HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

Capstone project report

Customer Growth & Retention (E-commerce)

Submitted by

Nguyễn Nhật Minh - 20225510

Trần Nhật Minh - 20225511

Trần Nam Tuấn Vượng - 20225540

Đới Sỹ Thắng - 20225528

Guided by

Assoc. Prof. Nguyễn Bình Minh

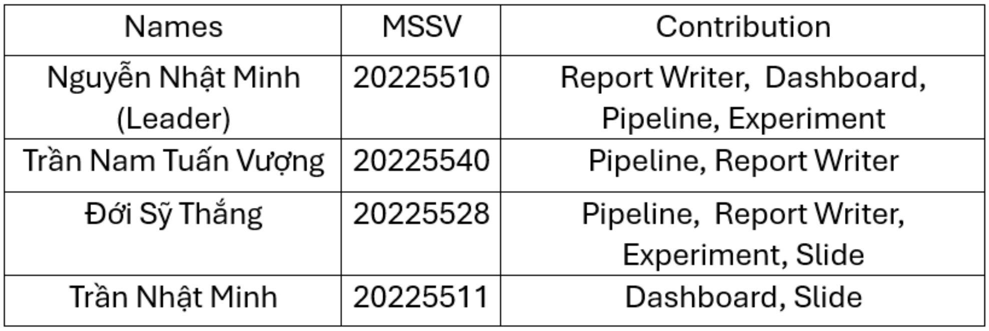
Hanoi, December 2025

Abstract

Customer growth and retentions has been central challenges for organization seeking to optimize marketing efficiently and long-term profitability, regardless of being in the era of unprecedented AI advancements. The project proposes an integrated analytic pipeline combining Customer Segmentation, Predictive CLV, Churn Prediction and Experiment based on Policy assumptions to generate actionable insights and personalized interventions. Firstly, segmentation techniques are applied to identify the heterogeneous customer groups based on customer behaviors(RFM features), thus predictive CLV models estimate the future contribution of each customer from each group. Churn prediction and survival analysis module is then utilized to detect at-risk users, finally, in decision making, Monte Carlo experiment quantify the incremental impact of assumption-based policy simulation, allowing the design of targeted strategies that maximize conversion and retention lift. Together, this end-to-end pipeline provide robust framework for the problem of customer growth. All code is available at code

Contents

Part I
Introduction



Names	MSSV	Contribution
Nguyễn Nhật Minh (Leader)	20225510	Report Writer, Dashboard, Pipeline, Experiment
Trần Nam Tuấn Vượng	20225540	Pipeline, Report Writer
Đới Sỹ Thắng	20225528	Pipeline, Report Writer, Experiment, Slide
Trần Nhật Minh	20225511	Dashboard, Slide

Figure 1 . Contribution of each member

1 Problem formulation

Business operates in the era of accelerated AI-driven transformations, where customer growths have become critical determinants of long-term competitiveness. Despite having access to rich and enormous interaction, transaction and engagement data, translating this into actionable strategies is also a difficult problem. Key strategic questions remain unanswered: Which marketing interventions truly cause positive behavioral change? Many traditional approaches such as rule-based segmentation, aggregate reporting can not provide the reliable and sufficient insight needed for the effective promotion rule. This project addresses the need for a unified data-driven framework capable of supporting end-to-end customer growth and retention decision-making. All code is available at code[1]

2 General Approach

The central problem addressed in this study is the absence of an integrated, data-driven framework that can simultaneously:

- **Customer Segmentation:** Identify meaningful customer segments based on behavior and value potential.
- **Predictive CLV:** Predict future customer lifetime value (CLV) to inform resource allocation and prioritization.
- **Churn Prediction and Survival Analysis:** Detect early signals of churn risk to enable proactive retention strategies.
- **Monte Carlo experiment with policy assumptions :** Measure the assumed policy of marketing actions and determine which customers benefit from specific treatments through Monte Carlo simulation.

[1] `https://github.com/Helooeverybody/Ecommerce-CustomerAnalytics`

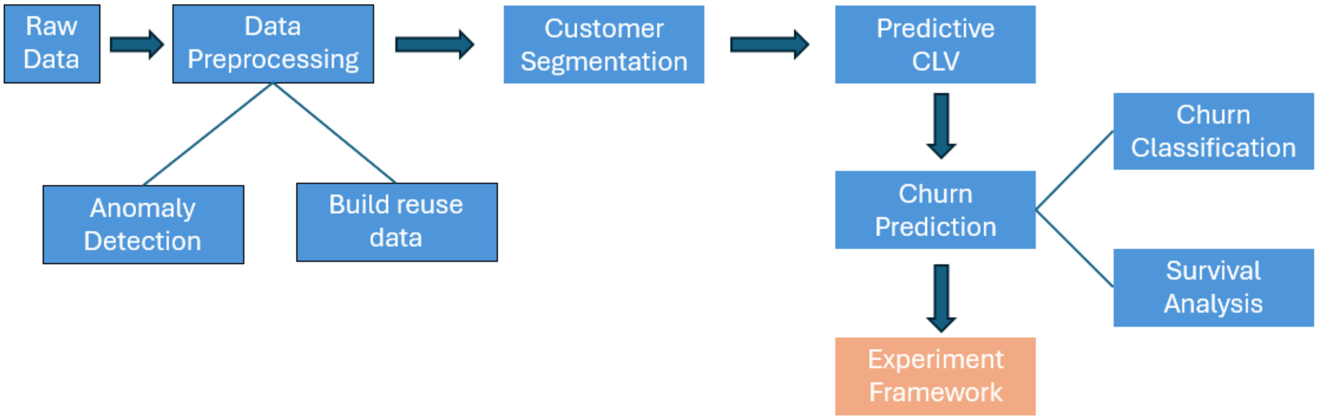


Figure 2 . General pipeline

Part II

Data

3 Data Description

The dataset is composed of three separate files: a behavioral log (`events.csv`), a file containing item attributes (`item_properties.csv`), and a file describing the product category hierarchy (`category_tree.csv`). The data was collected from a real-world e-commerce platform and is provided in its raw form, meaning that no content-level preprocessing or transformations were applied. Due to confidentiality constraints, all recorded values are anonymized using hashing techniques.

The behavioral dataset captures user interactions over a period of approximately 4.5 months. Each visitor can generate three types of events: *view*, *addtocart*, and *transaction*. In total, the dataset contains 2,756,101 events, including 2,664,312 views, 69,332 add-to-cart actions, and 22,457 transactions, generated by 1,407,580 distinct visitors. For roughly 90% of these events, corresponding item attributes are available in the `item_properties.csv` file.

The item properties file (`item_properties.csv`) consists of 20,275,902 records describing 417,053 unique items. Owing to file size constraints, this data is split into two separate files. Since item attributes may change over time (for example, price variations), each record is associated with a timestamp. Consequently, the file can be interpreted as a sequence of weekly snapshots aligned with the observation period of the behavioral data. If a particular item property remains unchanged throughout the entire period, only a single snapshot entry is stored.

Limitation of data This dataset was originally created for the task of implicit recommendation in machine learning. Therefore, the dataset is relatively sparse and limited in scope. Furthermore, price attribute is not explicitly provided and must be inferred. In addition, the data only covers a four-month period, which is relatively short and makes it challenging to capture and evaluate user behavior, hence , there are limited opportunities for extensive feature engineering and pipeline design. Nevertheless, we aim to make the best possible use of the available data and exploit all informative insights presented in the dataset.

4 Data Understanding

4.1 Overview

The chart 3 shows that while the market attracts a stream of visitors with a weekly pattern and a peak in late July, most users are just ”window shopping” rather than buying. The funnel analysis reveals a massive drop in engagement: out of over 2.6 million product views, only 2.6% of users actually added an item to their cart, and less than 1% completed a purchase. This show that it fails to convince visitors to take action, making the step from viewing to adding to cart.

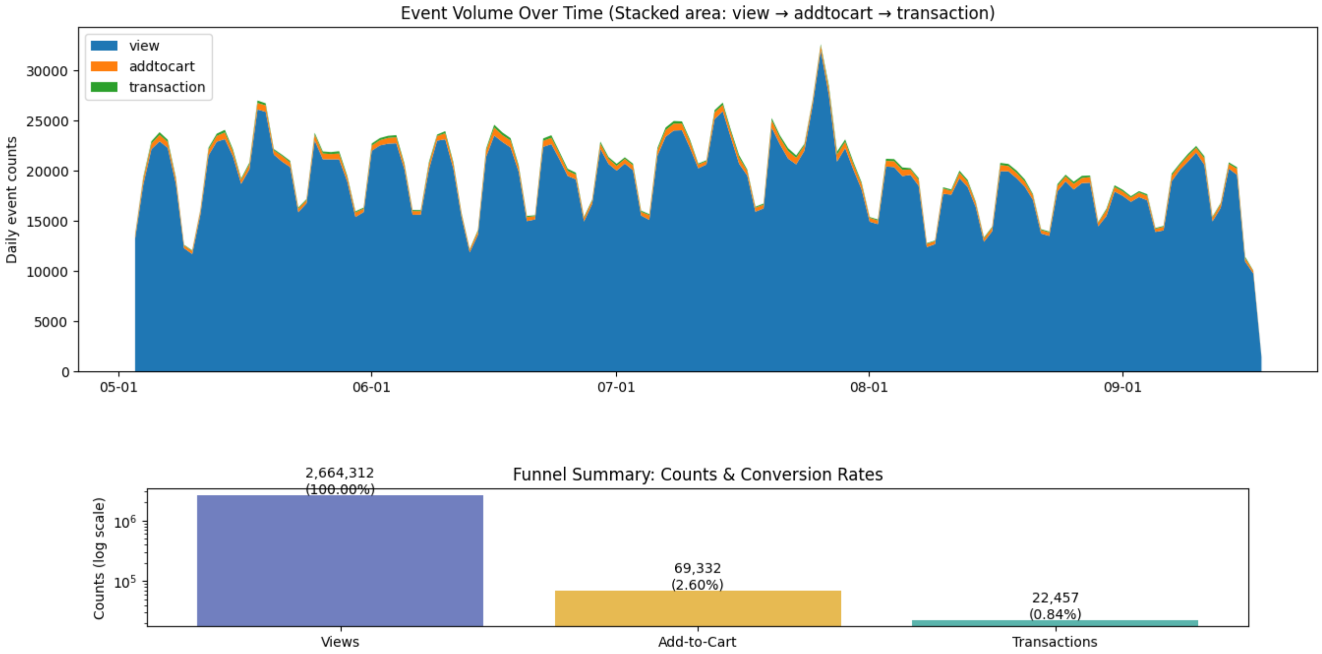


Figure 3 . Event Volume and Funnel Summary

The chart 4 reveal a weekly trend in user activity, showing that engagement follows a ”up and down” behaviour rather than a straight growth line. While daily traffic is mostly steady with a single large spike in late July, the hourly data uncovers a strong preference for nighttime usage. Activity is lowest during morning work hours (08:00–11:00) and peaks in the evening around 8:00 PM, suggesting that users view the platform as a leisure activity to enjoy during their free time rather than during the business day.

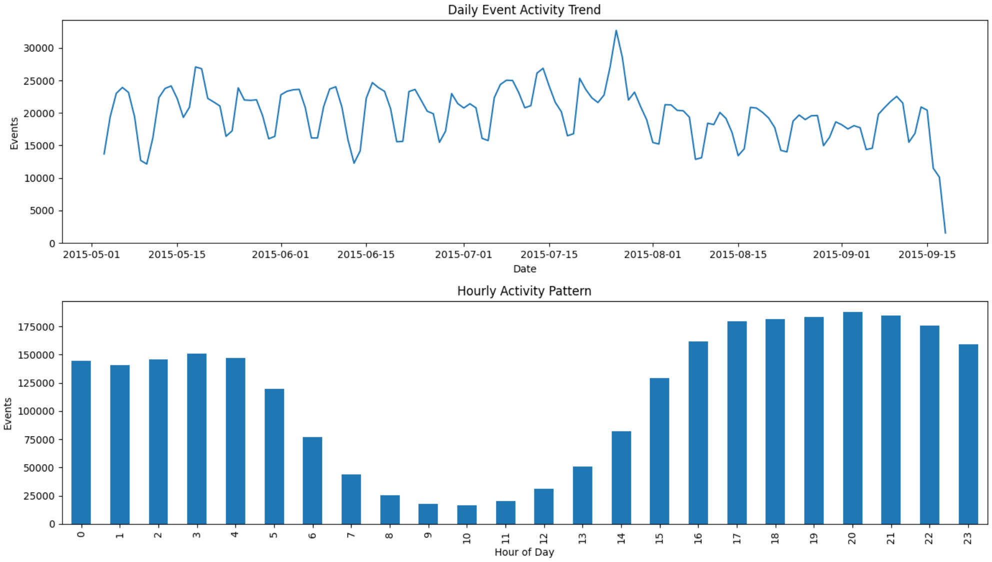


Figure 4 . Daily event activity trend and Hour activity pattern

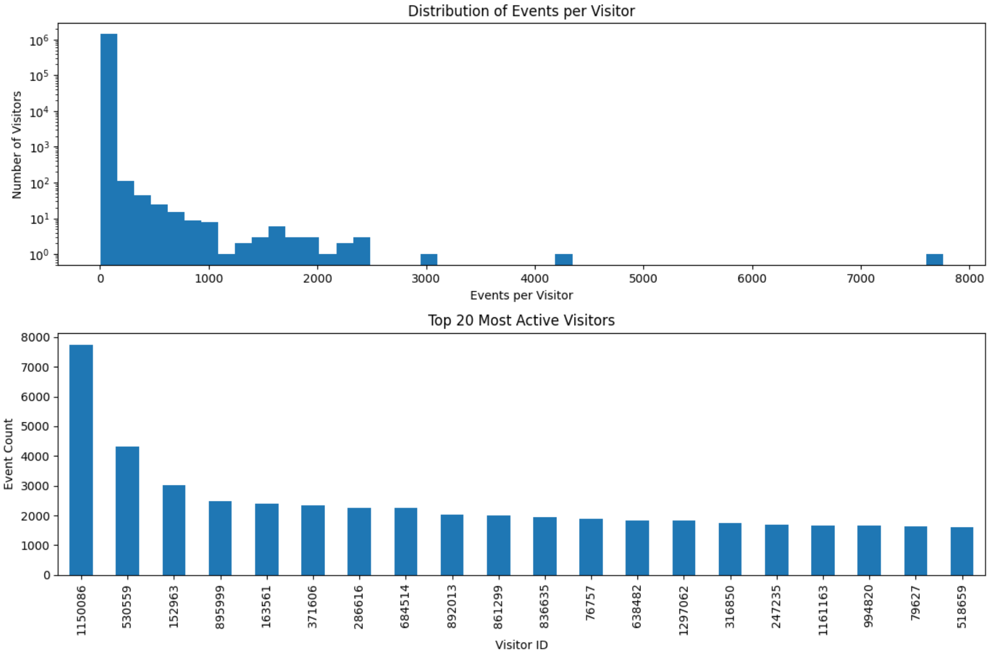


Figure 5 . Overview of events per visitor

The chart 5 reveals a uneven pattern in activity, showing that a very small group of users and products drives the majority of engagement. The charts indicate that while millions of visitors interact very little with the site, ”super-users” are extremely active, with the top visitor generating nearly 8,000 events alone. A similar trend applies to the products in the chart 6: while most items receive almost no attention, a few ”bestsellers” generate thousands of interactions. This suggests the platform relies heavily on this small minority of active users and hit products rather than an even spread of activity across the board.

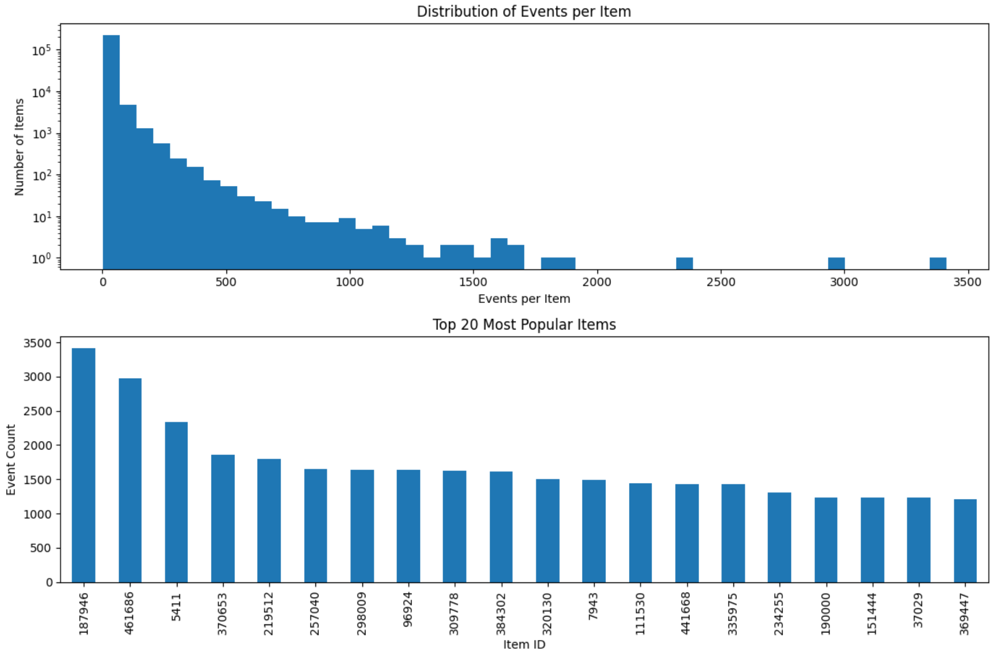


Figure 6 . Overview of events per item

4.2 Transaction Overview

The chart 7 compares the daily number of transactions against the average amount spent per order. The transaction volume (blue line) shows a repetitive "up and down" trend due to weekly cycles, reaching a peak of 216 sales in late June, while the overall trend remains relatively stable between 100 and 150 daily sales. However, a distinct shift occurs in September, while the total number of transactions begins to decrease (blue line drops), the average value of each transaction (orange line) rises significantly to over 200k. This suggests that although the platform processed fewer orders at the end of the observed period, the remaining customers were making much larger purchases.

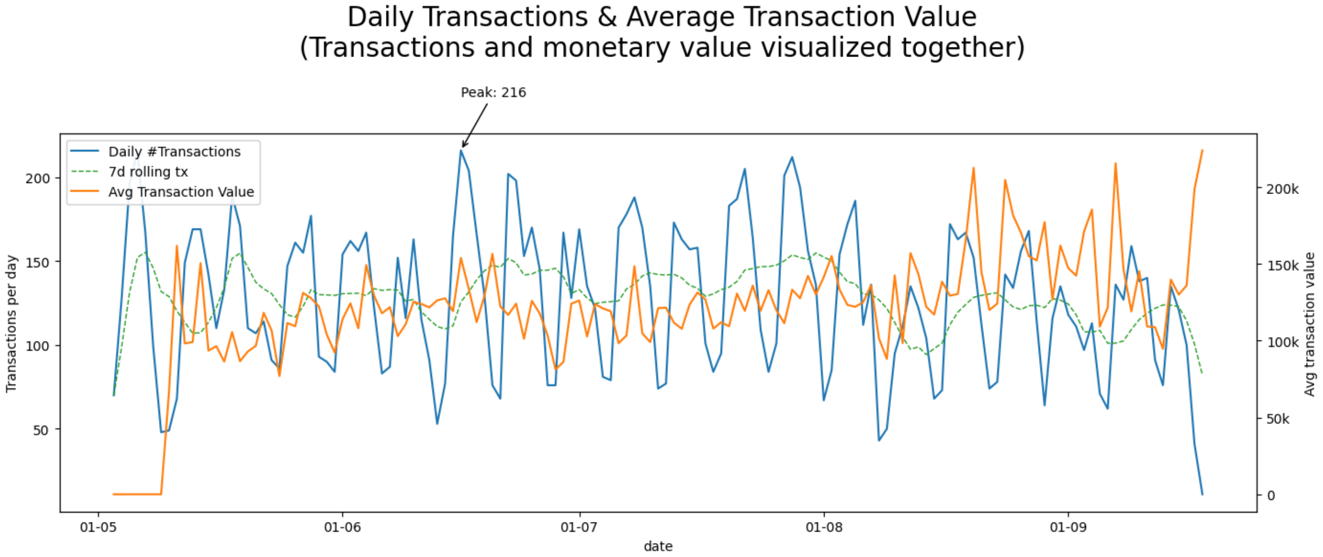


Figure 7 . Daily transactions and average transaction value

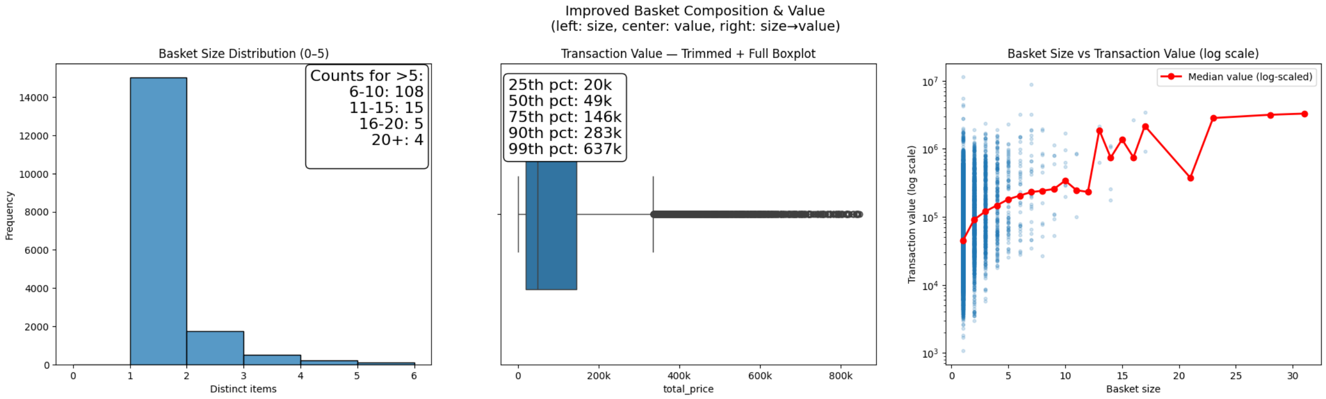


Figure 8 . Basket composition and Value

The data 8 shows that the vast majority of customers buy only one item at a time, making single-item transactions the most common behavior by far. Because most orders are so small, the typical amount spent is relatively low—usually around 49k—though there are rare cases where customers spend significantly more. While buying more items naturally leads to a higher total bill, these multi-item purchases happen so infrequently that the business currently relies on a high volume of small, individual sales rather than large, bulk orders.

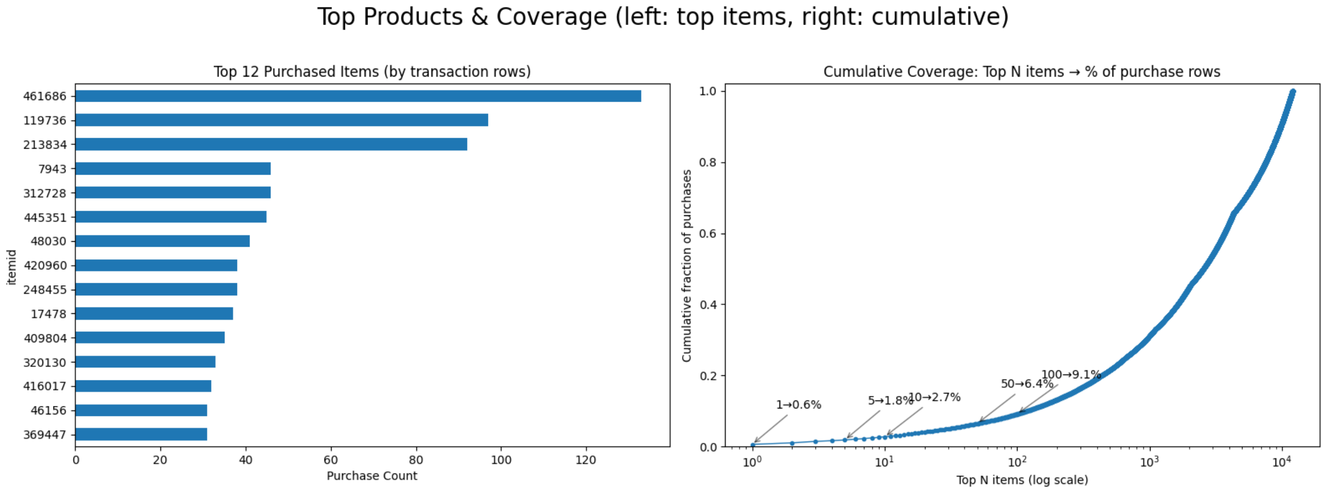


Figure 9 . Top products and coverage

The analysis of product sales in Figure 9 shows that the business does not rely on just a few ”hit” products; instead, sales are spread out across a massive variety of items. While the bar chart highlights one standout product (ID 461686) with over 130 purchases, the cumulative chart reveals that even the top 100 most popular items combined only make up about 9% of all sales. This indicates that the vast majority of revenue comes from selling small quantities of thousands of different unique items, rather than depending on a small group of bestsellers.

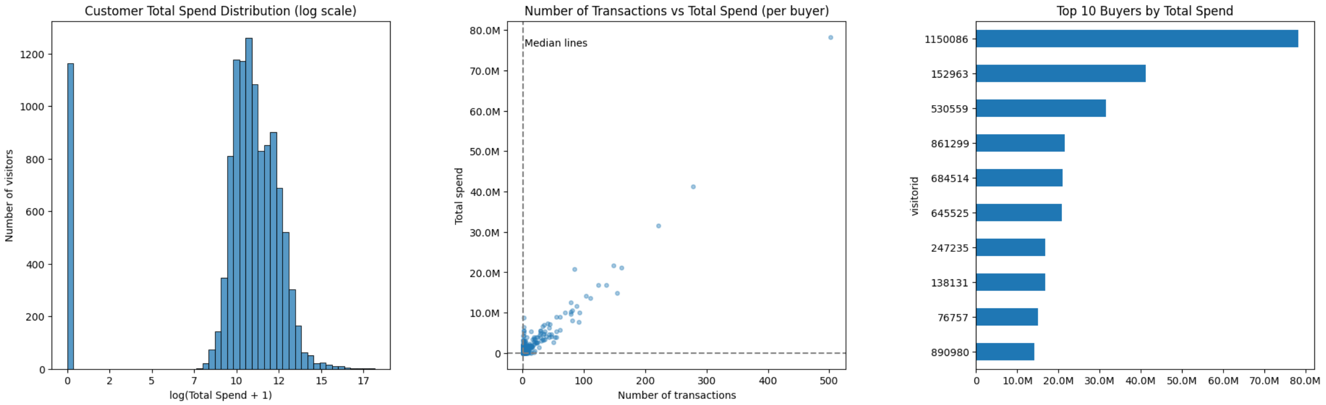


Figure 10 . Overview of customer spend

The figure 10 reveals a highly unequal spending pattern. It is clear that while the vast majority of visitors spend nothing or very little, there is a strong connection between frequency and value: customers who transact more often end up spending significantly more. Most notably, the data identifies extreme outliers, such as a single top buyer who spent nearly 80 million, which is far higher than even the other top ten customers.

5 Data Preprocessing

5.1 General Preprocessing

Before any modelling or analysis can be performed, these raw data sources must be cleaned, standardised, and transformed into a structured format suitable for generating analytical features. The overall preparation consists of timestamp conversion, anomaly removal, item property attachment, and finally, user-level feature construction.

The first stage focuses on converting the timestamp values into a unified and readable datetime format. The original event files store timestamps in Unix millisecond form. These timestamps were therefore transformed into standard calendar time to enable chronological sorting and time-based feature computation. During this step, invalid or malformed timestamps were identified and removed, ensuring that only reliable event records were kept for analysis.

The next step involves predicting the price attribute from the set of hashed item properties. From the table 1, we can strongly assert that property 790 is the price attribute, as 100% of its values are positive numerical values. The fact that it has fewer unique values compared to other properties is reasonable. In practice, prices are often set at appealing levels, such as values ending in 9, 5, or 0. Moreover, retail markets typically focus on a limited range of price points, making a smaller number of unique price values both normal and expected. This characteristic is more convincing than other properties, such as property 917, which contains as many as 84,000 unique values. In short, the main factor supporting the identification of property 790 as the price attribute is that all of its values are strictly positive.

Table 1 . Top Numeric Property Candidates for Potential Price Features

Property ID	Rows	Unique Values	Positive Numeric Ratio	Unique Value Ratio
917	91,083	84,043	0.856	0.923
790	1,790,516	33,052	**1.000**	0.018
888	128,149	26,780	0.359	0.209
202	29,276	19,101	0.729	0.652
575	14,799	14,746	0.761	0.996
810	90,954	4,553	0.429	0.050
761	30,416	2,012	0.005	0.066

After detecting price property, an additional quality filtering step was applied to remove outlier visitors.(The section below for detailed).

Once the event dataset was cleaned, the next stage attached product information (price, availability, category) to every user action. Because product attributes changed over time, a historical matching approach was used. Each event was linked to the closest valid product snapshot recorded before that event. This method allowed price and category information to reflect the actual product state at the moment of interaction.

For the preparation of churn prediction, the dataset was then aggregated into a user-level structure. All events recorded up to a selected cutoff time were grouped by customer. From these grouped records, a series of user features were engineered to describe behavioural patterns and interaction characteristics. Temporal features such as first event time, last event time, recency, and tenure length were calculated as follows:

$$\text{recency}_u = (T_c - t_{u,\text{last}}), \qquad \text{tenure}_u = (t_{u,\text{last}} - t_{u,\text{first}} + 1), \tag{1}$$

where T_c is the cutoff time(or observed time), $t_{u,\text{last}}$ is the most recent event timestamp, and $t_{u,\text{first}}$ is the earliest recorded interaction for user u.

Behavioural features were then computed by counting the number of each event type (view, add-to-cart, and transaction) for each user. Proportional ratios such as add-to-cart rate or transaction rate were also introduced to capture differences in conversion behaviour. Short-term interaction intensity was described using rolling event windows (7-day, 14-day, and 30-day periods).

5.2 Anomaly detection

5.2.1 Theoretical Background- Isolation Forest

Anomaly detection aims to identify observations that are significantly different from expected data behavior. Traditional approaches typically rely on statistical or density-based models, defining anomalies as low-probability events under an assumed distribution. However, such methods often struggle with high-dimensional, non-linear, or structurally complex data, motivating recent approaches that use structural properties rather than strong distributional assumptions.

The Isolation Forest algorithm [6] is based on the principle that anomalies are more easily isolated than normal observations. Rather than explicitly modeling normal behavior, it constructs an ensemble of randomly generated isolation trees that recursively partition the feature space. At each node, a feature and a split value are selected at random within the feature's range. Since anomalies are rare and typically lie in sparse regions of the data space, they tend to be isolated in fewer splits than normal observations.

Overall, the design of Isolation Forest provides an efficient solution for anomaly detection. The method scales well in high-dimensional environments because it avoids expensive distance or density calculations, and its randomised partitioning process enables the model to operate with a time complexity of $\mathcal{O}(n \log n)$, making it suitable for large datasets. More importantly, the algorithm does not depend on strong distributional assumptions, allowing it to adapt to a wide range of domains.

5.2.2 Implementation Overview

The implementation to detect abnormal browsing and purchasing behaviour consists of both rule-based heuristics and model-based Isolation Forest predictions. The workflow operates on the raw event logs and produces a final binary flag indicating whether a user is considered anomalous.

Event-Level Feature Engineering To capture behavioural patterns at the event granularity, we first construct a feature augmentation function. Given the raw event dataset containing visitor identifiers, timestamps, item identifiers, and event types, we generate additional fields associated with event repetition speed and item interaction frequency.

Each timestamp field is converted to a unified datetime format, and all records are sorted by user, timestamp, and item. We compute the inter-event time difference for each visitor as

$$\Delta t_i = t_i - t_{i-1},$$

measured in seconds. A binary variable is then assigned to highlight extremely rapid interactions:

$$\text{very\_fast}_i = \begin{cases} 1, & \Delta t_i \leq 1 \text{ second}, \\ 0, & \text{otherwise}. \end{cases}$$

This helps detect automated or bot-like behaviours characterised by machine-speed clicks.

We also examine repetitive interactions with identical items. For each user, we compare the current event item with the previous one and set

$$\text{same\_item\_repeat}_i = \begin{cases} 1, & \text{item}_i = \text{item}_{i-1}, \\ 0, & \text{otherwise}. \end{cases}$$

To estimate user activity bursts, we compute the number of events per minute by grouping timestamps to minute windows. Let $N_{u,m}$ represent the number of events generated by user u within minute m. This becomes an important intensity feature, allowing us to later measure abnormal session burst patterns.

User-Level Feature Aggregation Event-level features are aggregated at the user level to form numerical descriptors suitable for anomaly detection models. For each user u, we compute:

- Proportion of extremely fast interaction events,

$$p_{\text{fast}}(u) = \frac{1}{n_u} \sum_{i=1}^{n_u} \text{very\_fast}_i,$$

- Proportion of repeated item interactions,

$$p_{\text{repeat}}(u) = \frac{1}{n_u} \sum_{i=1}^{n_u} \text{same\_item\_repeat}_i,$$

- Maximum events per minute,

$$\max_m(N_{u,m}),$$

- Statistics of inter-event time distribution:

$$\min(\Delta t_i), \quad \text{median}(\Delta t_i), \quad \sigma(\Delta t_i).$$

These engineered attributes enhance the representation of user behaviour and provide detailed structural information beyond simple event counts.

Rule-Based Detection Module In addition to model-based anomaly detection, we incorporate a deterministic rule-based module designed to flag users exhibiting extreme or implausible behavioural patterns. This module operates on aggregated user-level features and serves as a transparent and interpretable screening mechanism.

Let u denote a user and let the following aggregated features be available: total number of events, number of events in the last 30 days, number of active days, maximum event intensity per minute, proportion of extremely fast events, proportion of repeated item interactions, and number of distinct items interacted with.

A set of binary rule indicators is defined by thresholding these features. Specifically, we introduce the following flags:

$$\begin{aligned}
\text{flag}_{\text{events}}(u) &= I\left(\text{total\_events}(u) \geq \tau_{\text{events}}\right), \tau_{\text{events}} = 6000, && (2)\\
\text{flag}_{\text{events30d}}(u) &= I\left(\text{events}_{30d}(u) \geq \tau_{\text{events30d}}\right), \tau_{\text{events30d}} = 1000 && (3)\\
\text{flag}_{\text{epd}}(u) &= I\left(\text{events\_per\_day}(u) \geq \tau_{\text{epd}}\right), \tau_{\text{epd}} = 500 && (4)\\
\text{flag}_{\text{epm}}(u) &= I(\max_m N_{u,m} \geq \tau_{\text{epm}}), \tau_{\text{epm}} = 50 && (5)\\
\text{flag}_{\text{fast}}(u) &= I\left(p_{\text{fast}}(u) \geq \tau_{\text{fast}}\right), \tau_{\text{fast}} = 0.5 && (6)\\
\text{flag}_{\text{repeat}}(u) &= I\left(p_{\text{repeat}}(u) \geq \tau_{\text{repeat}}\right), \tau_{\text{repeat}} = 0.5 && (7)\\
\text{flag}_{\text{distinct}}(u) &= I\left(\text{distinct\_items}(u) \geq \tau_{\text{distinct}}\right), \tau_{\text{distinct}} = 3000 && (8)
\end{aligned}$$

where $I(\cdot)$ denotes the indicator function and $\tau_\cdot$ are predefined thresholds.

The thresholds are chosen to represent extreme behaviour, including unusually high interaction volume, intense short-term activity bursts, excessive interaction speed, and repetitive or overly broad item exploration. Such patterns are unlikely to arise from normal human browsing behaviour and are commonly associated with automated agents or data integrity issues.

The individual rule indicators are aggregated into a single rule-based anomaly score:

$$\text{RuleScore}(u) = \sum_k \text{flag}_k(u), \tag{9}$$

where k indexes all rule conditions. Higher scores correspond to users that violate multiple behavioural constraints simultaneously.

This rule-based score is not intended to provide a probabilistic anomaly estimate but rather to offer an interpretable measure of behavioural extremeness. In the overall detection framework, it can be used independently for heuristic flagging or jointly with model-based anomaly scores for validation and diagnostic analysis.

Model-Based Detection Module To complement the rule-based heuristics, a model-based anomaly detection module is employed to capture multivariate and non-linear interaction patterns that are difficult to express through explicit thresholds. This module is based on the Isolation Forest algorithm and operates on a selected set of aggregated user-level features.

Feature Selection and Preprocessing Let u denote a user. A feature vector $\mathbf{x}_u \in R^d$ is constructed from the following behavioural attributes:

- Number of events in the last 7 days,
- Number of events in the last 30 days,
- Proportion of add-to-cart events,
- Proportion of transaction events,
- Average number of events per active day.

Only features available in the dataset are included in the model input. Missing values are imputed with zeros to ensure a complete feature matrix.

Isolation Forest Training An Isolation Forest model is trained on the scaled feature matrix using an ensemble of T isolation trees. Each tree is constructed by recursively partitioning the feature space via random feature selection and random split values. The model is parameterised with a fixed contamination rate, reflecting the assumption that anomalies constitute a very small fraction of the population.

The model is trained in an unsupervised manner on the full dataset, without requiring labeled anomalies.

Anomaly Scoring and Prediction For each user u, the trained model produces an anomaly score s_u, defined as the Isolation Forest decision function output. Lower values of s_u correspond to shorter average path lengths and thus higher anomaly likelihood.

In addition, the model produces a binary prediction:

$$\text{iso\_flag}(u) = \begin{cases} 1, & \text{if user } u \text{ is classified as anomalous,} \\ 0, & \text{otherwise.} \end{cases}$$

To facilitate interpretability and downstream combination with other detection signals, the raw anomaly scores are linearly normalised to the unit interval:

$$s_u^{\text{norm}} = \begin{cases} 0, & \text{if } \max(s) = \min(s), \\ 1 - \dfrac{s_u - \min(s)}{\max(s) - \min(s)}, & \text{otherwise,} \end{cases}$$

where the inversion ensures that higher values indicate greater anomalousness.

A final model-based anomaly indicator is obtained by thresholding the normalised score:

$$\text{model\_flag}(u) = I\left(s_u^{\text{norm}} \geq \tau_{\text{model}}\right),$$

where τ_{model} is a predefined decision threshold.

Hybrid Scoring and Final Output The final anomaly label combines the rule-based and model-based results. Let

$$S_{\text{comb}}(u) = \alpha S_{\text{rule}}(u) + (1 - \alpha) S_{\text{model}}(u),$$

where α is a weighting coefficient(we choose 0.6 in this project). Users are flagged as anomalous if

$$S_{\text{comb}}(u) \geq 0.5.$$

This hybrid decision logic leverages the interpretability of rules and the flexibility of a machine learning model, producing a robust and transparent anomaly detection framework.

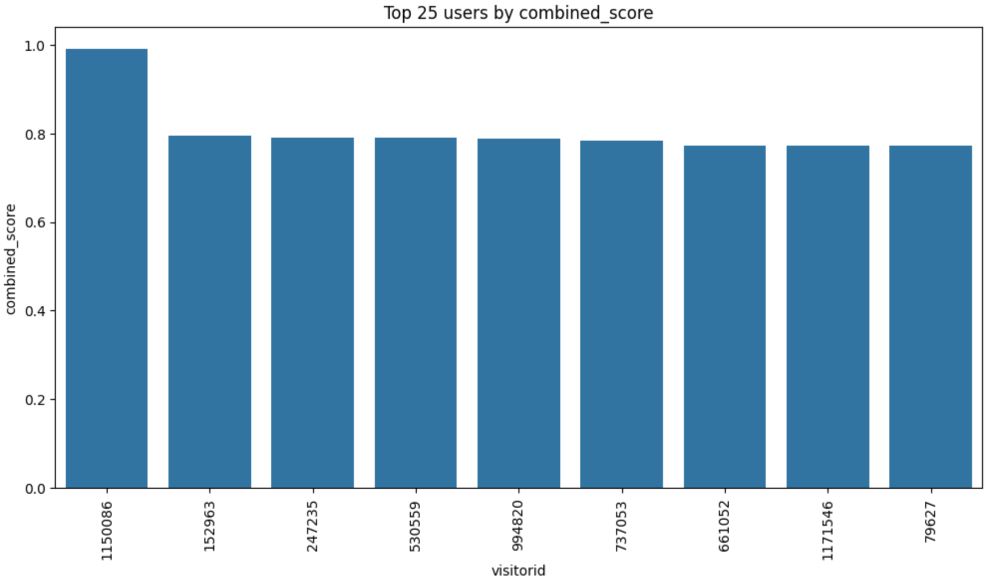


Figure 11 . Top anomalous users

The figure 11 shows that a large number of visitors achieve a combined score of approximately 0.8; however, only a single visitor reaches the maximum score of 1(**id: 1150086**). Given the limited size of our dataset, this outlier is excluded from subsequent analysis. And the profile of the visitor is integrated in the table 2.

Table 2 . Key Anomaly-Indicative Features for Flagged Visitor

Feature	**Value**
Visitor ID	**1150086**
Total events	7,700
Events in last 30 days	1,184
Events in last 7 days	233
Active days	74
Events per active day	104.05
Distinct items interacted	3,814
Proportion of add-to-cart events	0.093
Proportion of transaction events	0.073
Maximum events per minute	10
Proportion of repeated item interactions	0.271
Rule-based anomaly score	3
Isolation Forest score (normalised)	0.978
Combined anomaly score	0.991
Final anomaly flag	1

Part III
Approaches

6 Customer Segmentation

6.1 Theoretical Background

6.1.1 RFM Modeling

RFM (Recency–Frequency–Monetary)[1] analysis is based on the principle that a customer's past behaviour is a strong indicator of their future behaviour. Each component captures a distinct aspect of customer activity.

Let a customer i have a transaction history:

$$T_i = \{(t_{i1}, v_{i1}), (t_{i2}, v_{i2}), \dots, (t_{in_i}, v_{in_i})\},$$

where t_{ik} is the timestamp of transaction k and v_{ik} is the monetary value of the k-th transaction.

Let t^* be the reference date (observed date).

Recency (R) : Recency measures how much time has elapsed since the customer's last transaction:

$$R_i = t^* - \max_k t_{ik}.$$

A smaller R_i indicates that the customer has interacted more recently, which is empirically associated with higher engagement levels.

Frequency (F) : Frequency quantifies how often the customer purchases within the observation window:

$$F_i = n_i,$$

where $n_i = |T_i|$ is the total number of recorded transactions. Higher purchase frequency often indicates brand loyalty and sustained engagement.

Monetary Value (M) : Monetary value represents the aggregate or average amount a customer spends:

$$M_i = \frac{1}{n_i} \sum_{k=1}^{n_i} v_{ik}.$$

Alternatively, the total monetary value can be used:

$$M_i^{\text{total}} = \sum_{k=1}^{n_i} v_{ik}.$$

RFM as a Behavioral Feature Vector The RFM attributes combine into a behaviour vector for each customer:

$$\mathbf{r}_i = (R_i, F_i, M_i) \in R^3.$$

This vector is often normalized or transformed through quantile-based scoring to mitigate skewness.The theoretical grounding for RFM rests on empirical findings from direct marketing and behavioral economics:

- **Recency Effect**: Customers who purchased recently are more likely to purchase again, due to active engagement cycles.
- **Repeat Behaviour Principle**: Higher purchase frequency implies greater brand preference.
- **Economic Value Principle**: Higher monetary value indicates stronger profitability potential.

Additionally, RFM serves as a low-dimensional summary statistic approximating latent customer engagement intensity.

6.1.2 Clustering Algorithm

A clustering algorithm is applied based on RFM features to segment customers into distinct groups, which then enables the design of tailored policies for each customer segment.

K-means Clustering Given a dataset $\{x_i\}_{i=1}^N$ with $x_i \in R^d$, K-means aims to partition the data into K clusters by minimizing the within-cluster sum of squared distances. The objective function is defined as:

$$\min_{\{\mu_k\}_{k=1}^K} \sum_{i=1}^{N} \min_{k \in \{1,\ldots,K\}} \|x_i - \mu_k\|^2,$$

where $\mu_k \in R^d$ denotes the centroid of cluster k.

The algorithm proceeds iteratively by alternating between assigning each data point to the nearest centroid and updating centroids as the mean of assigned points. K-means produces a hard assignment, such that each observation belongs to exactly one cluster.

Gaussian Mixture Models Gaussian Mixture Models assume that the data are generated from a mixture of K Gaussian components. The probability density function of a GMM is given by:

$$p(x_i) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x_i \mid \mu_k, \Sigma_k),$$

where π_k are mixture weights satisfying $\sum_{k=1}^{K} \pi_k = 1$, and $\mathcal{N}(x_i \mid \mu_k, \Sigma_k)$ denotes a multivariate Gaussian distribution with mean μ_k and covariance matrix Σ_k.

Model parameters are typically estimated using the Expectation–Maximization (EM) algorithm, which maximizes the log-likelihood:

$$\mathcal{L} = \sum_{i=1}^{N} \log(\sum_{k=1}^{K} \pi_k \mathcal{N}(x_i \mid \mu_k, \Sigma_k)).$$

Unlike K-means, GMMs yield soft assignments through posterior probabilities

$$\gamma_{ik} = P(z_i = k \mid x_i),$$

which quantify the degree of membership of observation i in cluster k.

6.2 Implementation Overview

RFM Feature Construction. The raw transaction data may contain multiple item-level records within the same transaction. Therefore, transactions are first aggregated at the transaction level by grouping on `transactionid`. For each transaction, item identifiers are aggregated and monetary values are summed to obtain the total transaction amount.

RFM features are then constructed using the transaction-level data only. For each customer i, *Recency* is defined as the time elapsed since the most recent transaction, *Frequency* as the total number of transactions, and *Monetary* as the average transaction value over the observation period.

To ensure reliable behavioural representation, only customers with at least two observed transactions are retained in the dataset. This filtering step reduces noise from one-time purchasers and allows the RFM features to better capture stable purchasing patterns in the future pipeline.

Prior to clustering, all behavioral features are standardized using z-score normalization.

Clustering Model Configuration In this project, both K-Means and Gaussian Mixture Model (GMM) algorithms are experimentally evaluated. The number of clusters k is varied from 2 to 10, and the Silhouette Score is used as the evaluation metric to determine the optimal clustering performance. The resulting scores are summarized in the table 3.

Table 3 . Silhouette Scores for GMM and KMeans Across Different Numbers of Clusters

Number of Clusters (k)	GMM	KMeans
2	0.227	0.518
3	0.119	0.348
4	0.004	0.396
5	0.164	0.320
6	-0.032	0.321
7	0.015	0.328
8	0.070	0.338
9	0.045	0.339
10	0.052	0.320

It is evident from this metric that K-Means significantly outperforms GMM, and that using fewer clusters yields better results, with the Silhouette Score peaking at $K = 2$ for K-Means. However, in practice, model selection should not rely solely on quantitative metrics. The choice of the number of clusters should be guided by the intended business objectives. The Silhouette Score primarily reflects how well clusters are separated and how efficiently the data is partitioned, rather than the practical usefulness of the segmentation.

In fact, after clustering our customers, we assign the label for each to get the meaningful information for the future pipeline. To translate statistical segment profiles into actionable business categories, a deterministic labeling function is applied to each segment. The labeling is based on interpretable thresholds over recency, frequency, and monetary value.

Let $Q_{0.7}$ denote the 70th percentile of the segment-level average monetary value distribution $\{\overline{M}_s\}_{s=1}^{K}$. Each segment s is assigned a business label according to the following rules:

- **One-time Buyers**: $\overline{F}_s = 1$.
- **Champions**: $\overline{R}_s < 30$, $\overline{F}_s > 5$, and $\overline{M}_s > Q_{0.7}$.
- **Loyal Customers**: $\overline{R}_s < 60$ and $\overline{F}_s > 4$.
- **At Risk**: $\overline{R}_s > 65$ and $\overline{F}_s \leq 2$.
- **Potential Loyalists**: all remaining segments.

After extensive experimentation, we found that using a Gaussian Mixture Model (GMM) with $K = 4$ provides the most effective differentiation for our intended use case, and the profile of each segment is in table4. We merge the cluster 1 and 4 to the same group, which is assigned as the label: Potential Loyalists, then we get 3 clusters in total.

Table 4 . Customer Segment Profiles Derived from GMM Clustering

Customers	Avg. R	Avg. F	Avg. M	Avg. Purchase Rate	Revenue Share	Business Label
92	82.51	31.12	149,739	0.358	0.417	Potential Loyalists
80	37.29	19.74	182,741	0.686	0.276	Loyal Customers
730	69.32	2.00	158,476	0.071	0.232	At Risk
129	61.28	3.00	191,648	0.117	0.074	Potential Loyalists

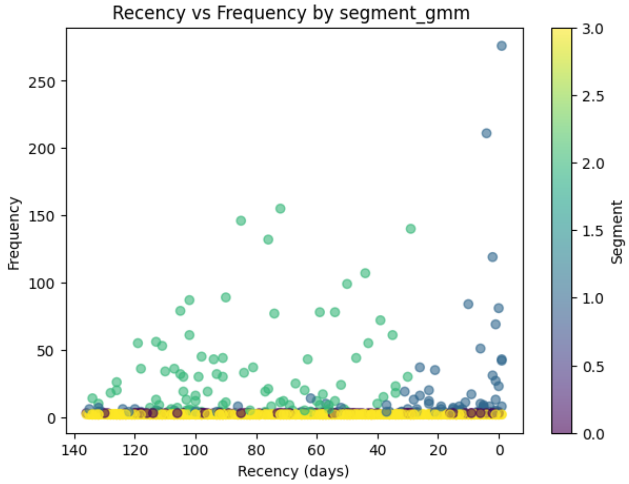


Figure 12

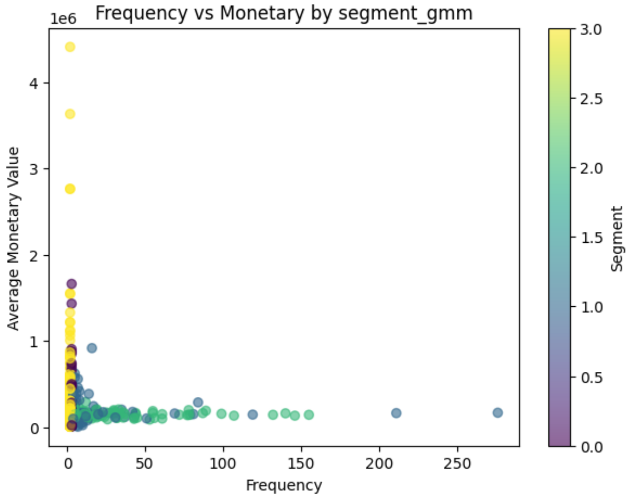


Figure 13

7 CLV Estimation

In customer-base analysis, probabilistic models offer a principled framework for estimating a customer's future purchasing behaviour using only historical transaction data. Among these, the **BG/NBD (Beta-Geometric / Negative Binomial Distribution)** model and the **Gamma–Gamma** monetary model represent two standard components. The BG/NBD model estimates the expected number of future transactions, while the Gamma–Gamma model estimates the expected monetary value of those future transactions. Together they form a complete probabilistic Customer Lifetime Value (CLV) framework.

7.1 Theoretical Background

In non-contractual customer settings, firms do not observe the time at which a customer becomes inactive. Instead, inactivity must be inferred probabilistically from past transactional behavior. Probabilistic models such as the Beta-Geometric / Negative Binomial Distribution (BG/NBD) model and the Gamma–Gamma model provide mathematically rigorous frameworks for estimating the expected volume and value of future transactions. This section provides an extensive and detailed theoretical exposition of both models, including their assumptions, mathematical foundations, intuitive reasoning, likelihood functions, and implications.

7.1.1 The BG/NBD Model

The BG/NBD model [4] is a stochastic customer-base model designed to estimate the expected number of future repeat transactions for customers in non-contractual settings. It is built upon a combination of a Poisson transaction process and a geometric dropout process, with heterogeneity modeled via Gamma and Beta distributions respectively.

Model Assumptions The BG/NBD model rests on four key assumptions, each motivated by empirical regularities in consumer behavior:

1. **Conditional Transaction Process (Poisson):**
   Conditional on a latent transaction rate λ_i, customer i generates purchases as a Poisson process:

   $$X_i(t) \mid \lambda_i \sim \text{Poisson}(\lambda_i t).$$

   The Poisson process is the simplest point-process model satisfying independence of increments, stationarity, and no memory.

2. **Heterogeneity in Transaction Rates (Gamma):**
   Transaction rates vary across customers, with:

   $$\lambda_i \sim \text{Gamma}(r, \alpha)$$

   where r is the shape and α the rate parameter. The Gamma distribution is mathematically conjugate to the Poisson likelihood, enabling marginalization in closed form. This heterogeneity is crucial: without it, the model would unrealistically assume identical underlying behavior for all customers.

3. **Dropout Process (Geometric):**
   After each transaction, the customer may "drop out" with probability p_i:

   $$D_i \mid p_i \sim \text{Geometric}(p_i),$$

   where D_i counts the number of transactions before dropout.

   A geometric process implies a memoryless hazard of churn: the chance that a customer becomes inactive after the next purchase does not depend on past purchases.

4. **Heterogeneity in Dropout Propensity (Beta):**
   Dropout probabilities vary across customers:

   $$p_i \sim \text{Beta}(a, b).$$

   The Beta distribution is flexible and serves as a conjugate prior to the geometric likelihood, supporting closed-form marginalization.

Observed Variables For customer i observed in the interval $[0, T]$:

$$x_i = \text{number of repeat purchases,} \tag{10}$$

$$t_{x_i} = \text{time of the last purchase.} \tag{11}$$

Customers with identical (x_i, t_{x_i}) but different latent parameters behave differently, hence the need for probabilistic inference.

Likelihood Function Under the BG/NBD model, the likelihood of observing x_i transactions for customer i over the period $[0, T]$ is:

$$\mathcal{L}(x_i \mid r, \alpha, a, b) = \frac{\Gamma(r + x_i)}{\Gamma(r)} \frac{\alpha^r}{(\alpha + T)^{r + x_i}} \cdot \frac{a}{a + b + x_i - 1} \cdot {}_2F_1(a + b, r + x_i; a + b + x_i; \frac{T}{\alpha + T}). \tag{12}$$

The likelihood combines two behavioral components. The first term corresponds to the probability of observing x_i purchases, capturing customer heterogeneity in purchase frequency through a Negative

Binomial model. The second term represents the probability that the customer remains active at time T, modeled using a Beta-Geometric dropout process. The hypergeometric term adjusts this probability based on the timing of the last observed transaction. The expected number of future purchases over a horizon τ is:

$$E[X_{\text{future}}(\tau)] = \frac{a+b+x_i-1}{a-1}[1-(\frac{\alpha+T}{\alpha+T+\tau})^{r+x_i}], \quad a > 1. \tag{13}$$

This expectation decomposes into the probability that the customer remains active and the expected purchase intensity over the future period. Thus, the BG/NBD model provides an interpretable framework that jointly captures customer survival and purchasing behavior.

7.1.2 Gamma–Gamma Monetary Value Model

While the BG/NBD model predicts purchase frequency, firms must also estimate the monetary value of future transactions. The Gamma–Gamma model [3] provides a parametric framework for modeling customer-level average transaction value.

Model Assumptions The Gamma–Gamma model is based on three assumptions:

1. **Conditional Monetary Distribution:**
   Let v_{ij} be the transaction value of the j-th purchase by customer i. Conditional on a latent scale parameter μ_i:
   $$v_{ij} \mid \mu_i \sim \text{Gamma}(p, \mu_i),$$
   where $p > 0$ is shape and μ_i is scale. Transaction values tend to be right-skewed with positive support. The Gamma distribution is flexible and analytically convenient.

2. **Heterogeneity of Monetary Scale (Gamma):** Customers differ in their monetary tendencies:
   $$\mu_i \sim \text{Gamma}(q, \gamma).$$
   Some customers consistently spend more due to higher income, preferences, or purchasing goals. Modeling heterogeneity allows the distributional mixture to capture this diversity.

3. **Independence of Monetary Value and Purchase Frequency:**
   The monetary process is assumed independent of the purchase process.

   Empirically, frequency and average monetary value often exhibit little correlation in many retail environments. This assumption enables separability between BG/NBD and Gamma–Gamma models.

Likelihood Function Let n_i denote the number of transactions for customer i and $\bar{v}_i$ the observed average transaction value. Under the Gamma-Gamma model, the likelihood is given by:

$$\mathcal{L}(\bar{v}_i, n_i \mid p, q, \gamma) = \frac{\Gamma(pn_i + q)}{\Gamma(q)} \frac{\gamma^q}{(\gamma + n_i\bar{v}_i)^{pn_i+q}} (n_i\bar{v}_i)^{pn_i}. \tag{14}$$

The term $pn_i + q$ combines prior information with observed data, acting as an effective shape parameter. The term $\gamma + n_i\bar{v}_i$ reflects both the prior scale and the total observed spending. By sharing information across customers, the model produces stable estimates even when n_i is small.

Expected Monetary Value The expected average transaction value for customer i is:

$$E[v_i \mid \bar{v}_i, n_i] = \frac{pn_i + q}{n_i + \gamma - 1}\,\bar{v}_i. \tag{15}$$

This expectation is a shrinkage estimate. For customers with many transactions, the estimate is close to the observed average $\bar{v}_i$. For customers with few transactions, the estimate shrinks toward the population-level mean $\frac{q}{\gamma-1}$.

7.1.3 Combining BG/NBD and Gamma–Gamma for CLV

Customer lifetime value (CLV) over a horizon τ with discount rate δ is computed as the discounted expected future revenue:

$$\text{CLV}_i(\tau) = \sum_{t=1}^{\tau} \frac{E[X_{\text{future}}(t)] \cdot E[v_i]}{(1+\delta)^t}. \tag{16}$$

This formulation separates customer value into two interpretable components. The BG/NBD model estimates the expected number of future transactions, capturing customer activity over time. The Gamma–Gamma model estimates the expected monetary value per transaction, capturing customer profitability.By combining these components, CLV is obtained as a probabilistic and forward-looking measure of expected customer revenue.

7.2 Implementation Overview

First, transaction-level data are transformed into Recency–Frequency–Monetary (RFM) features again. Next, the BG/NBD model is fitted using the frequency, recency, and customer age T features. This model estimates the expected number of future transactions by jointly modelling purchasing intensity and customer dropout behavior. In parallel, the Gamma–Gamma model is fitted using the monetary component of the RFM representation. This model estimates the expected average transaction value for each customer, conditional on observed spending behavior.

Customer lifetime value is then computed by multiplying the predicted number of future transactions from the BG/NBD model with the expected monetary value from the Gamma–Gamma model. To evaluate predictive performance, the estimated CLV is compared against realised revenue observed after the end of the observation window. In this study, the observation period is defined to end 15 days before the maximum timestamp in the dataset. This design ensures that future transactions are available for out-of-sample validation while avoiding information leakage.

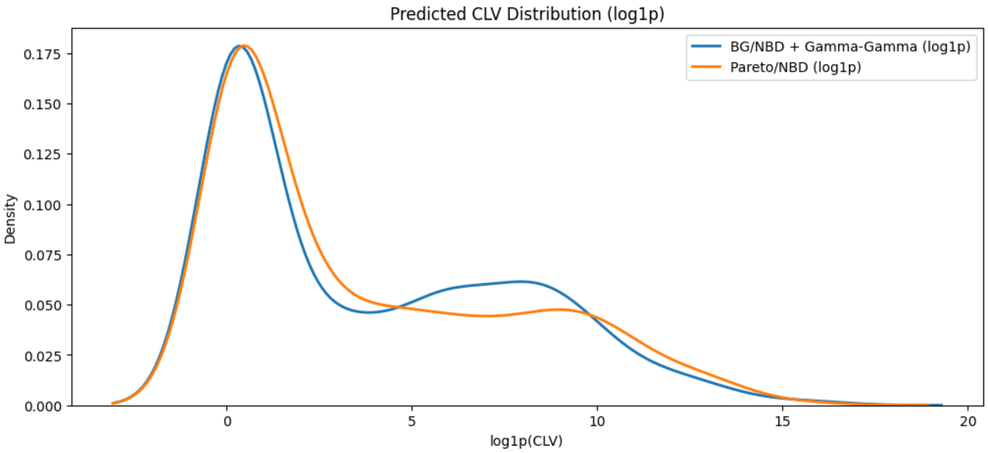


Figure 14 . BG/NBD and Pareto/NBD predicted distribution(log1p)

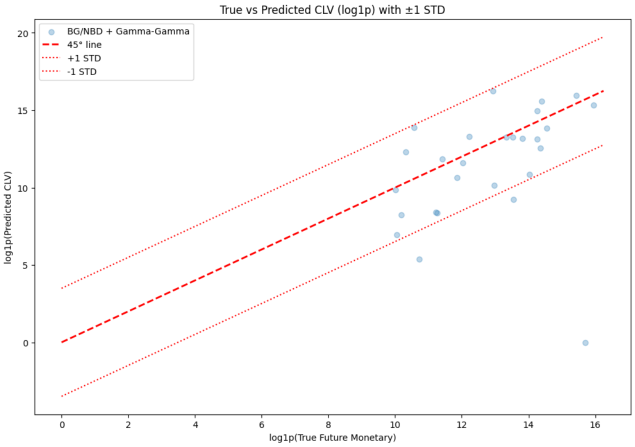


Figure 15 . BG/NBD with Gamma–Gamma Model

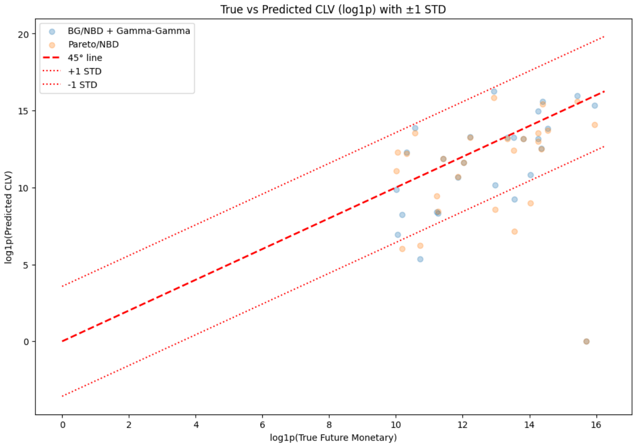


Figure 16 . True vs. Predicted CLV under BG/NBD and Pareto/NBD

The observation window ending 15 days before the maximum timestamp in the dataset is used for model evaluation, enabling comparison between predicted and real customer value on a held-out future period.

In contrast, for the future pipeline and downstream decision-making tasks, the CLV models are trained on the entire available dataset. In this setting, the observation period extends to the maximum timestamp observed in the data, thereby exploit all historical customer information.

8 Churn prediction and Survival Analysis

8.1 Theoretical Background

Customer churn prediction and survival analysis is a fundamental task in customer management, aiming to identify customers who are likely to discontinue their usage of a product or service. Accurate churn prediction enables organisations to design targeted retention strategies and allocate resources more efficiently. In the academic literature, churn modelling is commonly approached using two complementary methodological paradigms: classification-based modelling and survival analysis.

8.1.1 Churn Prediction

Classification-based approaches formulate churn prediction as a supervised binary classification problem. In this setting, the objective is to predict whether a customer will churn within a predefined future time window. Let $X \in R^d$ denote a vector of customer-level features, such as demographic attributes, behavioural summaries, and financial indicators. Let $Y \in \{0, 1\}$ represent the churn outcome, where $Y = 1$ indicates that churn occurs and $Y = 0$ otherwise. The goal is to estimate the conditional probability

$$P(Y = 1 \mid X),$$

which quantifies the likelihood of churn given observed customer characteristics.

Among classification methods, Gradient Boosting Decision Trees (GBDT) have become particularly popular due to their strong predictive performance and flexibility. GBDT models construct an additive predictor by sequentially combining weak learners, typically shallow decision trees. At iteration m, the model is updated according to

$$F_m(x) = F_{m-1}(x) + \nu h_m(x),$$

where $h_m(x)$ is a new base learner fitted to approximate the negative gradient of a specified loss function, and ν is a learning-rate parameter controlling the contribution of each learner. When logistic loss is used, the resulting model produces probabilistic outputs that can be directly interpreted as churn probabilities.

LightGBM is a highly efficient and scalable implementation of GBDT designed for large-scale and high-dimensional data. It introduces several algorithmic optimisations that are particularly relevant in churn prediction tasks. Gradient-based One-Side Sampling prioritises observations with large gradients, which contribute most to model learning, while reducing the number of less informative samples. Exclusive Feature Bundling reduces dimensionality by grouping mutually exclusive sparse features, improving computational efficiency. In addition, LightGBM adopts a leaf-wise tree growth strategy, which expands the leaf that yields the largest reduction in loss, allowing the model to capture complex patterns with fewer trees.

While classification models such as LightGBM are effective for predicting churn within a fixed horizon, they do not explicitly account for the timing of churn events or the presence of right-censored observations. Survival analysis addresses these limitations by modelling churn as a time-to-event process, enabling the estimation of both churn risk and expected customer lifetime.

8.1.2 Survival Analysis

Survival analysis is a class of statistical methods designed to model the time until a specific event occurs. A key feature of survival analysis is its ability to handle *right-censored data*, where the event of interest has not yet been observed for some individuals. In customer analytics, the event typically corresponds to churn, while the survival time represents the duration for which a customer remains active.

Kaplan–Meier Estimator(KM) [7]is a non-parametric method for estimating the survival function without making assumptions about the underlying distribution of survival times. Let $t_1 < t_2 < \cdots < t_m$ denote the ordered times at which churn events occur. The KM estimator is given by:

$$\hat{S}(t) = \prod_{t_j \le t} (1 - \frac{d_j}{n_j}), \tag{17}$$

where d_j is the number of churn events at time t_j, and n_j is the number of customers still at risk immediately before t_j.

Intuitively, the estimator updates the survival probability at each observed churn time by multiplying the conditional probability of surviving past that time. The Kaplan–Meier estimator provides an empirical baseline of customer retention and is commonly used to visualise survival curves and compare retention patterns across different customer segments.

Cox Proportional Hazards Model While the Kaplan–Meier estimator does not incorporate customer characteristics, the Cox Proportional Hazards (Cox PH) model [2] extends survival analysis by relating covariates to the risk of churn. For customer i with covariate vector X_i, the hazard function is defined as:

$$h(t \mid X_i) = h_0(t) \exp(X_i^\top \beta), \tag{18}$$

where $h_0(t)$ is the baseline hazard function and β is a vector of regression coefficients.

The key assumption of the Cox model is the *proportional hazards assumption*, which states that covariates have a multiplicative and time-invariant effect on the hazard. This means that covariates shift the risk of churn up or down by a constant factor, while the baseline hazard captures how churn risk evolves over time.

Estimation of β is performed using the partial likelihood:

$$L(\beta) = \prod_{j=1}^{m} \frac{\exp(X_{(j)}^\top \beta)}{\sum_{i \in R(t_{(j)})} \exp(X_i^\top \beta)}, \tag{19}$$

where $R(t_{(j)})$ denotes the set of customers still at risk at time $t_{(j)}$. The resulting coefficients $\hat{\beta}$ quantify the relative effect of covariates on churn risk and enable the estimation of individual hazard rates and survival curves.

Random Survival Forests [5] provides a flexible, non-parametric alternative to the Cox model. RSF extends random forest methodology to censored survival data by growing an ensemble of survival trees on bootstrap samples. At each split, candidate variables are selected to maximise differences in survival between child nodes, commonly using the log-rank test statistic.

Each tree produces an estimate of the cumulative hazard function, and the ensemble prediction is obtained by averaging across trees. Unlike the Cox model, RSF does not rely on the proportional hazards assumption and can naturally capture nonlinear effects and complex interactions among covariates. As a result, RSF is particularly well suited for high-dimensional customer data and heterogeneous churn behavior.

8.2 Implementation Overview

8.2.1 Churn prediction

For the churn prediction task, churn is defined using an inactivity-based rule derived only from transaction data. Specifically, a customer is labelled as churned if no transaction is observed within a fixed churn window of 50 days. Formally, let t_i^{last} denote the time of the most recent transaction for customer i, and let t_{max} be the maximum timestamp in the dataset. The churn indicator Y_i is defined as

$$Y_i = I(t_{\text{max}} - t_i^{\text{last}} > 50),$$

where $I(\cdot)$ is the indicator function. Only transaction-level data are used to define churn in order to capture confirmed disengagement reflected by purchasing behavior.

Customer-level features constructed during the data preprocessing stage are then used as inputs to the churn prediction model. These features include aggregated behavioural and transactional statistics that summarise historical customer activity. The objective is to learn a classifier that predicts whether a customer will churn ($Y_i = 1$) or remain active ($Y_i = 0$).

To evaluate predictive performance, the dataset is split into training and testing sets using an 80-20 partition. Due to class imbalance between churned and active customers, stratified sampling is employed to preserve the proportion of churn labels across both sets. This ensures that the classifier is trained and evaluated on data with consistent class distributions, reducing bias in performance estimation. The final performance of model in test set is presented in table 5

Table 5 . Test-Set Performance Metrics for the LightGBM Model in Churn Prediction

Metric	**Value**
ROC–AUC	0.923
Mean Squared Error (MSE)	0.083
Root Mean Squared Error (RMSE)	0.288
Mean Absolute Error (MAE)	0.099

8.2.2 Survival Analysis

Survival Analysis Training and Evaluation. For the survival analysis task, the dataset is partitioned into training and testing subsets using the same 80/20 split strategy as in the churn classification experiment. This consistent data-splitting protocol ensures comparability between modelling approaches. Right-censored observations are retained in both subsets to preserve the temporal structure of the data.

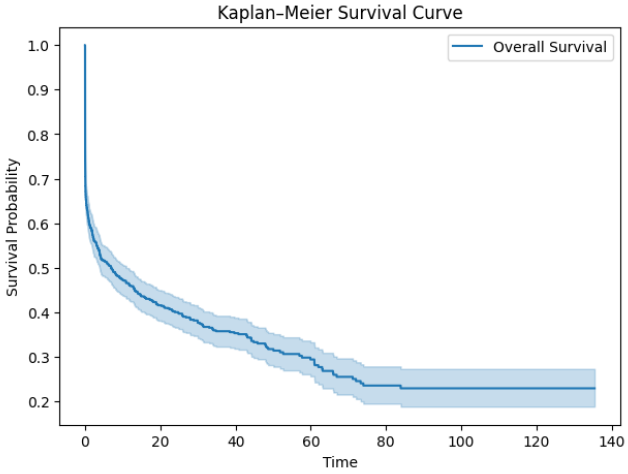


Figure 17 . Overall Kaplan–Meier Survival Curve

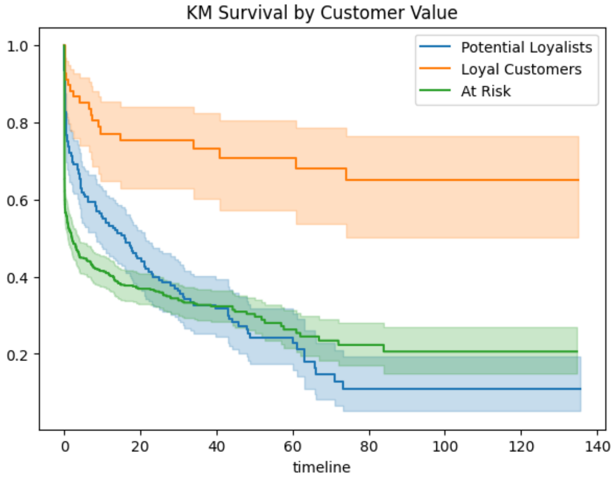


Figure 18 . Kaplan–Meier Survival Curves by Customer Segment

Three survival models are evaluated: the Kaplan–Meier estimator, the Cox Proportional Hazards (Cox PH) model, and the Random Survival Forest (RSF). The Kaplan–Meier estimator serves as a non-parametric baseline, capturing overall survival pattern(Figure 17,and segment patterns(Figure **??**). The

Cox PH model incorporates customer-level features to model the effect of covariates on churn risk under the proportional hazards assumption(results in Figure 19,20. The RSF model provides a flexible, non-parametric alternative capable of capturing nonlinear relationships and complex interactions(Figure21.

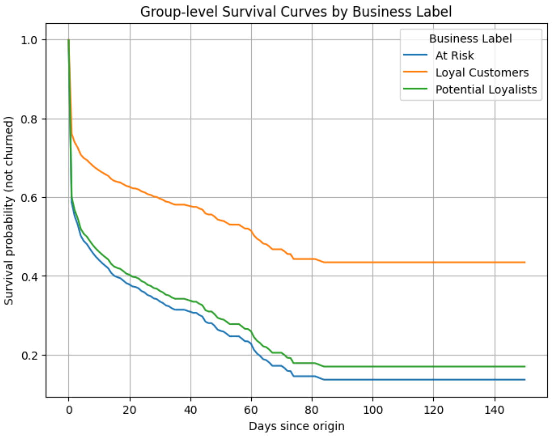


Figure 19 . Group-Level Survival Curves Estimated by Cox Proportional Hazards Model

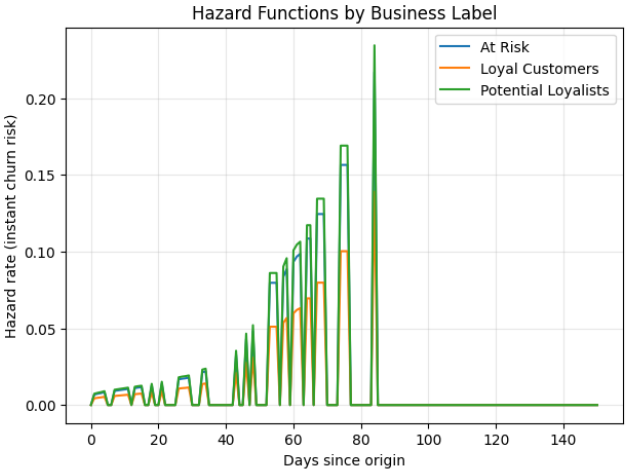


Figure 20 . Group-Level Estimated Hazard Functions

Model performance is evaluated on the test set using the concordance index (C-index), which measures the agreement between predicted risk scores and observed survival times. The C-index is defined as the proportion of all comparable customer pairs for which the model correctly orders survival times. In table 6, RSF is practically performed better than CoxPH.

Table 6 . Survival Model Performance Comparison on the Test Set

Model	**C-index**
Cox Proportional Hazards	0.696
Random Survival Forest	0.895

Therefore, given its superior predictive performance, RSF model is selected for downstream use in the final pipeline.

Part IV
Experimentation Framework

9 Theoretical Background

In practical decision-making systems, individual-level outcomes following an intervention are inherently stochastic. Even for customers with similar observed characteristics, the response to an action such as a promotion or reminder cannot be deterministically predicted. This uncertainty motivates the use of probabilistic outcome models rather than deterministic decision rules.

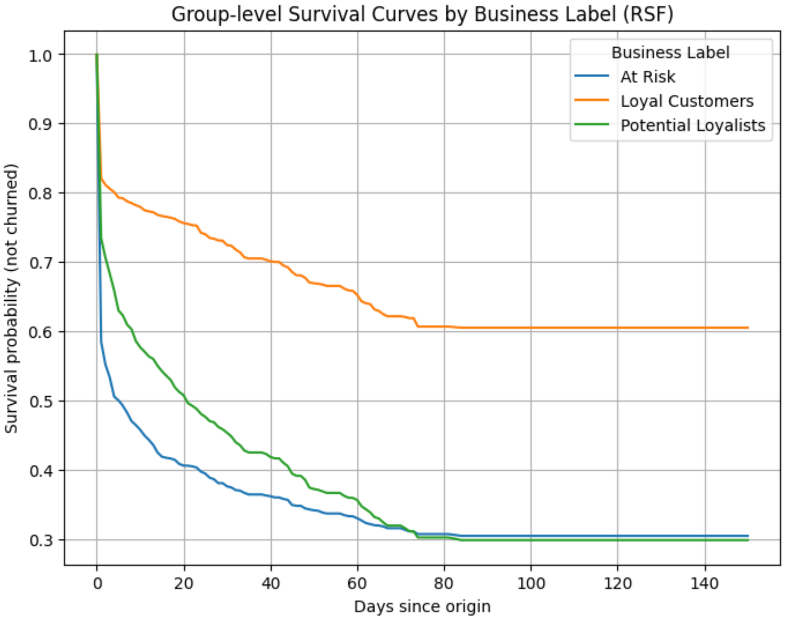


Figure 21 . Group-level Survival Curve by RSF

Let $Y_i \in \{0, 1\}$ denote the post-intervention outcome for customer i, where $Y_i = 1$ indicates a successful response. The outcome is modeled as a Bernoulli random variable with success probability p_i, reflecting the latent propensity of the customer to respond. Formally,

$$Y_i \sim \text{Bernoulli}(p_i).$$

In observational settings, p_i is not directly observable and must be approximated using predictive models or domain-informed assumptions.

Monte Carlo simulation provides a principled framework for approximating expected outcomes in stochastic systems when closed-form evaluation is infeasible. The core idea is to approximate expectations by repeatedly sampling from the underlying probability distributions and aggregating the resulting realizations.

In the context of customer targeting, Monte Carlo simulation enables the estimation of expected economic outcomes under a given policy by repeatedly simulating customer responses according to their modeled response probabilities. For a fixed policy π, the expected utility can be approximated as

$$E[U(\pi)] \approx \frac{1}{M} \sum_{m=1}^{M} U^{(m)}(\pi),$$

where $U^{(m)}(\pi)$ denotes the realized utility in the m-th simulation run, and M is the number of Monte Carlo iterations. By the Law of Large Numbers, this estimator converges to the true expected utility as $M \to \infty$.

Monte Carlo simulation is particularly well-suited to settings where outcomes are binary, costs are fixed, and decision rules depend on rankings or thresholds, as is the case in customer intervention policies.

10 Implementation Details

10.1 Segment-based policy assumptions

General Policy Design under Data Constraints. Due to data limitations, the available dataset contains only transactional information and lacks richer behavioural or contextual features such as customer sentiment, campaign history, or external trends. Consequently, complex personalised intervention strategies cannot be reliably constructed.

Instead, a general discount-based policy is created. Interventions are parameterised by a discount rate $X\%$, which is assumed to affect customer behaviour by increasing purchase lifetime and reducing churn probability. The impact of X is estimated empirically from historical data and is allowed to vary across customer segments. For each segment, the optimal discount level X^* is selected by directly evaluating segment-level outcomes under different discount values and choosing the value that maximises expected customer value.

To simulate heterogeneous customer responses to discounts, segment-specific behavioral parameters are tuned directly from observed data. The tunning relies on that customer segments with higher churn risk and lower engagement exhibit greater potential responsiveness to incentives.For each segment s, an average churn risk is computed as

$$\text{ChurnRisk}_s = 1 - E[\widehat{P}^{\text{survival}} \mid s],$$

which summarizes the propensity of customers in the segment to disengage within the evaluation horizon. This quantity serves as a primary driver of incentive sensitivity. A purchase responsiveness coefficient is then defined as

$$k_s = 1.3 \cdot \frac{\text{ChurnRisk}_s}{\max_{s'} \text{ChurnRisk}_{s'}},$$

assigning larger expected purchase lift to segments with higher relative churn risk, where incentives are more likely to influence behavior. Churn sensitivity is modeled as an inverse function of purchase activity:

$$c_s = 0.6 \cdot (1 - \frac{E[\widehat{P}^{\text{no-purchase}} \mid s]}{\max_{s'} E[\widehat{P}^{\text{no-purchase}} \mid s']}),$$

reflecting the intuition that less frequently purchasing segments exhibit stronger marginal response to discounts. The churn inflection parameter

$$m_s = 0.12 \cdot \text{ChurnRisk}_s$$

anchors the location of the behavioral transition to the empirically observed risk level of the segment, ensuring that churn reduction effects occur at realistic incentive magnitudes. Using these calibrated parameters, discount-dependent lift ranges are constructed. Purchase lift follows a diminishing-returns power-law relationship,

$$\text{Lift}_s^{\text{purchase}}(X) \in [k_s X^{0.85},\ k_s X^{1.05}],$$

while churn reduction is modeled through a sigmoid response,

$$r_s(X) = \frac{1}{1 + \exp\left(-c_s(X - m_s)\right)},$$

with uncertainty represented by a multiplicative band around the central response. This approach enables robust offline policy evaluation by grounding incentive response assumptions in segment-level empirical behavior while explicitly accounting for uncertainty and heterogeneity.

10.2 Monte Carlo simulation experiment

Baseline CLV Construction To evaluate the impact of incentive policies, a baseline reference value is first defined for each customer. Let $\widehat{P}_i^{\text{no-purchase}}$ denote the predicted probability of a purchase event, $\widehat{V}_i$ the predicted monetary value per purchase, and $\widehat{P}_i^{\text{survival}}$ the predicted survival probability. The baseline customer lifetime value (CLV) is defined as

$$\widehat{\text{CLV}}_i^{(0)} = \widehat{P}_i^{\text{no-purchase}} \cdot \widehat{V}_i \cdot \widehat{P}_i^{\text{survival}}.$$

This quantity represents the expected future value of a customer in the absence of any promotional intervention and serves as the counterfactual benchmark for policy evaluation.

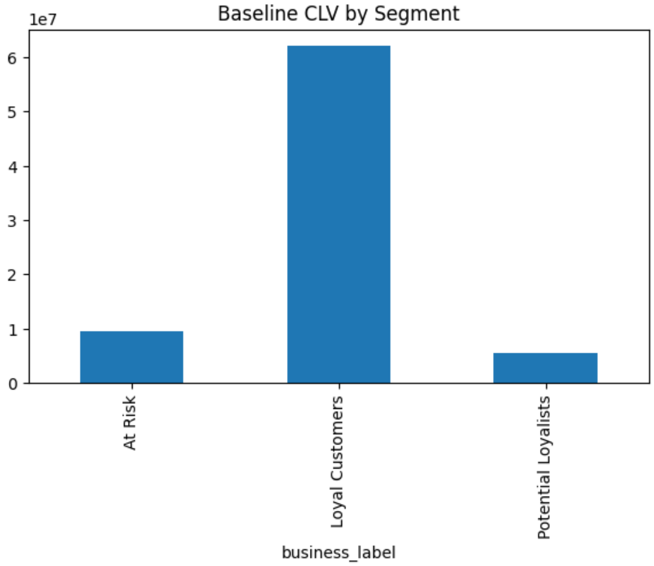


Figure 22 . Baseline CLV for 3 customer segments

Monte Carlo Simulation of Discount Policies To assess the effect of a discount policy, we simulate its impact under uncertainty using a Monte Carlo framework. For a given customer segment s and discount level X, segment-specific purchase and churn lift ranges are obtained from the calibrated response functions. Within each simulation run, a purchase lift and a churn reduction factor are independently sampled from their respective ranges, reflecting uncertainty in behavioral response.

For customers belonging to segment s, the purchase probability is adjusted as

$$\widehat{P}_{i,\text{new}}^{\text{no-purchase}} = \widehat{P}_i^{\text{no-purchase}} \cdot (1 + \delta^{\text{purchase}}),$$

while the churn probability is reduced according to

$$\widehat{P}_{i,\text{new}}^{\text{churn}} = (1 - \widehat{P}_i^{\text{survival}}) \cdot (1 - \delta^{\text{churn}}),$$

yielding an updated survival probability

$$\widehat{P}_{i,\text{new}}^{\text{survival}} = 1 - \widehat{P}_{i,\text{new}}^{\text{churn}}.$$

Customers outside the targeted segment retain their baseline probabilities.

The post-intervention CLV is then computed as

$$\widehat{\text{CLV}}_i^{(X)} = \widehat{P}_{i,\text{new}}^{\text{no-purchase}} \cdot \widehat{V}_i \cdot \widehat{P}_{i,\text{new}}^{\text{survival}}.$$

Accounting for Discount Cost The cost of offering a discount is explicitly incorporated into the evaluation. For customers in the targeted segment, the expected discount cost is defined as

$$\text{Cost}_i(X) = X \cdot \widehat{P}_{i,\text{new}}^{\text{no-purchase}} \cdot \widehat{V}_i,$$

while no cost is incurred for non-targeted customers.

The net gain of a discount policy is therefore given by

$$\text{Gain}(X) = \sum_i (\widehat{\text{CLV}}_i^{(X)} - \widehat{\text{CLV}}_i^{(0)} - \text{Cost}_i(X)).$$

The simulation procedure is repeated multiple times to obtain an empirical distribution of net gains for each discount level. From this distribution, summary statistics are computed, including the mean gain, median gain, probability of positive gain, and lower and upper tail quantiles. This provides a risk-aware assessment of each policy rather than relying on point estimates alone.

Discount optimization is performed by evaluating a discrete set of candidate discount levels $X \in \mathcal{D}$. For each segment, the optimal discount is selected as

$$X_s^* = \arg\max_{X \in \mathcal{D}} E[\text{Gain}(X)],$$

where the expectation is approximated by Monte Carlo simulation. This segment-wise optimization reflects heterogeneity in customer behavior and enables targeted incentive strategies that balance expected return against downside risk.

Table 7 . Optimal Discount Policy Outcomes by Customer Segment

Segment	**Optimal Discount**	**Mean Gain**	**Median Gain**	**Worst 5% Gain**	**Best 95% Gain**
At Risk	0.12	579,85	581,74	281,68	872,26
Potential Loyalists	0.08	303,06	302,34	169,03	436,89
Loyal Customers	0.01	589,95	589,41	336,71	847,90

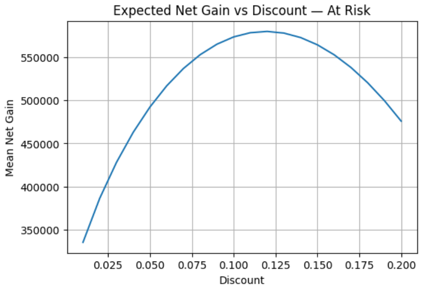


Figure 23 . At-Risk Segment Discount Simulation

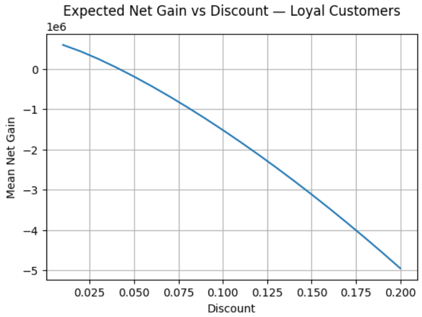


Figure 24 . Loyal Customers Discount Simulation

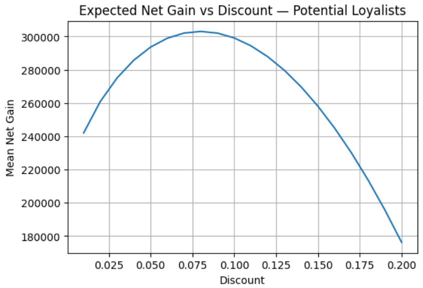


Figure 25 . Potential Loyalists Discount Simulation

After experimentation, the optimal discount levels obtained for the At-Risk, Potential Loyalists, and Loyal Customers segments are 12%, 8%, and 1%, respectively(Table7 and Figures 23,24,25. These results reveal an interesting behavioral pattern. At-risk customers tend to be highly price-sensitive and exhibit low brand loyalty; they are more likely to purchase from whichever option offers the lowest price, and increasing discounts effectively stimulates their purchasing behavior.

As customer loyalty increases, the effectiveness of monetary discounts diminishes. Loyal customers typically continue purchasing from familiar vendors regardless of discounts; in this case, discounts serve only as a mild incentive rather than a strong driver of behavior. For this segment, non-monetary, psychologically driven strategies are likely to be more effective, such as VIP recognition, personalized birthday messages, or premium customer service experiences, rather than purely financial incentives.

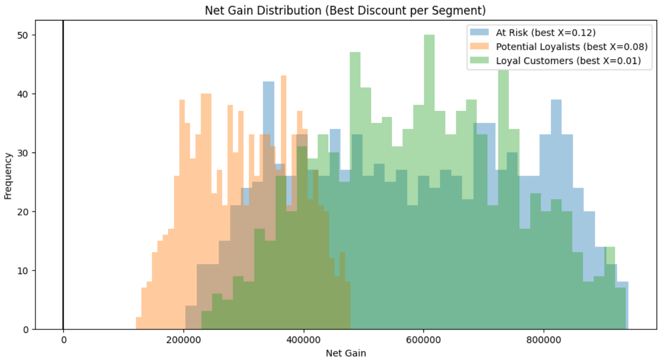


Figure 26 . Net Gain distribution

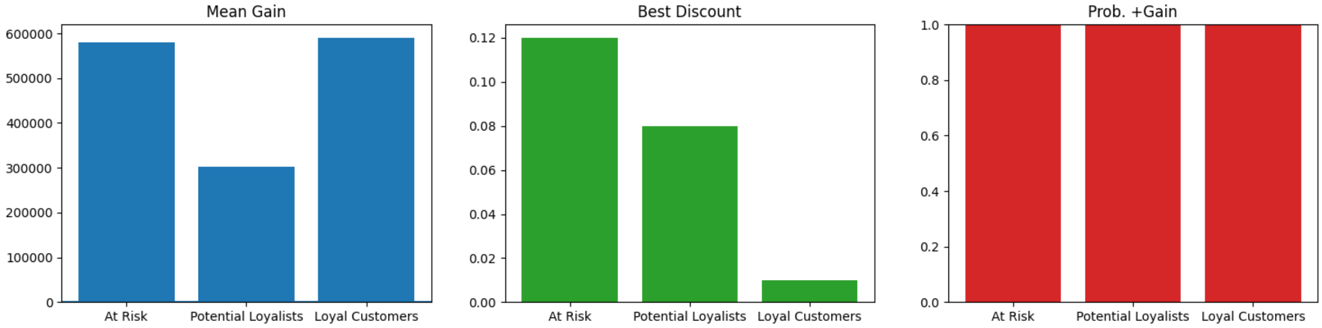
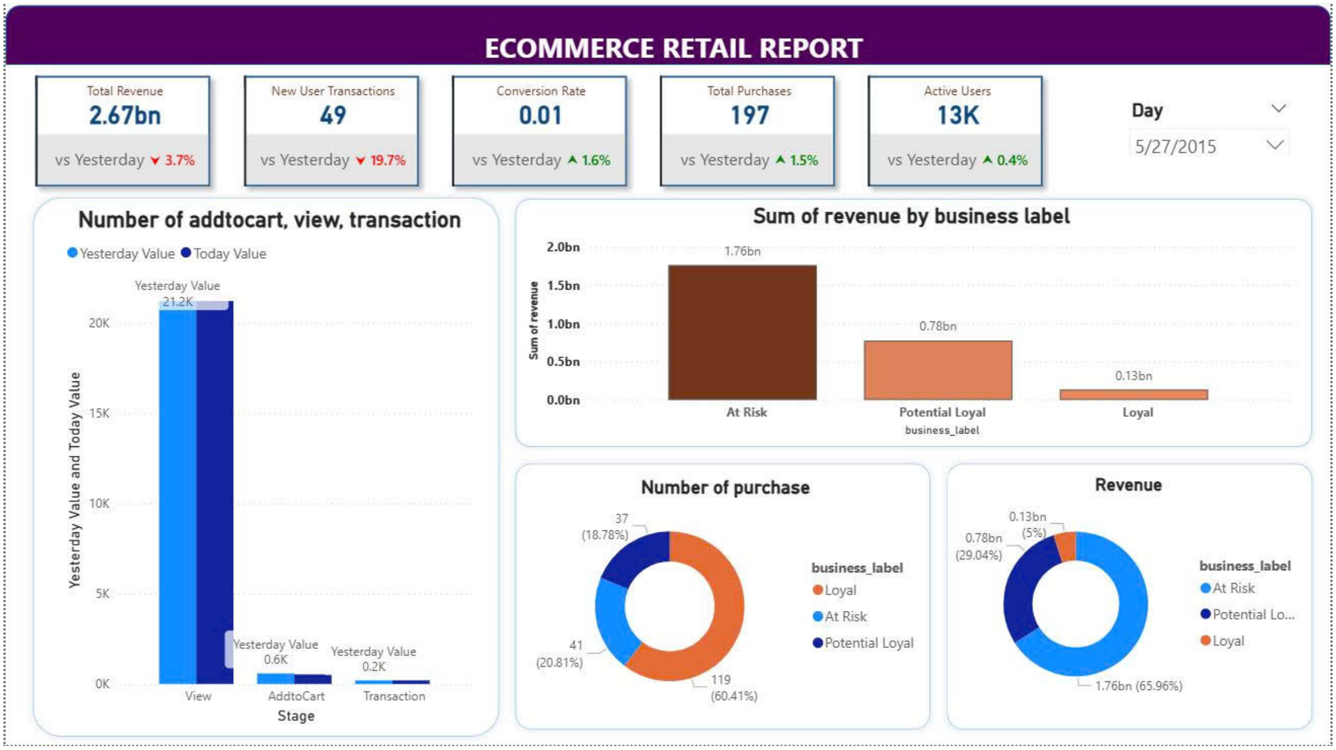


Figure 27 . Group-level policy effects

Figures 26 and 27 show that the optimal discount policies yield strictly positive net gains across all customer segments, indicating robust profitability under uncertainty. At-Risk customers generate high gains due to strong responsiveness to incentives, whereas Potential Loyalists exhibit smaller and more concentrated gains, reflecting limited marginal returns. Loyal Customers achieve the largest gains with minimal discounts, confirming that modest incentives suffice for high-value customers and underscoring the importance of segment-specific discount strategies.

11 Dashboard



ECOMMERCE RETAIL REPORT

Total Revenue 2.67bn vs Yesterday ▼ 3.7%

New User Transactions 49 vs Yesterday ▼ 19.7%

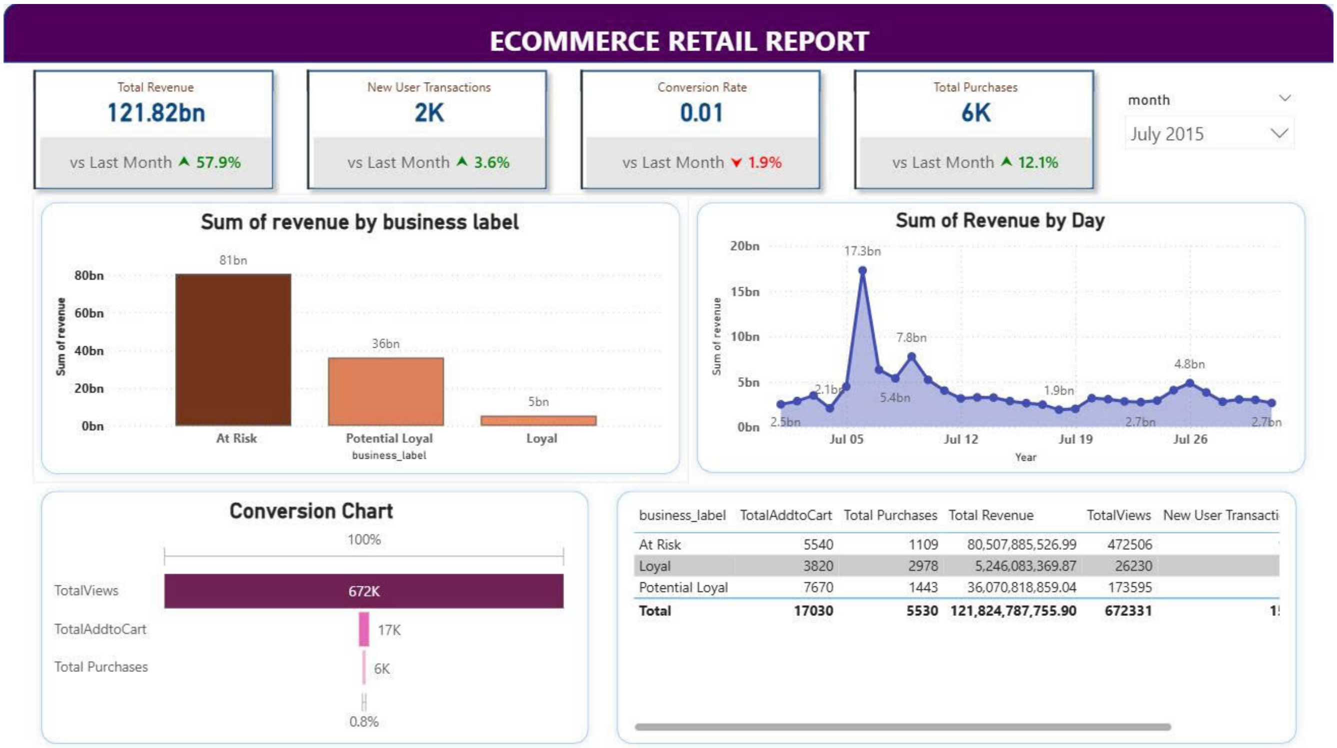
Conversion Rate 0.01 vs Yesterday ▲ 1.6%

Total Purchases 197 vs Yesterday ▲ 1.5%

Active Users 13K vs Yesterday ▲ 0.4%

Day: 5/27/2015

Figure 28 . Strategic Dashboard



ECOMMERCE RETAIL REPORT

Total Revenue	New User Transactions	Conversion Rate	Total Purchases
121.82bn	2K	0.01	6K
vs Last Month ▲ 57.9%	vs Last Month ▲ 3.6%	vs Last Month ▼ 1.9%	vs Last Month ▲ 12.1%

month: July 2015

business_label	TotalAddtoCart	Total Purchases	Total Revenue	TotalViews	New User Transacti
At Risk	5540	1109	80,507,885,526.99	472506	
Loyal	3820	2978	5,246,083,369.87	26230	
Potential Loyal	7670	1443	36,070,818,859.04	173595	
Total	**17030**	**5530**	**121,824,787,755.90**	**672331**	**1!**

Figure 29 . Operational Dashboard

Part V

Conclusion

In short, we constructed an end-to-end pipeline that spans from raw data ingestion to the decision layer. At each layer, the data is split appropriately to evaluate model performance, after which the best-performing model is selected and retrained on the full dataset before being passed to the subsequent layer in the pipeline.

Weakness & Further Work

Although our pipeline is relatively extensive and extract multiple insights as well as the construction of a decision layer, we were not able to fully exploit information in our dataset related to product availability, categoryid, or non-transactional events such as add-to-cart and view actions,which significantly reduces the overall analytical depth. Since our dataset is primarily centered around Customer Lifetime Value (CLV) modeling, the analysis focuses mainly on transactional data, which represents a notable weakness of the project.

Besides, the dataset is relatively sparse and lacks critical information necessary for more advanced analysis. Key variables such as product prices had to be assumed and estimated, and there is no historical user behavior data, campaign history, or temporal trends available. As a result, it is challenging to design discount or intervention policies that are both robust and strategically meaningful.

Furthermore, the dataset is relatively small and lacks sufficient diversity to support the construction of all three standard types of dashboards. Consequently, in this project, we focus on developing only two dashboards—namely, the strategic dashboard and the operational dashboard.

In future work, if given the opportunity to revisit a similar project, we would incorporate multiple data sources to develop a more professional, robust, and customer-centric system.

Nevertheless, we are satisfied with the insights obtained and the outcomes achieved in this project.

Acknowledgment

We would like to express our sincere gratitude to Professor Nguyen Binh Minh, our instructor for this final semester project, for giving us invaluable knowledge through Business Intelligence course. Besides, giving respect to community industry and engineering in field of data science, machine learning and business intelligence, whose public material and tutorial about these fields. Finally, this report is heard by all members of our team.

References

[1] Simon Cooke. Database marketing: strategy or tactical tool? *Marketing Intelligence & Planning*, 12(6):4–7, 1994.

[2] David R Cox. Regression models and life-tables. *Journal of the Royal Statistical Society: Series B (Methodological)*, 34(2):187–202, 1972.

[3] Peter S Fader, Bruce GS Hardie, and Ka Lok Lee. Rfm and clv: Using iso-value curves for customer base analysis. *Journal of marketing research*, 42(4):415–430, 2005.

[4] Peter S Fader, Bruce GS Hardie, and Ka Lok Lee. “counting your customers” the easy way: An alternative to the pareto/nbd model. *Marketing science*, 24(2):275–284, 2005.

[5] Hemant Ishwaran, Udaya B Kogalur, Eugene H Blackstone, and Michael S Lauer. Random survival forests. 2008.

[6] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. Isolation forest. In *2008 eighth ieee international conference on data mining*, pages 413–422. IEEE, 2008.

[7] KAPLAN SE. Non-parametric estimation from incomplete observation. *J Am Stat Assoc*, 53:457–481, 1958.