)

all_cm.count()

states_df = spark.createDataFrame(
    [("ACTIVE",), ("AT RISK",), ("CHURNED",)],
    ["state"]
)

models_df = spark.createDataFrame(
    [("Logistic Regression",), ("Random Forest",), ("XGBoost",)],
    ["model"]
)
```

```
model_state_grid = models_df.crossJoin(states_df)

tp = (
    all_cm
    .filter(F.col("actual_state") == F.col("predicted_state"))
    .groupBy("model", F.col("actual_state").alias("state"))
    .agg(F.sum("count").alias("tp"))
)

fp = (
    all_cm
    .filter(F.col("actual_state") != F.col("predicted_state"))
    .groupBy("model", F.col("predicted_state").alias("state"))
    .agg(F.sum("count").alias("fp"))
)

fn = (
    all_cm
    .filter(F.col("actual_state") != F.col("predicted_state"))
    .groupBy("model", F.col("actual_state").alias("state"))
    .agg(F.sum("count").alias("fn"))
)

support = (
    all_cm
    .groupBy("model", F.col("actual_state").alias("state"))
    .agg(F.sum("count").alias("support"))
)

class_metrics_df = (
    model_state_grid
    .join(tp, ["model", "state"], "left")
    .join(fp, ["model", "state"], "left")
    .join(fn, ["model", "state"], "left")
    .join(support, ["model", "state"], "left")
    .fillna(0, subset=["tp", "fp", "fn", "support"])
    .withColumn(
        "precision",
        F.when((F.col("tp") + F.col("fp")) > 0,
               F.col("tp") / (F.col("tp") + F.col("fp")))
         .otherwise(F.lit(0.0))
    )
    .withColumn(
        "recall",
        F.when((F.col("tp") + F.col("fn")) > 0,
               F.col("tp") / (F.col("tp") + F.col("fn")))
         .otherwise(F.lit(0.0))
    )
    .withColumn(
        "f1_score",
        F.when((F.col("precision") + F.col("recall")) > 0,
               2 * F.col("precision") * F.col("recall") /
               (F.col("precision") + F.col("recall")))
         .otherwise(F.lit(0.0))
    )
    .select(
        F.col("model").alias("Model"),
        F.col("state").alias("Class"),
        F.col("support").cast("long").alias("Support"),
        F.round("precision", 4).alias("Precision"),
        F.round("recall", 4).alias("Recall"),
        F.round("f1_score", 4).alias("F1 Score")
    )
    .orderBy("Model", "Class")
)

class_metrics_df.show(50, truncate=False)
```

Python Implementation - Aggregate Model Comparison and AT RISK Focus

```
accuracy_df = (
    all_cm
    .groupBy("model")
    .agg(
        F.sum("count").alias("total_predictions"),
        F.sum(
            F.when(F.col("actual_state") == F.col("predicted_state"), F.col("count"))
```

```
            .otherwise(F.lit(0))
        ).alias("correct_predictions")
    )
    .withColumn("Accuracy", F.round(F.col("correct_predictions") / F.col("total_predictions"), 4))
    .select(F.col("model").alias("Model"), "Accuracy")
)

aggregate_metrics_df = (
    class_metrics_df
    .groupBy("Model")
    .agg(
        F.sum("Support").alias("Total Support"),
        F.round(F.sum(F.col("Precision") * F.col("Support")) / F.sum("Support"), 4).alias("Weighted Precision"),
        F.round(F.sum(F.col("Recall") * F.col("Support")) / F.sum("Support"), 4).alias("Weighted Recall"),
        F.round(F.sum(F.col("F1 Score") * F.col("Support")) / F.sum("Support"), 4).alias("Weighted F1"),
        F.round(F.avg("F1 Score"), 4).alias("Macro F1")
    )
)

model_comparison_df = (
    aggregate_metrics_df
    .join(accuracy_df, on="Model", how="left")
    .select(
        "Model",
        "Accuracy",
        "Weighted Precision",
        "Weighted Recall",
        "Weighted F1",
        "Macro F1",
        "Total Support"
    )
    .orderBy(F.desc("Macro F1"))
)

model_comparison_df.show(truncate=False)
```

```
at_risk_metrics_df = (
    class_metrics_df
    .filter(F.col("Class") == "AT RISK")
    .select("Model", "Class", "Support", "Precision", "Recall", "F1 Score")
    .orderBy(F.desc("F1 Score"))
)

at_risk_metrics_df.show(truncate=False)
```

Python Implementation - Persist Prediction Outputs

```
from pyspark.sql import functions as F
from pyspark.ml.functions import vector_to_array

output_schema = "rdc_udp_globalsales_globalsales_dev.default"
run_id = "churn_sparkml_v1"
output_path = "dbfs:/FileStore/eb_thesis/churn_all_model_predictions_delta"


def get_label_mapping(pipeline_model):
    label_indexer_model = pipeline_model.stages[0]
    labels = label_indexer_model.labels
    return {float(i): label for i, label in enumerate(labels)}


def prepare_prediction_output(predictions_df, model_name, label_mapping):
    df = predictions_df.withColumn(
        "probability_array",
        vector_to_array(F.col("probability"))
    )

    actual_expr = None
    predicted_expr = None

    for idx, label in label_mapping.items():
        if actual_expr is None:
            actual_expr = F.when(F.col("label") == idx, F.lit(label))
            predicted_expr = F.when(F.col("prediction") == idx, F.lit(label))
        else:
            actual_expr = actual_expr.when(F.col("label") == idx, F.lit(label))
```

```
        predicted_expr = predicted_expr.when(F.col("prediction") == idx, F.lit(label))

    df = (
        df
        .withColumn("actual_state", actual_expr.otherwise(F.lit("UNKNOWN")))
        .withColumn("predicted_state", predicted_expr.otherwise(F.lit("UNKNOWN")))
        .withColumn("model", F.lit(model_name))
        .withColumn("run_id", F.lit(run_id))
    )

    for idx, label in label_mapping.items():
        clean_label = label.lower().replace(" ", "_")
        df = df.withColumn(
            f"prob_{clean_label}",
            F.col("probability_array")[int(idx)]
        )

    return df


prediction_columns = [
    "run_id",
    "model",
    "super_customer_id_order",
    "future_churn_state",
    "actual_state",
    "predicted_state",
    "prob_active",
    "prob_at_risk",
    "prob_churned",
    "customer_segment",
    "rx_subtype",
    "inactivity_bucket",
    "frequency_bucket",
    "consistency_bucket",
    "total_orders_observation",
    "days_since_last_order",
    "expected_return_days",
    "gap_cv",
    "total_transaction_value",
    "avg_order_value",
    "is_rx_plus_otc_customer",
    "is_otc_only_customer",
    "is_free_only_customer"
]


lr_label_mapping = get_label_mapping(lr_model)
rf_label_mapping = get_label_mapping(rf_model)
xgb_label_mapping = get_label_mapping(xgb_model)


lr_output = prepare_prediction_output(
    lr_predictions,
    "Logistic Regression",
    lr_label_mapping
).select(*prediction_columns)

rf_output = prepare_prediction_output(
    rf_predictions,
    "Random Forest",
    rf_label_mapping
).select(*prediction_columns)

xgb_output = prepare_prediction_output(
    xgb_predictions,
    "XGBoost",
    xgb_label_mapping
).select(*prediction_columns)


all_model_predictions = (
    lr_output
    .unionByName(rf_output)
    .unionByName(xgb_output)
)


all_model_predictions.write.mode("overwrite").format("delta").save(output_path)

print(f"Saved all model prediction outputs to: {output_path}")
```

```
print("Rows saved:", spark.read.format("delta").load(output_path).count())
```

Python Implementation - Persist Evaluation Outputs

```
from pyspark.sql import functions as F

all_cm = (
    lr_cm
    .unionByName(rf_cm)
    .unionByName(xgb_cm)
    .withColumn("run_id", F.lit(run_id))
)

class_metrics_output = (
    class_metrics_df
    .withColumn("run_id", F.lit(run_id))
)

model_comparison_output = (
    model_comparison_df
    .withColumn("run_id", F.lit(run_id))
)

at_risk_metrics_output = (
    at_risk_metrics_df
    .withColumn("run_id", F.lit(run_id))
)

base_path = "dbfs:/FileStore/eb_thesis"

cm_path = f"{base_path}/churn_confusion_matrices_delta"
class_metrics_path = f"{base_path}/churn_class_metrics_delta"
model_comparison_path = f"{base_path}/churn_model_comparison_delta"
at_risk_path = f"{base_path}/churn_at_risk_metrics_delta"

def sanitize_columns(df):

    cleaned_cols = []

    for c in df.columns:

        clean_c = (
            c.lower()
             .replace(" ", "_")
             .replace("-", "_")
             .replace(".", "")
             .replace("/", "_")
        )

        cleaned_cols.append(clean_c)

    return df.toDF(*cleaned_cols)

all_cm = sanitize_columns(all_cm)

class_metrics_output = sanitize_columns(class_metrics_output)

model_comparison_output = sanitize_columns(model_comparison_output)

at_risk_metrics_output = sanitize_columns(at_risk_metrics_output)

all_cm.write.mode("overwrite").format("delta").save(cm_path)

class_metrics_output.write \
    .mode("overwrite") \
    .format("delta") \
    .save(class_metrics_path)

model_comparison_output.write \
    .mode("overwrite") \
    .format("delta") \
    .save(model_comparison_path)

at_risk_metrics_output.write \
    .mode("overwrite") \
    .format("delta") \
    .save(at_risk_path)
```

```
print("Saved evaluation outputs successfully.")

print("Confusion matrices rows:",
      spark.read.format("delta").load(cm_path).count())

print("Class metrics rows:",
      spark.read.format("delta").load(class_metrics_path).count())

print("Model comparison rows:",
      spark.read.format("delta").load(model_comparison_path).count())

print("AT RISK metrics rows:",
      spark.read.format("delta").load(at_risk_path).count())
```

Python Implementation - Model Interpretability

```
from pyspark.ml import PipelineModel
from pyspark.sql import functions as F

pre_xgb_model = PipelineModel(stages=xgb_model.stages[:-1])
xgb_prepared_df = pre_xgb_model.transform(train_df)

def extract_feature_names_from_metadata(df, features_col="features_tree"):
    metadata = df.schema[features_col].metadata
    attrs = metadata.get("ml_attr", {}).get("attrs", {})

    feature_attrs = []
    for attr_type in ["numeric", "binary", "nominal"]:
        feature_attrs.extend(attrs.get(attr_type, []))

    feature_attrs = sorted(feature_attrs, key=lambda x: x["idx"])
    return [attr["name"] for attr in feature_attrs]

feature_names = extract_feature_names_from_metadata(
    xgb_prepared_df,
    "features_tree"
)

xgb_stage = xgb_model.stages[-1]
booster = xgb_stage.get_booster()

importance_dict = booster.get_score(
    importance_type="gain"
)

importance_rows = [
    (
        feature_names[int(k.replace("f", ""))] if k.startswith("f") and int(k.replace("f", "")) < len(feature_names) else k,
        float(v)
    )
    for k, v in importance_dict.items()
]

importance_named_df = (
    spark.createDataFrame(importance_rows, ["feature_name", "importance"])
    .orderBy(F.desc("importance"))
)

display(importance_named_df)
```

Validation and Governance Notes

The predictive modeling implementation follows a leakage-safe temporal structure. Observation-window data is used only for feature construction, while prediction-window data is used only for target assignment. Customers with insufficient behavioral history are assigned to UNDEFINED during target construction and excluded from supervised model training.

The final workflow combines SQL-based behavioral feature engineering with distributed Spark ML model training and Delta-based artifact persistence. This design supports reproducibility, scalability, and enterprise applicability, while keeping the predictive modeling layer aligned with the behavioral churn interpretation framework developed throughout the thesis.

Technical Stack and Libraries Used

The predictive modeling implementation was developed using Python within the Databricks environment and relied on distributed Spark-based processing for large-scale behavioral feature engineering and multiclass churn-state prediction. The following libraries and frameworks were used throughout the implementation.

Library / Framework	Main Components	Purpose
PySpark SQL	functions as F	Distributed DataFrame transformations and aggregations
Spark ML	Pipeline, StringIndexer, OneHotEncoder, VectorAssembler, Imputer, StandardScaler	Feature preprocessing and ML pipeline construction
Spark ML Classification	LogisticRegression, RandomForestClassifier	Multiclass predictive modeling
XGBoost Spark	SparkXGBClassifier	Distributed gradient-boosted ensemble modeling
PySpark ML Functions	vector_to_array	Probability vector conversion and prediction interpretation
Delta Lake	.write.format("delta")	Persistence of predictions and evaluation artifacts
DBFS	dbfs:/FileStore/...	Distributed artifact storage within Databricks