Aalto University
School of Science

Master’s Programme in Mathematics and Operations Research

Predicting purchase decisions of online store visitors

Tomas Toro

Master’s Thesis
2025

Aalto University
School of Science

Author Tomas Toro

Title Predicting purchase decisions of online store visitors

Degree programme Mathematics and Operations Research

Major Systems and Operations Research

Supervisor Doc. Tuomas Raivio

Advisor Doc. Tuomas Raivio

Date 28 July 2025 **Number of pages** 73 **Language** English

Abstract

Understanding the behavioral patterns that lead online store visitors to complete purchases is critical for improving the performance of e-commerce businesses. While much of the existing literature focuses on estimating purchase probabilities or optimizing the likelihood of purchase through interface changes and personalization, this thesis takes a modeling-oriented approach to classify whether a user session will result in a purchase. The objective is to build a model that predicts the binary purchase outcomes based on observed clickstream data, which consists of timestamped sequences of user interactions such as page views, clicks, and navigation paths on a website or digital platform.

To achieve this, the study applies a hidden semi-Markov model (HSMM), which captures both the sequence and duration of user behaviors during a session. The model is estimated on event-level data from a Finnish online clothing store, including detailed behavioral events such as product views, cart additions, and checkout actions. Through this temporal modeling framework, the analysis highlights the value of duration-aware modeling in understanding visitor behavior in an online store.

The results demonstrate that modeling session-level behavior as a sequence of hidden states with hidden semi-Markov models can provide accurate classification of purchase outcomes. In addition to the modeling work, this thesis presents a literature review that defines the purchase prediction task as a classification problem, examines factors that influence purchasing behavior, and reviews common modeling approaches used in online retail contexts. It also reflects on the challenges of preprocessing large-scale behavioral data and discusses the implications of model assumptions in the context of limited user tracking and session variability.

Keywords E-commerce, purchase prediction, hidden semi-Markov model, binary classification, conversion rate optimization

Aalto-yliopisto
Perustieteiden
korkeakoulu

Tekijä Tomas Toro

Työn nimi Verkkokaupan vierailijoiden ostopäätösten ennustaminen

Koulutusohjelma Mathematics and Operations Research

Pääaine Systems and Operations Research

Työn valvoja Dos. Tuomas Raivio

Työn ohjaaja Dos. Tuomas Raivio

Päivämäärä 28.7.2025 **Sivumäärä** 73 **Kieli** englanti

Tiivistelmä

Verkkokaupan toiminnan tehostaminen edellyttää ymmärrystä siitä, millaiset käyttäytymismallit johtavat asiakkaan ostopäätökseen. Aiempi tutkimus on keskittynyt pitkälti ostotodennäköisyyksien arviointiin tai oston todennäköisyyden parantamiseen esimerkiksi käyttöliittymäsuunnittelun tai personoinnin keinoin. Tässä opinnäytteessä lähestytään aihetta mallipohjaisesti, ja tavoitteena on luokitella käyttäjäistunnot ostoon johtaviin ja ei ostoon johtaviin niistä kerätyn datan perusteella. Tarkoituksena on rakentaa malli, joka ennustaa binääristä ostotapahtumaa havaitun käyttäytymisdatan perusteella. Data koostuu tapahtumaketjuista, jotka kuvaavat käyttäjän toimintaa verkkosivustolla tai digitaalisella alustalla, kuten sivujen katselut, klikit ja siirtymiset sivujen välillä.

Työssä hyödynnetään hidden semi-Markov mallia, joka ottaa huomioon sekä käyttäjätoiminnan järjestyksen että niiden keston istunnon aikana. Malli estimoidaan suomalaisen verkkovaatekaupan vierailijadatalla, joka sisältää yksityiskohtaisia käyttäytymistapahtumia, kuten tuotesivujen katseluita, ostoskoriin lisäämisiä ja kassalle siirtymisiä. Lähestymistapa korostaa ajallisen rakenteen ja tapahtumien keston huomioimista ostopolkujen ymmärtämisessä.

Tulokset havainnollistavat, että tässä aineistossa käyttäjäistuntojen mallintaminen piilotettujen tilojen sarjana hidden semi-Markov malleilla mahdollistaa ostotapahtumien luotettavan luokittelun. Varsinaisen mallinnustyön lisäksi opinnäyte sisältää kirjallisuuskatsauksen, jossa ostokäyttäytymisen ennustamista lähestytään luokitteluongelmana. Katsauksessa tarkastellaan ostokäyttäytymiseen vaikuttavia tekijöitä sekä yleisimpiä mallinnusmenetelmiä, joita on sovellettu verkkokauppaympäristössä. Työ käsittelee myös laajamittaisen käyttäytymisdatan esikäsittelyn haasteita ja mallin oletusten vaikutuksia tilanteissa, joissa käyttäjäseuranta on rajoitettua ja istunnoissa esiintyy suurta vaihtelua.

Avainsanat Verkkokauppa, oston ennustaminen, hidden semi-Markov malli,
konversio-optimointi, binääriluokittelu

Preface

I want to thank Docent Tuomas Raivio for his advice and patience during this thesis. I also want to thank my partner, family, and friends for their support throughout my studies. While challenging at times, working on this thesis was very interesting, and I am happy with how it turned out.

Espoo, 28 July 2025

Tomas Toro

Contents

Abbreviations

CR	conversion rate
CRO	conversion rate optimization
CDN	content delivery network
SEO	search engine optimization
SERP	search engine results page
UX	user experience
GDPR	general data protection regulation
HMM	hidden Markov model
HSMM	hidden semi-Markov model
EM	expectation-maximization
RNN	recurrent neural network
LSTM	long short-term memory
GTM	Google Tag Manager
GA4	Google Analytics 4
GCP	Google Cloud Platform
MCC	monotonicity, convexity, and concavity
ML	machine learning
DL	deep learning
SVM	support vector machine
MCMC	Markov chain Monte Carlo
CDF	cumulative distribution function
PPT	past purchasing tendency
DD-HMM	duration-dependent hidden Markov model
BPTT	backpropagation through time
Caser	convolutional sequence embedding recommendation model
MAP	maximum a posteriori
ROC AUC	receiver operating characteristic area under the curve

1 Introduction

E-commerce, or electronic commerce, has grown to become a dominant force in global retail, with its market share steadily increasing each year (Coppola [16]). Simultaneously, competition has intensified as technological advancements have made it easier for businesses to establish online stores with minimal effort (Dushnitsky and Stroube [21]). Unlike traditional brick-and-mortar shopping, where physical location and store ambiance play a role in purchasing decisions, online retail offers customers the ability to browse multiple stores effortlessly, compare prices, and switch between retailers within seconds. This heightened accessibility increases competition and reduces brand loyalty, making it more difficult for businesses to retain visitors and convert them into paying customers. As a result, understanding the factors that influence purchase decisions has become a critical success factor for e-commerce businesses (Gudigantala et al. [36]).

The process of increasing the proportion of visitors who convert is known as conversion rate optimization (CRO) (Saleh and Shukairy [48]). A conversion refers to an event where a visitor completes a desired action, such as signing up for a newsletter or making a purchase. The conversion rate is the percentage of visitors who complete such actions. In this thesis, conversion rate optimization refers to the practice of refining an online store's design, content, and user experience to increase the proportion of visitors who complete a purchase. Conversion rate optimization has been a topic of research since the early days of online shopping (Moe and Fader [62], Montgomery [64]).

While businesses employ various strategies to increase conversion rates, achieving meaningful improvement remains an ongoing challenge. Common CRO techniques include optimizing website design for usability, implementing A/B testing to compare different layouts and call-to-action buttons, using recommendation algorithms to personalize product offerings, and leveraging targeted discounts or remarketing campaigns to re-engage potential buyers (Saleh and Shukairy [48]). Despite these efforts, conversion rates in e-commerce have remained relatively low, often averaging between 1% and 4.9% (Zumstein and Kotowski [90]). This persistence suggests that traditional approaches may not be fully capitalizing on user behavior data to anticipate and respond to purchase intent in real time. Thus, predicting conversion decision based on behavioral signals presents an opportunity to enhance decision-making and optimize conversions more effectively.

The primary objective of this thesis is to estimate whether a visitor will make a purchase in an online store based on their on-site behavior, prior to action clearly indicating a purchase. Identifying this can help store owners make real-time decisions that guide visitors toward making a purchase. For instance, high-intent visitors could be shown personalized product recommendations, while hesitant users could receive time-sensitive discount offers to encourage immediate action (Moe and Fader [62]). Additionally, identifying behaviors associated with cart abandonment allows businesses to implement proactive interventions such as automated email reminders or chatbot assistance, minimizing lost sales opportunities. By leveraging purchase predictions, businesses can move beyond reactive strategies and proactively tailor the shopping experience to individual visitor needs, increasing revenue and improving customer satisfaction.

This thesis is organized as follows. Chapter 2 provides background on key aspects of e-commerce, including business characteristics, consumer behavior, web analytics, and conversion rate optimization. Chapter 3 reviews prior research on modeling purchase decisions in online stores. It introduces the classification framing used in this thesis, discusses features unique to purchase prediction, and describes prevailing modeling approaches. Chapter 4 presents the empirical analysis, beginning with a description of the data and preprocessing steps, followed by the implementation of a hidden semi-Markov model for purchase prediction. Chapter 5 presents the results, including outcomes related to data preparation as well as insights from the model's performance, behavioral patterns it captures, and its predictive capabilities. Chapter 6 discusses the findings in relation to the research objectives, and Chapter 7 concludes the thesis.

The empirical examples in this thesis are subject to limitations related to data consent, tracking scope, and the generalizability of results from a single online store. Despite the limitations, the hidden semi-Markov model seems to produce a reliable estimate of purchase intent in the dataset used in this study.

2 Foundations of user behavior in online retail

2.1 Key characteristics of e-commerce business

E-commerce businesses operate in a digital environment that fundamentally differs from traditional brick-and-mortar retail. While both models aim to facilitate transactions and serve customer needs, the online marketplace introduces distinct opportunities and challenges. One of the most significant differences is accessibility: a physical store can only serve customers within a certain geographic area, whereas an online store is typically accessible to anyone with an internet connection, regardless of location. Additionally, visiting a brick-and-mortar store requires time and effort, whereas an online store can be accessed within seconds from any internet-enabled device. This ease of access means that online stores generally receive significantly higher traffic than their physical counterparts (Moe and Fader [62]). This also contributes to a more competitive landscape, where businesses must continuously refine their strategies to attract, engage, and convert visitors into paying customers.

However, higher traffic does not necessarily translate to higher sales. Unlike physical stores, where customers are more likely to be intent on making a purchase, online visitors may browse casually without committing to a transaction (Bucklin et al. [12]). One reason for this difference in purchase intent may be explained by the sunk cost fallacy: a cognitive bias where individuals feel compelled to follow through with an action once they have invested time or effort into it (Arkes and Blumer [4]). In the context of brick-and-mortar retail, the effort required to physically visit a store can psychologically reinforce the decision to buy. In contrast, the low-effort nature of visiting an online store makes it easier for users to abandon the shopping process at any stage. Another key distinction is the limitation of space and personnel in traditional stores. Only a certain number of customers can be present at a time, and purchases are constrained by the availability of checkout staff. Online stores, on the other hand, do not face these restrictions; thousands of customers can browse and complete purchases simultaneously. Previously, server capacity could have presented challenges when handling a high number of visitors, potentially leading to slow load times or downtime during traffic spikes. However, with the increasing adoption of modern technologies such as content delivery networks (CDNs), this is no longer a problem in most cases. CDNs are distributed networks of servers that cache and deliver content from locations closer to users, reducing latency and distributing traffic loads (George and George [29]).

The low barriers to entry in e-commerce have contributed to a highly competitive market. Nowadays, almost anyone can set up an online store at little to no cost, even without programming knowledge or advanced technical skills (Dushnitsky and Stroube [21]). Additionally, the widespread use of popular e-commerce platforms, such as Shopify and WooCommerce, has led to a standardized look and feel among online stores, as these platforms offer pre-designed templates aimed at making it easier for visitors to navigate the store and make purchases.

A crucial challenge for online retailers is attracting visitors to their stores. All online stores exist in the same virtual space, where visibility depends entirely on factors such as direct traffic (customers entering the site's URL), search engine rankings, and online advertisements. Search engine optimization (SEO) is a key aspect in increasing store visibility, with retailers investing significant effort into researching relevant search terms and optimizing their websites accordingly to improve their ranking on search engine results pages (SERPs). Higher search rankings can have a substantial impact on sales (Baye et al. [6]).

In addition to SEO, many online retailers invest in paid advertising. Search engines sell ad placements at the top of search results, allowing businesses to bid for prominent visibility. Social media advertising is also widely used, as it enables seamless redirection to the store with a single click. Moreover, many advertising platforms, such as Meta and Google, allow businesses to showcase their products directly within advertisements, which serves as an external display window for the store.

2.2 Consumer behavior in e-commerce

Consumer behavior in e-commerce differs significantly from shopping in physical stores due to the nature of the digital environment. While online stores offer convenience and a wider selection of products, the absence of in-person interaction and tangible product experiences impacts how customers make purchasing decisions.

One characteristic of online consumer behavior is the ease of comparison shopping. Customers can easily compare offerings from different stores by visiting multiple websites or using price comparison tools. As a result, prices among competitors tend to be closely aligned, forcing online retailers to differentiate themselves in other ways (Lee et al. [53]). This increased competition makes it important to have a visually appealing and user-friendly website, along with high-quality product images and detailed descriptions. Providing comprehensive product information is especially critical, as customers often cannot physically inspect items before making a purchase.

Another major factor influencing online shopping behavior is trust. Unlike in physical stores, where customers can directly see and assess products before buying, online shoppers must rely on digital representations and product descriptions. Trust is further complicated by the requirement to enter sensitive information, such as payment details, before receiving the product. To mitigate this uncertainty, many online retailers leverage customer reviews and testimonials as social proof of their reliability. Additionally, the overall design and professionalism of an online store play a crucial role in establishing credibility. Customers may hesitate to enter payment information on a website that appears untrustworthy or poorly maintained (Seckler et al. [76]).

The user experience (UX) of an online store also significantly impacts consumer behavior. Research indicates that even minor inconveniences, such as slow loading times, can lead to site abandonment, making performance optimization essential (Pushkar et al. [72]). Additionally, minimizing distractions and streamlining navigation can help keep potential customers engaged (Seckler et al. [76]).

Replicating the personal touch of an in-store shopping experience is another challenge for online retailers. Physical stores offer direct customer service, allowing shoppers to ask questions and receive immediate assistance. In contrast, online shoppers interact with a digital interface, making it more difficult to establish trust and provide personalized service (Madhuri et al. [56]). To address this, many e-commerce businesses implement chatbots, personalized recommendations, and live customer support features to enhance the shopping experience.

2.3 Web analytics

Web analytics refers to the practice of collecting and analyzing web data to understand how visitors interact with a website. It is commonly used to evaluate how effectively a site fulfills its purpose, and many websites rely on analytics tools to monitor performance and user behavior. Common metrics include visits, page views, traffic sources, and user-specific data such as device type, browser, and geographical location. Many website hosting services provide basic analytics on site performance, while external measurement tools can be integrated for more detailed data collection and the ability to define custom metrics (Bekavac and Garbin Praničević [7]). In online retail, understanding visitor interactions is particularly crucial, as it directly affects sales performance and customer satisfaction (Gudigantala et al. [36]).

In Europe, regulations such as the General Data Protection Regulation require websites to obtain user consent before gathering any data that is not essential for the website's functionality (European Union [23]). This consent is typically requested through a cookie banner: a notice displayed when a user first visits a site. Cookies are small data files stored in the user's browser that enable websites to track and remember user activity across sessions. While essential cookies support basic site functionality, others collect analytics or marketing data. The cookie banner determines what types of cookies may be set, thereby affecting how much behavioral data can be collected for analysis.

Web analytics enables online retailers to analyze user behavior on their digital storefronts and make data-driven decisions to guide user actions toward more desirable outcomes (Bucklin et al. [12]). Various qualitative and quantitative methods are available for examining user behavior. An example of a qualitative method is session recording, where a script tracks a user's mouse movements, clicks, and scrolling behavior throughout their session. This data is then compiled into a video-like playback that overlays interactions onto the webpage. Website owners can analyze these recordings to identify usability issues and optimize the site's design and functionality (Filip and Čegan [26]). Another tool is heat maps, which visually represent areas of a webpage that receive the most user interaction, typically through clicks, scrolls, or mouse movements (Șoavă and Raduteanu [79]). To analyze user navigation patterns, clickstream data is often utilized. Clickstream data consists of a sequence of recorded clicks that map out a user's journey through a website, providing insights into browsing habits, drop-off points, and areas for improvement (Bucklin et al. [12]).

Web analytics has inherent characteristics and limitations that must be considered when interpreting data. One key limitation in Europe is that only data from visitors who have accepted cookies is collected, meaning that users who decline tracking remain unaccounted for (European Union [23]). Additionally, ad blockers and privacy extensions can prevent analytics tools from functioning by blocking all third-party scripts from loading, effectively making these users invisible in the dataset (Garimella et al. [28]). Most analytics tools track users based on their browser rather than a unique user identity. As a result, a single user visiting from multiple devices may be counted multiple times, while multiple individuals sharing a device may appear as a single user (Bekavac and Garbin Praničević [7]). Analytics data is also limited to on-site activity; it does not capture a user's prior browsing behavior, such as visits to competitor sites or exposure to advertisements, even though these factors may strongly influence purchasing decisions (Ferrini and Mohr [24]). Furthermore, website traffic may include non-target users, such as competitors analyzing the site or bots collecting data, leading to interactions that do not reflect genuine user behavior (Xu et al. [87]). Despite these limitations, web analytics remains an essential tool for understanding user behavior. While individual data points may be incomplete or skewed, large-scale traffic analysis can still reveal meaningful trends, provided that the missing or excluded data is not systematically biased. However, this assumption may not always hold: privacy-conscious users, for example, may differ in their behavior from those who consent to tracking, potentially distorting observed patterns.

Given these challenges in data accuracy, different tracking methods offer distinct advantages and trade-offs. In web analytics, data can be collected either through client-side tracking or via server logs, each offering a different perspective. Client-side analytics tools, such as Google Analytics, run in the user's browser and uses scripts to capture page views and events. This approach provides detailed insight into user interactions and typically ignores traffic that doesn't execute the tracking code, like many bots, but it may miss visits from users who disable scripts or block tracking. Server log data, on the other hand, is automatically recorded by the web server, logging every request to the site, so it includes all visits including those from users with blocked scripts and automated bots. Because of these differences, each approach has trade-offs in accuracy: server logs offer a more complete raw dataset but include non-human traffic that must be filtered out, whereas client-side data is pre-filtered for genuine users but can underestimate visits when tracking is blocked. For example, Suchacka and Chodak [24] used server log records to identify browsing patterns that signal a high probability of purchase in an online store, demonstrating that while server logs can yield rich insights, making sense of them may require additional processing compared to the more straightforward reports from client-side tools. (Analytics Market [2])

2.4 Conversion rate optimization

Conversion rate is considered one of the most important metrics for assessing the performance of an online store (Gudigantala et al. [36]). However, it should be analyzed in conjunction with other key performance indicators, such as total visitors, total sales, and market share, to gain a comprehensive understanding of a store's success. A high conversion rate alone does not guarantee profitability if the overall number of visitors is low.

The typically low conversion rates in e-commerce imply that the vast majority of visitors do not finalize a purchase. However, this does not necessarily indicate a failure on the part of the store. Many users visit online stores for reasons other than making an immediate purchase, such as researching products, comparing prices, or planning future purchases. Additionally, some visitors may arrive at the website by mistake, either while searching for unrelated content or by accidentally clicking on an advertisement. Despite this, a core objective of online retailers is to maximize the number of potential customers who complete a purchase. Conversion rate optimization refers to the practice of refining an online store's design, content, and user experience to increase the proportion of visitors who complete a desired action.

Besides that, there can be other goals in CRO, such as enhancing user satisfaction and long-term engagement. These are related to increasing customer loyalty and encouraging repeat visits, which are more closely associated with customer retention than with acquiring new customers. In this thesis, we do not consider these objectives and focus solely on the purchase rates.

CRO follows an iterative process (Saleh and Shukairy [48]). First, businesses analyze visitor behavior to identify common user journeys that lead to conversions. Next, they pinpoint critical points where users tend to exit without completing a purchase. Based on these insights, modifications are implemented, ranging from minor design tweaks to more substantial user flow adjustments aimed at reducing drop-offs. However, because website owners lack perfect insight into user behavior, the effectiveness of these changes must be validated through controlled experiments.

A/B testing is a commonly used method to evaluate these modifications systematically (King et al. [49]). In A/B testing, users are randomly assigned to different versions of a webpage, typically a control version (A) and a variant (B). The conversion rates of each version are then compared, and statistical tests are performed to determine whether any observed differences are significant. A/B tests rely on hypothesis testing, where the null hypothesis (H_0) assumes no difference between the two versions, and the alternative hypothesis (H_1) suggests a significant difference. The statistical significance of an A/B test is typically evaluated using a significance level (α), commonly set at 0.05. If the p-value of the test falls below α, the null hypothesis is rejected, indicating that the difference in conversion rates is unlikely to be due to random chance (King et al. [49]).

A well-designed A/B test must satisfy key statistical assumptions. First, the control and treatment groups must be comparable in all relevant characteristics to ensure a fair comparison. Second, the groups must be independent, meaning the experience of one user should not influence another. Additionally, A/B tests must account for potential statistical errors. Type I errors (false positives) occur when a difference is detected where none truly exists, leading to the mistaken implementation of ineffective changes. Type II errors (false negatives) occur when a real effect goes undetected, causing beneficial modifications to be overlooked. To minimize these risks, tests must maintain a sufficiently large sample size and statistical power to detect meaningful differences.

Despite these challenges, A/B testing remains one of the most reliable methods for optimizing conversion rates. It allows businesses to test new approaches with lower risk and assess their impact before fully implementing them. By comparing outcomes between randomized groups, A/B testing provides a basis for establishing whether a change actually affects user behavior. This causal interpretation can be reasonably assumed to hold if the statistical assumptions listed above are satisfied. When these conditions are met, any observed difference in conversion rates can be attributed to the change itself, rather than to other influences or random variation.

3 Modeling online store purchase decisions in the literature

3.1 Problem description

3.1.1 Definition of a classification problem

The problem of predicting the purchase decision of an online store visitor based on their actions can be formulated as a binary classification problem, where the task is to assign the visitor, based on their actions on the store prior to the purchase action or leaving the site, to one of two categories: purchase or no purchase. More generally, classification involves assigning an input vector $\mathbf{x}$, representing observed features, to one of a finite set of discrete classes. In the binary case, this means choosing between two mutually exclusive classes, often denoted C_1 and C_2.

A probabilistic formulation of classification aims to model the posterior distribution $p(C_k \mid \mathbf{x})$, where $k \in \{1, 2\}$. This represents the probability that the input belongs to class C_k given the observed data. The decision-making process then involves selecting the class that maximizes the posterior probability, which constitutes a rational decision under a 0–1 loss function, as it minimizes the probability of classification error. The 0–1 loss function assigns a loss of 1 to incorrect classifications and 0 to correct ones. This framework allows for a wide variety of models, including generative models that estimate the joint distribution $p(\mathbf{x}, C_k)$, and discriminative models that directly target the posterior. A deterministic formulation involves constructing a decision function $f : \mathbb{R}^d \to \{C_1, C_2\}$, where d denotes the dimension of $\mathbf{x}$, that assigns each input vector directly to a class label. (Bishop [10])

3.1.2 Ways to describe and solve classification problems

A simple approach to classification is the nearest-neighbor method, a nonparametric technique that requires no explicit model. Given a new input vector $\mathbf{x}$, the method searches in the set of input vectors for the k nearest neighbors, typically using Euclidean distance, and assigns the class label based on a majority vote among these neighbors. In the case of 1-nearest-neighbor, the prediction corresponds to the class label of the single closest example. The method does not require any assumptions about the underlying data distribution, allowing it to adapt to the observed data directly (Cover and Hart [17]). Although it can achieve low training error, defined as the proportion of misclassified examples in the training set, its performance on new data is sensitive to the choice of k, the distance metric, and the dimensionality of the feature space (Beyer et al. [8], Prasath et al. [71]). Nearest-neighbor classification can also be interpreted probabilistically by assuming the posterior class probability as the proportion of neighbors belonging to each class within the local region around $\mathbf{x}$ (Bishop [10]). While simple, this method can be problematic in high-dimensional settings, where the distinction between near and far neighbors diminishes (Beyer et al. [8]).

A straightforward way of solving classification problems is through linear discriminant functions or decision functions as defined above. In these methods, the input vector $\mathbf{x}$ is assigned to a class through a linear function. A simple example of a linear discriminant function is given by $y(\mathbf{x}) = \mathbf{w}^\top \mathbf{x} + w_0$, where $\mathbf{w}$ is a vector of weights and w_0 is bias, which determines the location of the decision surface. The input vector $\mathbf{x}$ is assigned to class C_1 if $y(\mathbf{x}) \geq 0$ and to class C_2 otherwise. One technique that builds on this formulation is least squares classification, which treats classification as a regression problem by assigning target values to class labels (typically $t = 1$ for C_1 and $t = 0$ for C_2). In this approach, t represents a fixed numeric value chosen to represent each class, allowing the use of regression to approximate class boundaries. The model parameters are then estimated by minimizing the squared difference between the model output $y(\mathbf{x})$ and the target t (Bishop [10]). However, using least squares in classification has some drawbacks. It is sensitive to outliers and can produce poor probability estimates, especially when class distributions, that is, the spread of feature values within each class, overlap significantly (Hastie et al. [39]). Because least squares assumes Gaussian noise with constant variance, it does not align well with the binary nature of classification targets (Bishop [10]).

If the dimensionality of the problem is high, it can be reduced by using Fisher's linear discriminant, which aims to find a linear projection of the input data that maximizes the separation between the classes (Murphy [66]). Specifically, the idea is to project the high-dimensional input vector $\mathbf{x}$ onto a one-dimensional space using a weight vector $\mathbf{w}$ such that the projected points from the two classes are well separated (Murphy [66]). The criterion for selecting $\mathbf{w}$ is to maximize the ratio of the between-class variance to the within-class variance of the projected data (Fisher [27]). This leads to an optimal direction for projection, represented by the weight vector $\mathbf{w}$, in which the classes are most distinct (Hastie et al. [39]). To classify a new point $\mathbf{x}$, it is assigned to one class if $\mathbf{w}^\top \mathbf{x}$ exceeds a threshold (typically the midpoint between the class means) and to the other class otherwise (Bishop [10]).

Classification problems can also be approached through probabilistic generative models, where the class-conditional probabilities $p(\mathbf{x} \mid C_k)$ and the prior probabilities of the classes $p(C_k)$ are modeled and used to compute the posterior probabilities $p(C_k \mid \mathbf{x})$ through Bayes' theorem (Raina et al. [74]). In the case of continuous input variables, a common choice is to model the class-conditional densities using multivariate Gaussian distributions (Bishop [10]). When assuming that all the classes share the same covariance matrix but have different means, the resulting posterior probability has a logistic sigmoid form, leading to a linear decision boundary (Bishop [10]). The parameters of these Gaussian distributions, namely, the means and shared covariance matrix, can be estimated from data using maximum likelihood principle estimation (Murphy [66]). This involves maximizing the likelihood function with respect to the parameters by computing sample estimates: the mean for each class is given by the empirical average of the data points in that class, the shared covariance is a weighted average over class-specific covariances, and class priors are estimated by the relative frequencies of each class in the training set (Murphy [66]). This approach yields a generative classifier that not only performs classification but also provides interpretable probabilistic models of the data distribution within each class (Bishop [10]).

In contrast to generative approaches, probabilistic discriminative models directly model the posterior probability $p(C_1 \mid \mathbf{x})$ without explicitly modeling the input distribution $p(\mathbf{x})$ (Murphy [66]). One of the most widely used discriminative models for binary classification is logistic regression, which assumes that the posterior probability can be modeled using the logistic sigmoid function applied to a linear combination of input features (Hosmer et al. [44]). The model has the form

$$p(C_1 \mid \mathbf{x}) = \sigma(\mathbf{w}^\top \mathbf{x}) = \frac{1}{1 + e^{-\mathbf{w}^\top \mathbf{x}}}, \tag{1}$$

where $\sigma(\cdot)$ is the sigmoid function and $\mathbf{w}$ is the weight vector learned from data (Hosmer et al. [44]). The parameters are determined by maximum likelihood estimation, which finds the weight vector that maximizes the likelihood of the observed labels (Murphy [66]).

Another approach in the same family is probit regression, which also models the posterior probability as a function of a linear combination of inputs, but uses the cumulative distribution function (CDF) of a standard normal distribution (Albert and Chib [1]):

$$p(C_1 \mid \mathbf{x}) = \Phi(\mathbf{w}^\top \mathbf{x}), \tag{2}$$

where Φ denotes the Gaussian CDF. While both logistic and probit models produce similar outputs in practice, they differ in their assumptions about the underlying noise distribution: logistic regression assumes a logistic noise model, whereas probit regression assumes Gaussian noise (Liu [54]). Both models provide smooth posterior estimates and linear decision boundaries in the feature space, making them suitable for a wide range of binary classification tasks (Hastie et al. [39]).

Neural networks offer a flexible and powerful framework for solving classification problems, particularly when the relationship between the input vector and class labels is highly nonlinear (Goodfellow et al. [31]). A typical neural network used for classification is a feed-forward neural network, which consists of a sequence of layers composed of functions, including one or more hidden layers and a final output layer trained to approximate a target function that maps inputs to class labels (Goodfellow et al. [31]). For binary classification, the output layer typically uses a logistic sigmoid function, producing a probability estimate $p(C_1 \mid \mathbf{x})$ analogous to logistic regression but with greater representational capacity due to the hidden layers (Goodfellow et al. [31]). The network parameters are trained by maximum likelihood and optimized via gradient-based methods such as backpropagation (see Werbos [86]), which calculates how the loss function changes with respect to each parameter and updates the parameters in a way that moves the model toward minimizing the loss (Rumelhart et al. [75], Goodfellow et al. [31]). Neural networks can approximate complex decision boundaries, making them well-suited to problems with high-dimensional or structured input data (Hornik et al. [43], LeCun et al. [52]). However, their flexibility comes at the cost of increased risk of overfitting, which is commonly addressed through regularization techniques such as early stopping, weight decay, or the incorporation of prior distributions in a Bayesian framework (Srivastava et al. [80], MacKay [55]).

Probabilistic graphical models can be used for representing and reasoning about complex probabilistic models using graphs to encode conditional dependencies between variables (Koller and Friedman [50]). In the context of classification, Bayesian networks, a class of directed graphical models, are especially useful for modeling the joint distribution of input variables and class labels in a structured and interpretable way (Ang et al. [3]). A Bayesian network defines a directed acyclic graph (DAG) in which each node corresponds to a random variable, and the edges represent direct probabilistic dependencies (Pearl [69]). The joint distribution is factorized into a product of local conditional distributions, allowing for efficient computation and inference (Koller and Friedman [50]). For classification tasks, a common structure places the class variable C as a parent of the observed input variables $\mathbf{x}$, reflecting the generative assumption $p(\mathbf{x} \mid C)$ (Ang et al. [3]). This facilitates the application of Bayes' theorem to compute posterior class probabilities $p(C \mid \mathbf{x})$, enabling classification decisions (Bishop [10]). Bayesian networks can incorporate domain knowledge through their structure and allow for missing data handling, inference under uncertainty, and integration with latent variables, making them suitable for complex classification problems (Koller and Friedman [50], Darwiche [18]). Inference in these models can be carried out using either precise algorithms or simplified approximations, with the choice depending on the complexity and structure of the network (Koller and Friedman [50], Darwiche [18]).

Clustering methods such as k-means and Gaussian mixture models (GMMs) can also be applied to classification problems (Jain et al. [46]). The k-means algorithm groups data points into clusters by assigning each one to the closest center, aiming to make the points within each cluster as similar as possible based on their squared distance from the center (Jain et al. [46]). However, this hard assignment approach is limited in its ability to capture uncertainty and is sensitive to initialization and outliers (Bradley et al. [11], Jain [47]). GMMs address these limitations by modeling the data as a mixture of Gaussian distributions, allowing for soft assignments where each point is associated with a probability of belonging to each cluster (McLachlan and Peel [60]). Nevertheless, these unsupervised clustering methods are not specifically designed for classification tasks and can struggle when class distributions overlap (Jain et al. [46]), as in the case where only a small fraction of users make purchases. In such cases, clustering may yield uninformative or misleading groupings, highlighting the importance of integrating external information, such as domain-specific knowledge, when classification accuracy is critical (Jain [47]). Clustering or segmentation can serve as a preprocessing step before modeling, helping to improve performance by dividing the data into more homogeneous groups (Kotsiantis et al. [51]).

An important class of solution approaches to classification problems involves Markov models (Rabiner [73], Murphy [66]). These models assume that the current state depends only on a limited history, often just the previous state, enabling tractable inference in time-series or sequential data. In classification tasks, hidden Markov models (HMMs) can be used to model the joint distribution of observed features and hidden states, offering a framework for problems involving temporally evolving behavior, such as user interactions in an online environment. These models are discussed in more detail in Section 3.3.1.

3.2 Purchase decision specific problem features

3.2.1 Factors explaining purchase decisions

Many studies have analyzed which user behaviors correlate with a higher likelihood of making a purchase in an online session. Generally, visitors who engage more deeply with the website tend to be more likely to buy. For example, a greater number of page views or product detail views in a session is often associated with a higher purchase probability (Hendriksen et al. [41], Huang and Van Mieghem [45]). Similarly, the time spent on the site or on key pages, indicative of interest in the products, shows a positive correlation with purchase intent (Huang and Van Mieghem [45]). In contrast, very brief sessions with only a couple of page views rarely lead to a sale.

Among all behavioral signals, shopping cart related actions are consistently found to be the strongest indicators of eventual purchase. Adding an item to the cart or wishlist is a clear expression of purchase intent, and sessions with such events have a much greater chance of ending in a purchase (Hendriksen et al. [41]). Other actions that reflect high engagement, such as repeatedly visiting the same product, using the site search, or starting the checkout process, likewise correspond to an increased probability of purchase. On the other hand, certain browsing patterns can signify a lower intent (for instance, excessive product comparison without cart addition).

Contextual and user-specific factors also play a role. Returning visitors (those who have visited the store before) are generally more likely to make a purchase than first-time visitors, especially if the return visit happens soon after a previous session (Hendriksen et al. [41]). The timing of the visit can matter as well: prior research has observed differences in purchase likelihood by time of day or day of week (e.g., shopping activity might peak in evenings or weekdays when users are more likely to purchase) (Hendriksen et al. [41]). Similarly, the device type used (desktop vs. mobile) has been linked to different purchase behaviors; for instance, desktop users often have higher average order values, whereas mobile users might browse more but purchase less frequently (Hendriksen et al. [41]).

3.2.2 Static vs. dynamic approaches

The binary classification problem of estimating the probability of a user making a purchase can be approached in two ways: as a static problem, where all data is available at once, or as a dynamic problem, where predictions are updated as new data becomes available. The selected approach has implications for model selection, as certain methods, particularly those designed for sequential or time-dependent data, are more naturally suited to the dynamic setting, while others assume fixed input representations and perform best in the static case. Unsupervised clustering methods and deep learning methods are more tailored for the static approach while probabilistic classifiers, such as hidden Markov models, and linear machine learning models, such as logistic regression, are well suited for the dynamic approach (Cirqueira et al. [15]).

The two approaches also differ in their business applications. In the static case, the online retailer receives an estimate of a user's purchase probability after their session, which can be used for follow-up actions such as marketing emails or targeted promotions. In the dynamic case, interventions suitable for the estimated purchase state can be made in real time while the user is still browsing the online store, allowing the retailer to take immediate actions aimed at increasing the likelihood of a purchase. The ways in which the store owners or marketers can utilize the estimated probabilities are discussed in more detail in Section 3.2.4.

3.2.3 Two-level approaches

The dataset used for building a model to predict purchase decisions typically includes full sessions. This provides the opportunity to tailor the model to a specific group of interest by using categorical segmentation, by user demographics for example, or a data clustering method in order to cater the model to the particular users. E-commerce visitors are a diverse population: different people may exhibit distinct browsing patterns, respond to site content in various ways, and have different propensities to buy (Suchacka and Chodak [81]). Therefore it is likely that many visitors do not have any intention of making a purchase and it would not be meaningful to estimate their purchase probabilities.

Chang et al. [13] address this challenge by introducing a customer anticipation model that segments the users based on purchasing behavior and personal attributes. Their approach utilizes clustering analysis to group loyal customers, identified by a metric called past purchasing tendency (PPT), according to demographic features such as age, gender, and education level. These clusters serve as behavioral profiles representing typical purchasing patterns. Potential customers are then matched to these clusters based on their own personal information, allowing the model to focus its predictions only on those whose profiles resemble those of known buyers. This use of clustering filters the customer base, concentrating predictive efforts on individuals who are behaviorally aligned with likely purchasers, and avoiding irrelevant data from visitors with no apparent purchase intent.

Suchacka and Chodak [81] pursue a similar goal of refining purchase prediction by focusing on meaningful subpopulations of users, but instead of applying clustering methods, they employ a rule-based segmentation strategy grounded in observed product preferences. They divide customers into two predefined groups, traditional and innovative, based on whether a user viewed only printed books or also browsed multimedia products like audiobooks and films. This distinction, informed by domain knowledge from the online bookstore, is used to separately determine association rules for each group, allowing the identification of session features that are strongly correlated with purchase decisions within each segment. By tailoring rules to specific user types, their approach effectively isolates high-intent sessions and mitigates the dilution of predictive patterns by low-engagement traffic. This segmentation enables more accurate and interpretable probability estimates while avoiding the added complexity and computational cost of clustering.

However, the importance of segmentation does depend on the method used for modeling. Some models, such as hidden Markov models and deep learning models, can internalize what segmentation would have achieved externally making segmentation less relevant (Rabiner [73]). In the case of HMMs this results from them utilizing hidden states that evolve over time based on observed user behavior, effectively capturing underlying structure and intent without requiring explicit segmentation of user groups (Rabiner [73]). Each hidden state can represent a distinct behavioral pattern or user type, allowing the model to differentiate between, for example, casual browsers and serious buyers through probabilistic inference (Rabiner [73]). On the other hand, methods such as regression can benefit significantly from segmentation, as fitting the model to a more homogeneous group of users typically leads to more accurate predictions (Kotsiantis et al. [51]). In the dynamic setting, the model begins with very limited information about the user and cannot reliably determine which specific model to apply. Therefore, it should be capable of internally inferring the user's underlying intent as more data becomes available in order to be effective at predicting the purchase decision.

3.2.4 Ways to affect purchase decisions

The primary motivation for predicting whether an online store visitor will make a purchase is to enable actions that can either dynamically influence the visitor's likelihood of purchasing during the session or guide tailored marketing efforts after the session. To maximize impact, different strategies should be applied to high and low purchase probability visitors.

For visitors who are estimated to have a high probability of making a purchase, the main objectives are to ensure they complete their purchase and to encourage them to buy as many items as possible. If they ultimately do not make a purchase, personalized marketing efforts can be directed toward them. Common in-session actions include providing navigation shortcuts to the checkout, product recommendations, and cart reminders (Esmeli and Gokce [22], Suh et al. [82]). These measures are designed to simplify the path to checkout and increase order value by recommending relevant products. These users are also an ideal target audience for advertising, as they already exhibit purchase intent, making advertising more likely to convert into actual sales (Yeo et al. [88]). Additionally, they can be prioritized for direct outreach by sales representatives over users with a low purchase probability (Habel et al. [37]). Examples of the specific actions by purchase probability and timing are given in Table 1.

For visitors who are estimated to have a low probability of making a purchase, the objective is to increase their likelihood of purchasing, either by making the act of ordering more appealing or by making the decision not to order less attractive. Since these users are considered unlikely to make a purchase, they can be offered special incentives, such as discount codes, that might not be given to others (Esmeli and Gokce [22]). The discount codes can also be provided to these users afterwards by email for example. The rationale is that it is more beneficial for them to make a purchase with reduced profit margins than not to purchase at all. To make not ordering less attractive, the store can, for example, display messages indicating that an item is almost sold out or implement a countdown timer that automatically empties the cart if the items are not purchased before it expires (Esmeli and Gokce [22]). A low purchase probability may also signal that the visitor needs assistance: perhaps in finding the right product or resolving concerns about shipping or store policies. In such cases, providing access to customer service or other forms of support could help address their hesitation (Mokryn et al. [63]). Additionally, these users may be given lower priority in follow-up marketing efforts to allow more focus on those with a higher likelihood of returning and completing a purchase (Habel et al. [37]).

Table 1: Example actions based on visitor purchase probability and timing.

Timing	High purchase probability	Low purchase probability
Real-time	- Navigation shortcuts to checkout - Product recommendations - Cart reminders	- Special discount codes or offers - Urgency cues (e.g., "almost sold out") - Live chat/help for support
Afterwards	- Personalized marketing (emails, ads) - Priority outreach by sales reps	- Discount codes - Deprioritized in follow-up efforts

3.3 Prevailing modeling approaches

As discussed in Section 3.1.2, classification problems can be described and solved in several different ways. In this section, we focus on the methods that have been particularly relevant in the context of predicting purchase decisions in recent literature. These prevailing approaches can be broadly categorized based on whether they take a Markovian modeling perspective, a machine learning approach, or a sequence modeling viewpoint. Each model category offers different strengths, depending on the structure of the user behavior data, suitability for real-time analysis and the needs for interpretability versus predictive accuracy.

3.3.1 Structural modeling: Markov approaches

Markov models provide a simple but effective framework for modeling sequential data, where the order of observations carries meaningful information and dependencies between elements must be captured explicitly (Rabiner [73]). These models are built on the Markov assumption, which states that the current state depends only on the previous state (Markov [57]). In classification problems involving sequences, such as user behavior, speech signals, or biological data, Markov models can be estimated for each class to capture the characteristic patterns of observation sequences associated with that class (Rabiner [73]). Each class C_k, where $k \in \{1, \ldots, N\}$ (here $N = 2$) can be associated with a separate first-order Markov chain, which defines a distribution over sequences based on a set of state transition probabilities and emission probabilities, assuming that the observations are generated by the model (Ghahramani [30]). Emission probabilities refer to the probabilities of observing particular actions given the current state. When classifying an observed sequence $\mathbf{x} = (x_1, \ldots, x_T)$, the likelihood $p(\mathbf{x} \mid C_k)$ is computed using the model associated with each class. Classification is then performed by selecting the class with the highest posterior probability, as determined via Bayes' theorem (Rabiner [73]).

Hidden Markov models extend Markov models by introducing hidden states, allowing them to model more complex, structured dependencies in sequential data (Rabiner [73]). In an HMM, each observation in the sequence is assumed to be generated by a hidden state, which evolves according to a Markov process (Ghahramani [30]). This makes HMMs especially powerful for classification tasks involving sequences that exhibit high variability or may be only partially observed (Ghahramani [30], Rabiner [73]). Estimating the model parameters, such as transition probabilities, emission probabilities, and the initial state distribution, requires iterative methods, even though exact inference in HMMs is tractable (Rabiner [73]). HMMs provide a balance between flexibility and computational efficiency, making them a widely used tool for sequence classification problems (Ghahramani [30], Rabiner [73]).

Markovian approaches, especially hidden Markov models, have proven effective for assessing online store visitors' purchase probabilities in real time, where the estimated likelihood is continuously updated as new events occur (Ding et al. [20], Montgomery et al. [65]). In this context, observable user behaviors, such as viewing a product, adding an item to the cart, or initiating checkout, are viewed as actions driven by an underlying sequence of hidden behavioral states. These hidden states may correspond to unobserved user intentions like casual browsing, price comparison, or high purchase intent. The state transition dynamics and emission probabilities allow the model to infer these hidden intent shifts and estimate the evolving likelihood of conversion as the session progresses. In their review, Cirqueira et al. [15] compare different models in their effectiveness of predicting purchase probabilities in suitability for real-time analysis, interpretability and sequential modeling and rate HMMs high on each category.

Montgomery et al. [65] use a hidden Markov model to represent user navigation behavior through an online retailer's website, modeling hidden user intent states and observed page views via a dynamic multinomial probit framework. Each user session is modeled as a sequence of page views, and the observed categories of pages (e.g., home, category, product, cart, order) form the emissions of the model. The hidden states are intended to represent unobserved user browsing modes, most notably, browsing-oriented and deliberation-oriented states, which correspond to different cognitive states or levels of purchase intent. Importantly for purchase probability estimation, Montgomery et al. [65] show that the hidden state sequence is highly informative of user intent, and transitions between states can indicate shifts from casual browsing to serious purchasing. They estimate the model using Bayesian Markov chain Monte Carlo (MCMC) methods and show that purchase predictions made after just six page views can reach over 40% accuracy, meaning that over 40% of sessions were correctly predicted as either purchase or non-purchase based on model output after six page views. Real-time purchase probabilities are computed through simulation of future paths, incorporating current state estimates and forecasting the likelihood that an order page will be reached before the session ends.

Ding et al. [20] extend the application of hidden Markov models to not only infer user purchase intent in real time but also to guide concurrent, optimal web page transformation aimed at reducing cart abandonment and increasing purchase conversion. Their approach treats user intent as a hidden cognitive state that evolves dynamically during the session and is inferred from a combination of observed behaviors, primarily shopping cart choices, such as adding or removing items, making purchases, or exiting the site. Each user session is modeled as a sequence of page-level decisions, and these decisions are assumed to be probabilistically generated by an underlying, unobserved sequence of intent states, modeled using a continuous-time hidden Markov model. Based on data from the American bookstore Barnes & Noble, the model reduces cart abandonment by 32.4% and increases purchase conversions by 6.9% when optimal interventions begin after only a few page views, demonstrating its effectiveness for intent prediction and dynamic personalization.

A limitation of traditional HMMs is their inability to account for the duration a user remains in a given hidden state, due to the Markov property: transitions depend only on the current state, not on how long it has been occupied. While observable metrics like time on page can be modeled, the duration spent within unobserved behavioral states remains unaccounted for, which may limit the model's ability to fully capture patterns relevant to purchase intent (Hatt and Feuerriegel [40]). To address this limitation, Hatt and Feuerriegel [40] developed a duration-dependent hidden Markov model (DD-HMM) to predict user exits during web sessions. This model extends conventional HMMs by allowing transitions between hidden states to depend on the time spent in the current state. The authors argue that prolonged duration in a particular hidden state can influence the likelihood of remaining in that state. For instance, a user who has spent a long time in a browsing state may be less likely to transition to a purchase-oriented state compared to someone who has spent only a brief time browsing. By relaxing the traditional Markov property, the model enables hidden state transitions to be duration-sensitive. This type of model is categorized as a hidden semi-Markov model because of the relaxation of the Markov property.

In this work, we are specifically interested in developing a model that can estimate whether a session will end in a purchase as new events occur. Because of their ability to capture hidden behavioral dynamics and sequential dependencies, we focus on Markovian approaches, specifically hidden Markov models. Hidden Markov and semi-Markov models are described in more detail in Section 4.6.

3.3.2 Black box modeling: machine learning and deep learning

In addition to Markov models, machine learning (ML) and deep learning (DL) approaches have become central to modeling user purchase intent in online environments (Cirqueira et al. [15]). ML models are algorithms that identify patterns in data to generate predictions or decisions without needing explicit programming for each task. Deep learning is a subfield of ML that uses multi-layered neural networks to model complex relationships in data. While both paradigms fall under the broader umbrella of predictive modeling, they differ in key respects. Traditional machine learning models generally rely on feature engineering, that is, the manual construction of input variables that capture relevant patterns or domain knowledge from raw data, and operate on structured input using models such as decision trees or logistic regression. In contrast, DL models, especially those using neural network architectures, learn representations directly from raw or minimally processed input, often excelling in tasks involving sequential, high-dimensional, or unstructured data. (Goodfellow et al. [31])

Machine learning approaches offer a flexible and scalable framework for predicting online purchase behavior by leveraging high-dimensional features derived from user interactions. Unlike Markovian models, which are constrained by explicit state transition structures, ML models focus on learning direct mappings from session features to purchase outcomes. Machine learning methods, particularly discriminative models like logistic regression, support vector machines (SVMs), and neural networks, are powerful tools for pattern recognition, enabling the modeling of complex, nonlinear relationships without requiring detailed assumptions about data generation processes (Bishop [10]).

ML models applied for purchase prediction in previous work range from feature-based methods such as boosted decision trees to neural architectures designed for sequential data, such as recurrent neural networks (Hendriksen et al. [41]). These models can incorporate a wide array of features including page views, dwell time, device type, and user history. Feature-based methods are valued for their interpretability and robustness to heterogeneous input data, while neural networks are favored for their ability to capture temporal dependencies and nonlinear patterns in high-volume datasets.

Hendriksen et al. [41] provide a large-scale empirical evaluation of ML-based purchase predictors using over 95 million anonymized sessions from a major European e-commerce platform. They distinguish between anonymous and identified sessions, building separate feature-based models for each. For anonymous sessions, where user history is unavailable, models rely exclusively on dynamic, session-level features such as device type, number of visited pages, and channel of entry. For identified sessions, they enrich the feature space with historical indicators like prior purchase frequency and time since last order. The authors construct multiple classifiers, including logistic regression, decision trees, gradient boosting machines, and neural networks, and show that in the anonymous setting, tree-based models and neural networks outperform support vector machines, logistic regression and k-nearest neighbors, achieving an F1 score improvement of over 17.54%. F1 score is a metric that balances precision and recall and is calculated as the harmonic mean of the two. For identified users, the addition of historical features enables ML models to achieve up to 96.2% F1 accuracy on held-out data. Their work highlights how feature importance evolves during the session: dynamic features grow more predictive over time, especially in anonymous settings where no prior context exists.

In contrast to purely supervised classifiers, Nishimura et al. [68] propose a latent-class machine learning model tailored to product-choice prediction. Their method uses clickstream data to estimate purchase probabilities for individual products, while incorporating behavioral assumptions about how viewing behavior relates to purchase likelihood. Building on prior work using monotonicity, convexity, and concavity (MCC) constraints in regression, the authors introduce a latent-class variant that accounts for product heterogeneity, such as differences in price sensitivity or purchase frequency across product types. Their approach classifies products into latent groups and fits separate MCC models to each, estimating class-specific product-choice probabilities as a function of the recency and frequency of product page views. This model is trained using an expectation-maximization algorithm and is shown to outperform both standard MCC models and latent-class logistic regression in predictive accuracy.

Neural networks are a flexible class of models that can learn complex, nonlinear relationships from data, including subtle patterns that traditional models may miss (Bishop [10]). For predicting purchase intent over a user session, architectures designed for sequential input, such as recurrent neural networks (RNNs), are especially useful because they can account for how user behavior unfolds over time. These models do not require manually defined hidden states or fixed transition assumptions, as in HMMs, and instead estimate hidden representations and transition dynamics directly from data. These models are estimated using algorithms like backpropagation through time (see Werbos [86]), which allows them to learn from extended behavioral sequences. To address difficulties in estimating these models on long sequences, such as the tendency to forget earlier actions or to become unstable when trying to learn long-range patterns, more advanced architectures like gated recurrent units (see Cho et al. [14]) and long short-term memory networks (see Hochreiter and Schmidhuber [42]) have been introduced. These designs help the model retain important information over time while filtering out less relevant details (Bishop [10]).

Sheil et al. [77] propose an RNN-based framework for predicting user purchasing intent using anonymous session data. Their model is designed to eliminate the need for extensive domain-specific feature engineering, a common requirement in tree-based models like gradient boosted machines (GBMs). Instead of manually constructing features, they represent user actions (such as item views or cart events) as input sequences and use deep RNNs to learn patterns that capture both short-term and broader behavioral dynamics. The design includes techniques to help the model perform well across sessions of varying length. Their results suggest that deep learning models can be effective for predicting purchase intent from anonymous session data, even in settings with strong class imbalance.

Tang and Wang [83] propose a model called Caser that applies deep learning to model user behavior over time. While not originally designed for binary classification, Caser is relevant to purchase intent modeling as it captures both long-term user preferences and short-term action patterns within a session. The model treats a user's sequence of interactions as a structured input and uses neural network components to detect recurring patterns, such as recent clicks or combinations of earlier actions that may influence later behavior. It also allows for flexibility in modeling situations where earlier behaviors affect outcomes even if they are not immediately followed by related actions. Compared to more traditional approaches, Caser is designed to better capture complex behavior sequences. In evaluation tasks involving product recommendation, the model consistently outperforms several established methods, suggesting its potential for broader behavioral prediction problems.

4 Predicting purchases in an online clothing store

4.1 Setup

In this section, we develop a model to predict whether a visitor will make a purchase in a Finnish online clothing store that specializes in exclusive sneakers and streetwear. The store features over 300 products, including a wide selection of shoes, clothing, and accessories. Its website follows a typical e-commerce structure, with a strong emphasis on product presentation and carefully designed navigation to help users easily locate items.

We formulate the prediction of a purchase event as a binary classification problem, as described in Section 3.1.1. User behavior on the online store is captured through timestamped events, each associated with a unique user ID. These events are used to construct input vectors representing individual user sessions. Each sequence of user actions is assumed to be generated by an underlying sequence of hidden user intent states, which evolve over time and are modeled using hidden semi-Markov models (HSMMs). Classification is based on the observed input vector as well as the duration a user spends in each hidden state, a key feature of HSMMs. Prior research by Hatt and Feuerriegel [40] has demonstrated the effectiveness of this temporal modeling approach in predicting user exits in online stores, suggesting that state duration could also serve as a valuable predictor for purchase behavior.

The modeling process is based on clickstream data collected over a period from January 21 to May 30, 2025. Clickstream data consists of all the actions a user takes while interacting with a website. The resulting dataset has 29 597 443 rows and contains 618 052 timestamped events generated by 42 626 unique users. Each event is linked to an anonymized user ID, allowing the reconstruction of complete behavioral sequences across user sessions. The HSMMs are estimated using the observed user event sequences, capturing both the types of actions (e.g., product views, cart additions) and the duration spent in underlying hidden states. This temporal modeling approach allows the system to account for the time users remain in each intent state, providing a more nuanced view of session progression than memoryless models like standard HMMs.

4.2 Data collection

In accordance with GDPR requirements, visitor consent was obtained via a cookie banner prior to initiating data collection. Visitor interactions on the website were tracked using Google Tag Manager (GTM) [33] and Google Analytics 4 (GA4) [32], with the collected data stored in BigQuery [34]. The tracking setup was implemented by integrating a custom pixel into the store's Shopify platform. A pixel is a small piece of code embedded in a webpage that sends information about user behavior to analytics or advertising platforms. Because data is only recorded for users who accept tracking cookies, the dataset may reflect a subset of visitors, introducing potential selection bias.

The pixel was configured to record specific user actions, such as viewing a product, adding an item to the cart, or completing a purchase. Each interaction generated an event containing a unique user identifier and contextual information related to the action. Table 2 lists the types of events that were collected during user sessions.

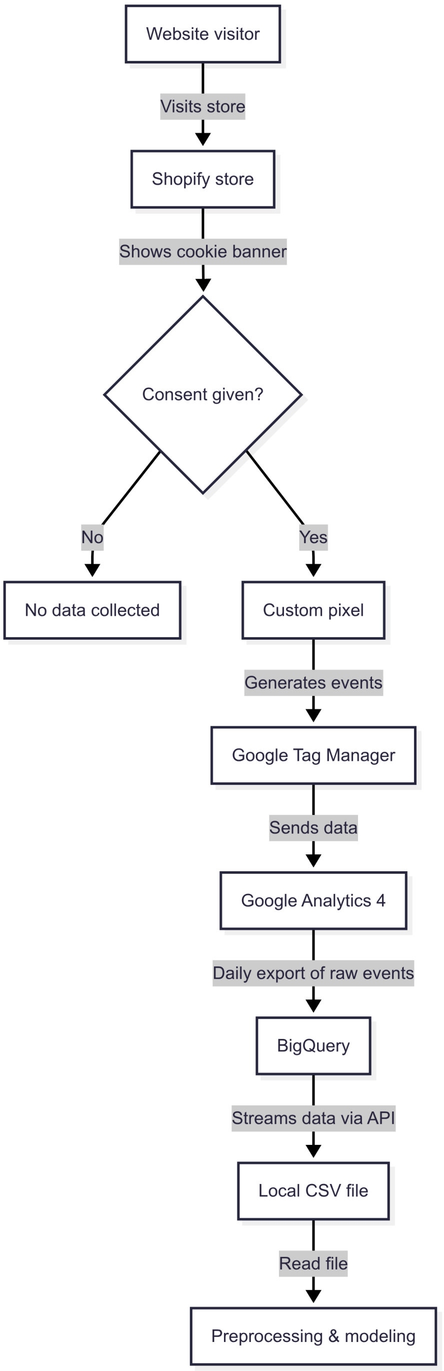


Figure 1: The data collection process.

Once recorded, the events were transmitted to GA4 and subsequently exported to BigQuery, where the data was stored for further processing. As GA4 does not provide direct access to raw event-level data in its interface, the export to BigQuery was configured to occur on a daily basis. From there, the dataset was retrieved as a local CSV file using Python 3.11.9 and the google-cloud-bigquery library (version 3.31.0) to allow for preprocessing and modeling. The data collection process is illustrated in Figure 1. While this workflow could have been managed entirely within Google Cloud Platform (GCP), a local environment was used to reduce cloud computing costs. The implementation followed publicly available guidance for integrating GA4 with GTM and Shopify [25].

Table 2: The events collected from the online store and their descriptions.

Event name	Description
first_visit	User visits the site for the first time
session_start	User first enters the site
page_view	A page on the site is viewed
menu_click	A link in the store menu is clicked
view_search_results	An on-site search is performed
view_item_list	A collection page is opened
product_click	A product card is clicked
view_item	A product page is opened
add_to_cart	An item is added in the shopping cart
view_cart	Shopping cart page is opened
remove_from_cart	An item is removed from the shopping cart
begin_checkout	User enters the checkout page
add_shipping_info	Shipping information is filled on checkout
add_payment_info	Payment information is filled on checkout
purchase	A purchase is completed

The number of events sent daily from Google Analytics 4 to BigQuery is shown in Figure 2. These are all the actions taken by different users in the online store each day. The daily volume ranges from 3,327 to 7,330 events. While there is noticeable day-to-day variation, no prolonged periods of unusually high or low activity are observed. This suggests that there were no errors in sending data from Google Analytics 4 to BigQuery. The short-term peaks likely correspond to specific campaigns or promotions. A decline in mid-February coincides with the winter holiday period, and lower event counts at the end of April align with the Easter holiday.

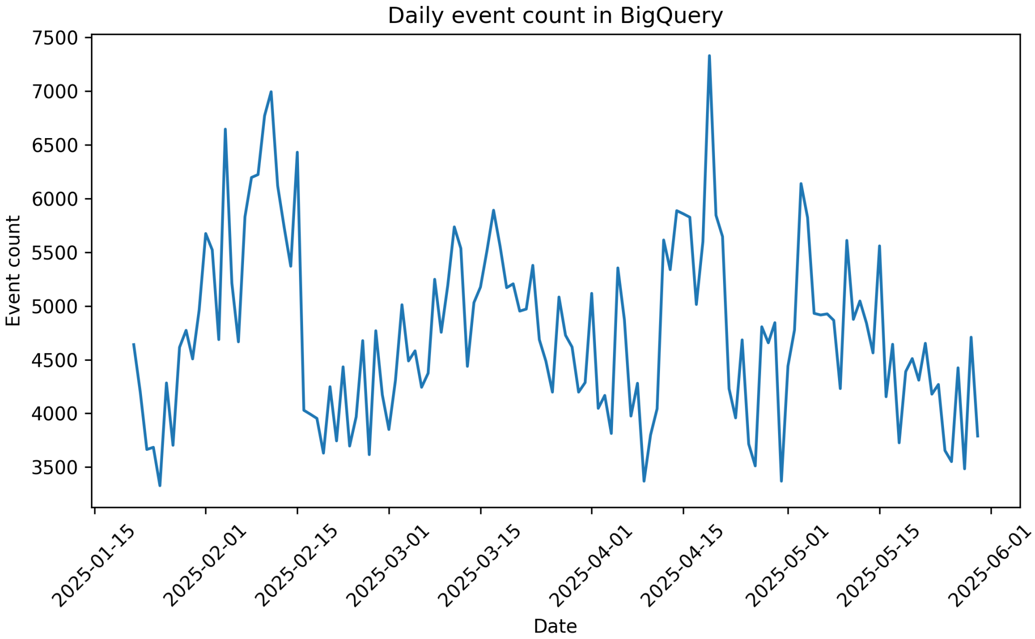


Figure 2: The daily number of events sent to BigQuery.

4.3 Data structure

User interaction data is recorded as a series of timestamped events. Each row in the table corresponds to a single event performed by a user on the website. Events are identified by a name (e.g., `page_view`, `product_click`) and are associated with an anonymized user identifier. Each event is accompanied by two types of metadata: event parameters and contextual variables. Event parameters are key-value pairs that provide additional information about the event itself. For example, a `page_view` event, triggered when a user loads a page, includes parameters such as the page URL, campaign identifiers, and session-related metadata. Table 3 provides an overview of typical parameters associated with a `page_view` event. Contextual variables describe the environment in which the event occurred. These include general information about the user's device (e.g., mobile vs. desktop) and geographical location (e.g., city, country).

The event parameter values can hold values of different data types, including strings, integers, floats, and doubles. The parameters are stored as nested records in the database. Each event contains a repeated field called `event_params`, where each entry includes a `key` and a nested `value` object. The `value` object contains one or more type-specific fields, depending on the data type of the parameter. This schema is illustrated in Table 4.

Figures 3 and 4 illustrate the same page view event from the Google Analytics 4 export to BigQuery in two different table schemas. In the raw schema (Figure 3), all event parameters are stored as nested fields within a single row. In contrast, the flattened schema (Figure 4) expands these parameters so that each key–value pair

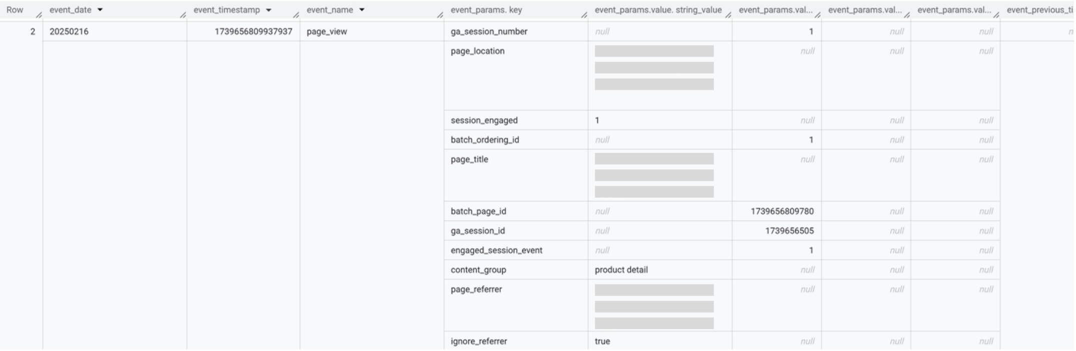
Table 3: The set of event parameters sent with a `page_view` event.

Parameter key	Description
`entrances`	Indicates whether a page is the first page viewed in a session
`page_location`	The URL of the visited page
`source`	From where the user came from, e.g., Google
`batch_ordering_id`	The number of network requests sent from the page
`ga_session_id`	A unique ID for the session
`srsltid`	Identifier parameter that is added if the user entered the site through an advertisement in Google Merchant Center
`page_title`	The title of the page
`content_group`	The category of the page, such as home or collection
`ga_session_number`	The number of sessions the user has had on the website
`session_engaged`	Boolean flag indicating whether a user actively engaged with the website
`medium`	The traffic medium through which a user arrived at the site, such as email
`batch_page_id`	Used to identify a specific page within a batch of page views
`campaign`	The marketing campaign through which a user arrived at the site
`term`	The search keyword or phrase that a user entered in a paid search campaign
`page_referrer`	The URL of the previous page the user visited
`engaged_session_event`	A parameter indicating whether a user actively engaged with the website
`ignore_referrer`	Used to indicate whether the referrer information should be disregarded
`content`	The specific element or creative content within a campaign or ad that led the user to the website
`campaign_id`	The ID of the campaign through which a user arrived at the site

occupies its own row. The grey rectangles are used to hide the company name that appears in some fields. While the nested schema is more efficient for storage and querying within BigQuery (see [35]), it complicates data processing outside that environment. Flattening the data makes it easier to process outside of BigQuery but results in redundancy and increased storage size. For example, the size of the nested table for events between January 21 and May 30, 2025, is 1.21 GB, while the flattened table for the same period occupies 6.44 GB, even though many columns were excluded from the flattened table.

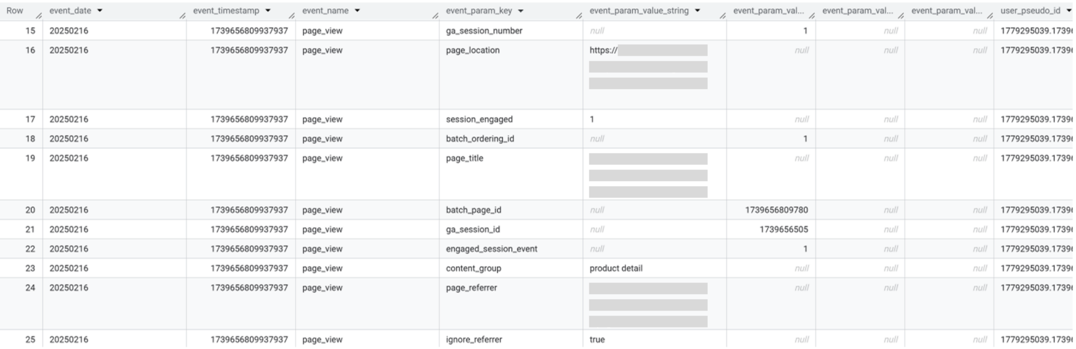
Table 4: User interaction data schema structure.

Field name	Description
event_timestamp	Timestamp of the event
event_name	Name of the event (e.g., page_view)
event_params	Repeated record of event parameters
key	Name of the parameter
value	Record containing the parameter value
string_value	Value as a string
int_value	Value as an integer
float_value	Value as a float
double_value	Value as a double
device	The device information
category	The device category: mobile, desktop etc.
...	
geo	The location information
city	The city the user is in
...	



Row	event_date	event_timestamp	event_name	event_params. key	event_params.value. string_value	event_params.val...	event_params.val...	event_params.val...	event_previous_ti
2	20250216	1739656809937937	page_view	ga_session_number	null	1	null	null	n
				page_location	[redacted]	null	null	null	
				session_engaged	1	null	null	null	
				batch_ordering_id	null	1	null	null	
				page_title	[redacted]	null	null	null	
				batch_page_id	null	1739656809780	null	null	
				ga_session_id	null	1739656505	null	null	
				engaged_session_event	null	1	null	null	
				content_group	product detail	null	null	null	
				page_referrer	[redacted]	null	null	null	
				ignore_referrer	true	null	null	null	

Figure 3: A page view event in the raw BigQuery event table exported from Google Analytics 4.



Row	event_date	event_timestamp	event_name	event_param_key	event_param_value_string	event_param_val...	event_param_val...	event_param_val...	user_pseudo_id
15	20250216	1739656809937937	page_view	ga_session_number	null	1	null	null	1779295039.1739
16	20250216	1739656809937937	page_view	page_location	https://[redacted]	null	null	null	1779295039.1739
17	20250216	1739656809937937	page_view	session_engaged	1	null	null	null	1779295039.1739
18	20250216	1739656809937937	page_view	batch_ordering_id	null	1	null	null	1779295039.1739
19	20250216	1739656809937937	page_view	page_title	[redacted]	null	null	null	1779295039.1739
20	20250216	1739656809937937	page_view	batch_page_id	null	1739656809780	null	null	1779295039.1739
21	20250216	1739656809937937	page_view	ga_session_id	null	1739656505	null	null	1779295039.1739
22	20250216	1739656809937937	page_view	engaged_session_event	null	1	null	null	1779295039.1739
23	20250216	1739656809937937	page_view	content_group	product detail	null	null	null	1779295039.1739
24	20250216	1739656809937937	page_view	page_referrer	[redacted]	null	null	null	1779295039.1739
25	20250216	1739656809937937	page_view	ignore_referrer	true	null	null	null	1779295039.1739

Figure 4: A page view event in the flattened BigQuery event table exported to a local CSV file.

4.4 Data preprocessing

Clickstream data is high-dimensional and often irregular, with sessions varying significantly in length and content, making preprocessing a necessary step. From a statistical standpoint, the heterogeneity complicates comparisons across sessions, as many modeling approaches assume inputs with a uniform structure. Preprocessing helps by reducing variability and organizing the data into a format where patterns in user behavior can be more reliably identified and analyzed. As described in Section 4.3, the exported data includes nested fields, most notably the event parameters, which must be unnested to produce a flat, tabular format suitable for analysis. Table 5 presents the schema of the flattened dataset used in this study, after unnesting the nested fields from the GA4 export using SQL. The structure and complexity of this data require tailored preprocessing steps, which vary depending on the data source and the types of events involved. In this case, preprocessing was designed for client-side clickstream data collected via Google Analytics 4 and exported to BigQuery. Key preprocessing tasks include constructing sessions, filtering out bots and anomalous behavior, selecting relevant events, and generating labels. We use Python 3.11.9 and the `pandas` library (version 2.2.3) (McKinney [59]) throughout the preprocessing. The objective is to transform the raw event stream into a clean, structured, and model-ready format that preserves the sequential and contextual properties necessary for downstream modeling tasks.

Session construction refers to the process of grouping all events generated by a specific user within a defined time window, often 30 minutes, and arranging them chronologically. This sequence of events captures the user's navigation path and event history throughout their visit to the online store, forming the basis for modeling behavior as a sequence of transitions between states, an assumption central to Markov chain-based approaches. In earlier studies, session construction was typically performed using server log data, which required researchers to infer user identity based on semi-unique identifiers such as IP addresses and to reconstruct event order from server request timestamps (Montgomery et al. [65], Sismeiro and Bucklin [78]). However, server log data has known limitations. Many modern websites implement caching mechanisms, which allow pages to be served from the browser's or intermediary proxy's memory when revisited, bypassing the server entirely. As a result, repeated views of the same page may not be recorded in server logs. Furthermore, IP-based user identification is unreliable: multiple users within the same network (e.g., a household or office) may share an IP address, and individual users may appear under different IPs over time due to dynamic allocation or mobile usage.

By contrast, client-side data collection, such as that performed through Google Analytics 4, captures events directly in the user's browser. This approach reliably logs every interaction, regardless of whether a server request was made, and typically assigns a unique user ID, such as `user_pseudo_id` in Table 5, at the time of first visit, assuming the user consents to tracking via cookies. This ID is stored in the browser and attached to each event, enabling straightforward and accurate session construction without relying on IP addresses. Nevertheless, this identifier is browser-specific, meaning a user visiting the site from multiple devices or using incognito mode will

Table 5: The initial dataset schema.

Name	Type
event_date	STRING
event_timestamp	INTEGER
event_name	STRING
event_param_key	STRING
event_param_value_string	STRING
event_param_value_int	INTEGER
event_param_value_float	FLOAT
event_param_value_double	FLOAT
user_pseudo_id	STRING
user_first_touch_timestamp	INTEGER
device_category	STRING
mobile_brand_name	STRING
device_language	STRING
geo_city	STRING
geo_country	STRING
traffic_source_name	STRING
traffic_source_medium	STRING
traffic_source_source	STRING
item_id	STRING
item_name	STRING
quantity	INTEGER
price	FLOAT

be assigned different IDs, fragmenting their activity across separate user profiles. Conveniently, we can use the user_pseudo_id and event_timestamp to construct the sessions where we define a session as a sequence of events generated by a user where the time difference between two subsequent events is less than 30 minutes. Google Analytics and other analytics platforms often use 30 minutes as the default session timeout which is why it was chosen.

Virtually all websites are regularly scanned by automated bots for purposes such as indexing, monitoring, or scraping. In server-side log data, these bots generate page requests that are indistinguishable from those made by human users unless explicitly filtered, leading to potential contamination of behavioral datasets. By contrast, client-side data collection, such as that used in this study, inherently filters out most bot traffic. This is because tracking scripts, typically loaded and executed via JavaScript, are not triggered by bots, which often do not render or execute JavaScript. As a result, bot activity is largely absent from datasets collected using tools like Google Analytics 4, eliminating the need for separate bot detection and filtering procedures.

However, anomaly detection remains an important preprocessing step. Sessions that are excessively short (e.g., a single event) or abnormally long may not contain meaningful behavioral signals and can distort model estimation or evaluation. Following practices noted in the literature, these anomalies are typically filtered out by applying thresholds on the number of page views or events per session (Bigon et al. [9]). This ensures that only sessions with sufficient behavioral information are retained for analysis. Following the preprocessing conducted in Montgomery et al. [65] and Hatt and Feuerriegel [40] we decide to exclude sessions consisting of less than three and more than 50 page views. This leaves us with 24 252 valid sessions in our dataset.

To identify the sessions that end in purchase, the session outcome is encoded as a binary label indicating whether a purchase occurred. Information such as transaction value or item identity were not used in this analysis. This simplification reduces target complexity and allows the model to focus solely on whether the purchase occurred rather than the characteristics of the purchase.

Of the 24 252 sessions in the dataset, 252 are labeled as purchase sessions. This level of class imbalance is common in online retail data and has to be taken into account in the modeling process. The impact of class imbalance can be mitigated before modeling by oversampling the minority class, undersampling the majority class or using different error weights for different classes (Menon et al. [61]). The decision threshold, the probability value above which a session is classified as a purchase, can also be adjusted for a probabilistic classifier to account for class imbalance. That is, instead of using a default threshold (e.g., 0.5), the threshold can be shifted to reflect the fact that purchase events are much less common than non-purchase events (Menon et al. [61]).

The events in Table 2 include some events that are closely related to a purchase event, namely `add_shipping_info` and `add_payment_info`. These correspond to the user filling out their information in the checkout which is the last required step before completing a purchase. Since these events occur so close to the purchase itself, they offer limited utility for proactive interventions and are excluded from the predictive model.

4.5 Reduction of dimensionality

To reduce the complexity of the input space, we exclude certain event types and contextual variables from the modeling data. Many of the events listed in Table 2 exhibit high correlation with one another, which can introduce multicollinearity and compromise model stability. Moreover, high-dimensional sets of input variables can increase model variance and reduce generalization performance, especially when sample sizes are limited. Variable selection helps mitigate these risks by eliminating redundancy and focusing on behaviorally informative signals. At the same time, care must be taken to avoid discarding relevant patterns that contribute to purchase decisions.

The `page_view` event and its associated parameters are retained due to their central role in capturing user navigation patterns. Previous studies (e.g., Montgomery et al. [65], Hatt and Feuerriegel [40]) have demonstrated the effectiveness of categorizing pages into broader groups, such as product, collection, or cart pages, rather than modeling each URL or product individually. This process can be viewed as a form of discretization, where a high-cardinality feature is transformed into a smaller set of interpretable categories that still capture meaningful user intent. By leveraging `page_view` data, it is possible to infer visits to product pages, search result pages, and even transitions to the checkout and thank-you pages, the latter of which indicates a completed purchase. However, some key events, such as `add_to_cart` and `remove_from_cart`, cannot be inferred from page views because they represent actions that are taken within a page so they are not removed from the data.

Based on the structure of the online store, we define the following page categories under which we group the pages: Home, Info, Search, Collection, Product, Cart, Checkout, Order. The category Info refers to all pages not included in other categories, such as shipping terms, privacy policy, and contact information. The categories are assigned based on the page URL, for example, if a page URL includes `/products/` we assign it to category Product. By adopting this categorization, we lose the ability to compare how individual products or collections influence purchase decisions, as each is represented by a shared label. In a store with hundreds of products, such comparisons would require substantially more data than is currently available in order to yield statistically meaningful results. While the contextual features in Table 5 are useful for comparing behavior across user segments (e.g., comparing mobile versus desktop users), they are excluded from the modeling features to avoid overly fragmenting the dataset and reducing the sample size within each segment.

Another way to reduce the dimensionality of the model is to use a parametric distribution for the durations. We use the discrete Weibull distribution for this purpose. This choice is motivated by prior work (Hatt and Feuerriegel [40]), which demonstrates that the discrete Weibull can closely approximate a wide range of duration patterns. Unlike the geometric distribution, implicitly assumed in standard HMMs, the Weibull can accommodate both light- and heavy-tailed behaviors through its shape parameter, highlighting its potential in modeling real-world temporal dynamics. The discrete Weibull distribution is defined over positive integers $d = 1, 2, \dots$ with probability mass function:

$$P(D = d) = (1 - \theta)^{(d-1)^c} - (1 - \theta)^{d^c}, \tag{3}$$

where $\theta \in (0, 1)$ is the scale parameter and $c > 0$ is the shape parameter. The shape c controls the "stickiness" or tail behavior of the duration, with $c < 1$ producing heavy tails and $c > 1$ producing lighter ones (Nakagawa [67]).

The summary statistics for the preprocessed dataset are presented in Table 6. On average, a session lasts approximately 3.99 minutes and includes 7.52 page views. Most user activity is concentrated in the Collection and Product categories, which account for an average of 3.83 and 2.28 page views per session, respectively. In contrast, transactional categories such as Cart, Checkout, and Order are the least visited. Interestingly, Checkout is accessed more frequently than Cart, despite appearing later in the purchase process. This can be explained by the store, like many other stores, offering a quick checkout button on the product pages which takes the user straight to checkout without visiting the shopping cart.

Table 6: Summary statistics of the dataset.

	Mean	**Std. dev.**
Session duration (in minutes)	3.99	7.31
Page views per session	7.52	6.21
Add to cart events per session	0.05	0.40
Remove from cart events per session	0.01	0.18
Category views per session		
CART	0.04	0.31
CHECKOUT	0.06	0.43
COLLECTION	3.83	4.03
HOME	0.27	0.68
INFO	0.84	1.17
ORDER	0.01	0.15
PRODUCT	2.28	2.85
SEARCH	0.19	0.81

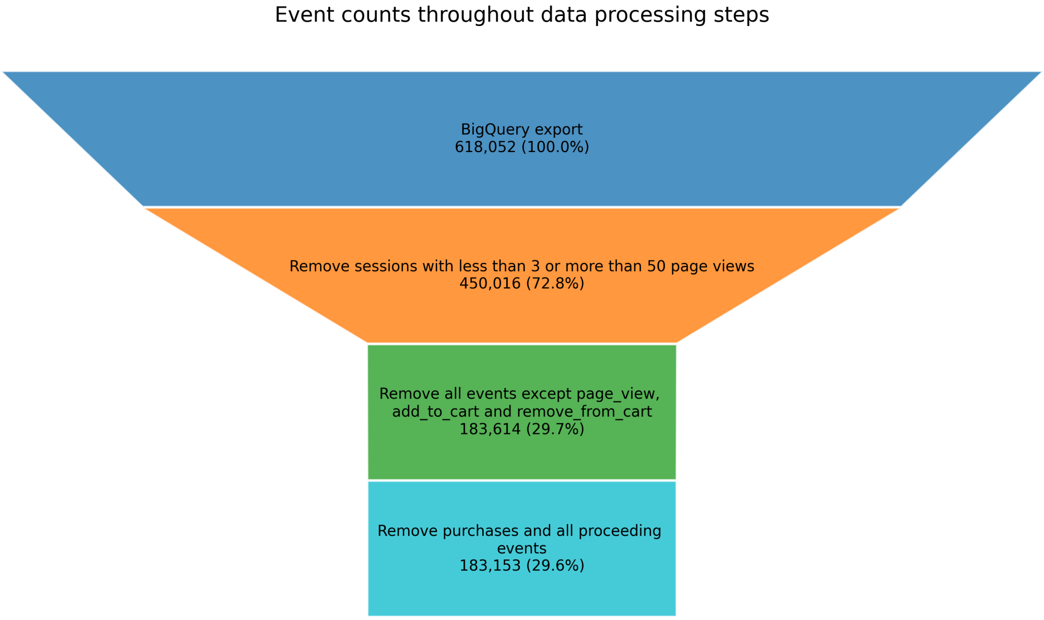


Figure 5: The total number of events after data processing steps.

The impact of data processing steps on session and event counts is illustrated in Figures 6 and 5. As expected, the number of events decreases at each step, since each stage involves filtering or discarding certain event types. In contrast, the session count only decreases at the step where sessions with fewer than three or more than fifty page views are removed. Subsequent filtering of events does not affect the number of sessions retained, only the contents of the sessions. The final dataset contains only 29.6% of the original events, highlighting the high-dimensional and irregular nature of clickstream data.

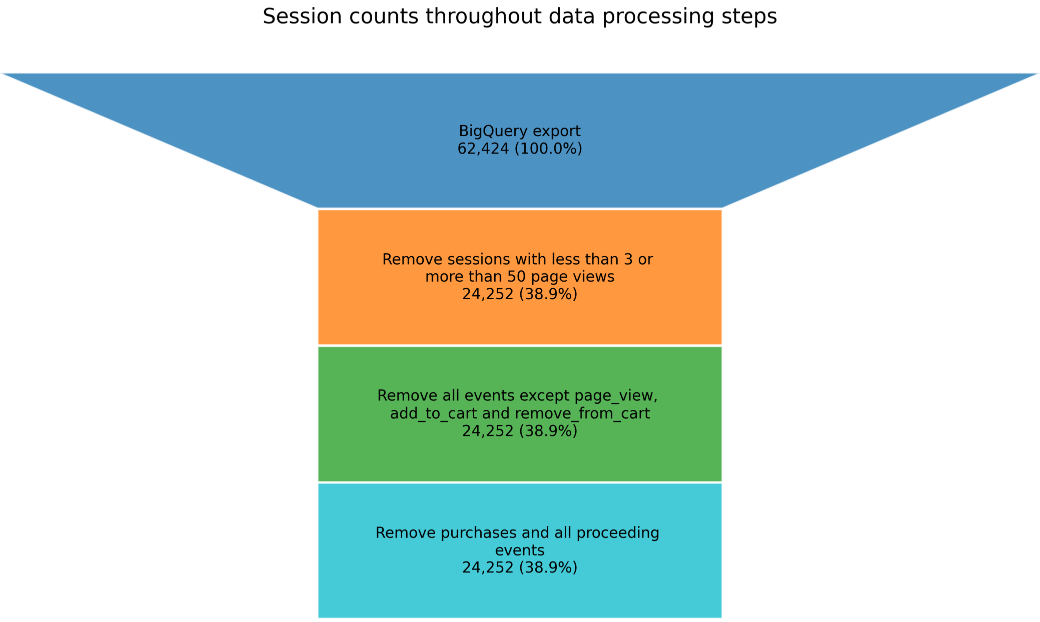


Figure 6: The total number of sessions after data processing steps.

4.6 Hidden semi-Markov model

We now return to the description of hidden Markov and semi-Markov models. A hidden Markov model is a generative probabilistic model for sequential data in which the observed sequence $O = \{O_1, O_2, \ldots, O_T\}$ is assumed to be generated by a sequence of unobserved (hidden) states $\mathcal{S} = \{S_1, S_2, \ldots, S_T\}$, where each $S_i \in \{0, \ldots, K-1\}$ is drawn from a finite state space and K is the number of distinct states. For example, if $K = 4$, a sequence of unobserved states could be $\{0, 0, 1, 3, 3, 2\}$. Figure 7 demonstrates this generative process. The model is defined by an initial state distribution $P(S_1)$, a state transition matrix $P(S_{i+1} \mid S_i)$, and an emission distribution $P(O_t \mid S_i)$. The sequence of hidden states forms a Markov chain, where the next state depends only on the current state. The state transition matrix describes the probability of transitioning from one hidden state to another. The emission distribution models the likelihood of a specific observation given the current hidden state and thus models the relationship between the hidden behavioral state and the observed user action. A key assumption in HMMs is that the hidden state transitions follow a first-order Markov process, and that the duration in each state is implicitly governed by a geometric distribution due to the memoryless nature of the transitions.

A hidden semi-Markov model generalizes the hidden Markov model by explicitly modeling the duration $d_i \in \mathbb{N}^+$ that the system remains in each hidden state S_i, where $i \in \{1, \ldots, N\}$. This extension is illustrated in Figure 8. The duration refers to the number of observations assigned to a hidden state and is therefore a positive integer. The HSMM defines a segmentation of the observation sequence $O = \{O_1, O_2, \ldots, O_T\}$

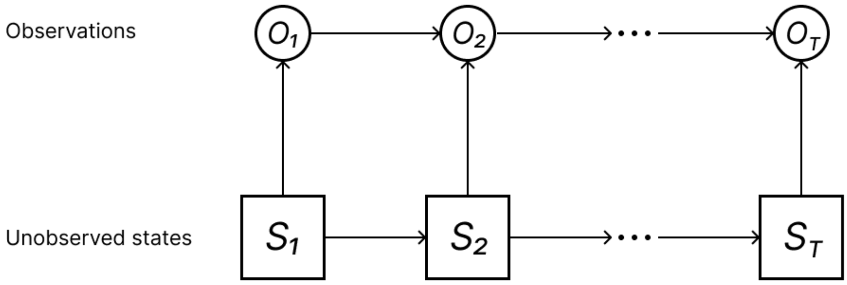


Figure 7: Hidden Markov model process.

into N non-overlapping, contiguous segments, where each segment corresponds to a single hidden state that persists for d_i consecutive time steps. The generative process can be described mathematically in the following way. Let t_i denote the starting time of the i-th segment, with $t_1 = 1$ and $t_{i+1} = t_i + d_i$. The generative process is defined as:

$$S_i \sim P(S_i \mid S_{i-1}), \quad \text{with } S_1 \sim P(S_1), \qquad \text{(Transition distribution)} \tag{4}$$
$$d_i \sim P_D(d_i \mid S_i), \qquad \text{(Duration distribution)} \tag{5}$$
$$O_t \sim P(O_t \mid S_i), \quad \text{for } t = t_i, \ldots, t_i + d_i - 1, \qquad \text{(Emission distribution)} \tag{6}$$

where each variable is drawn from the specified distribution subject to the constraint $\sum_{i=1}^{N} d_i = T$, which ensures that the total duration matches the full observation sequence of length T.

By allowing state durations to follow arbitrary distributions $P_D(d_i \mid S_i)$ rather than being implicitly geometric (as in HMMs), HSMMs provide greater degrees of freedom in modeling real-world temporal dynamics. For instance, the model can better capture prolonged hidden states such as a user deliberating before a purchase. In our case, we use the discrete Weibull distribution to model these durations. However, this increased expressiveness comes at a cost: model estimation and inference become more computationally intensive, since the model must explicitly consider all possible durations. This can impact scalability, particularly for long sequences or large datasets.

To apply the HSMM framework to our task, we model user sessions in terms of hidden intent states and their durations. Let $O = \{O_1, O_2, \ldots, O_T\}$ denote the observed sequence of user actions during a session of length T, where each O_t is a vector representing the user's behavior at time step t. The corresponding sequence of hidden states $\mathcal{S} = \{S_1, S_2, \ldots, S_N\}$ represents unobserved user intent states (e.g., browsing, comparing, considering, intending to buy, abandoning). The duration for which the system remains in a given hidden state S_i is modeled by the distribution $d_i \sim P_D(d \mid S_i)$. The generative process for this setting follows the formulation described above: state transitions are modeled with $P(S_i \mid S_{i-1})$, durations with

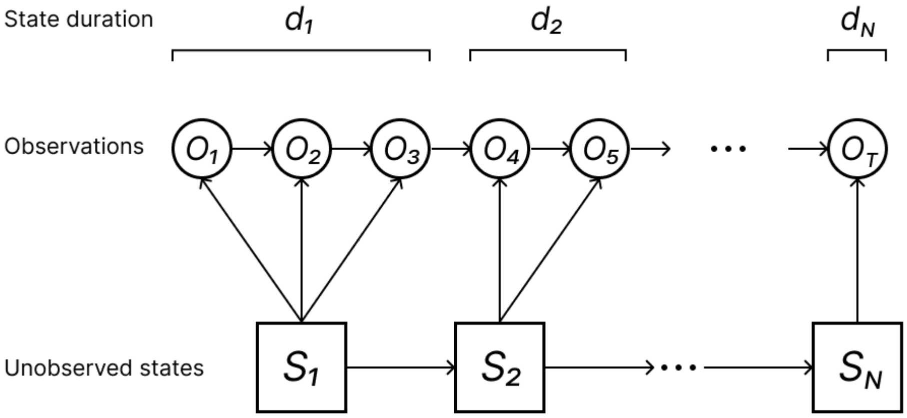


Figure 8: Hidden semi-Markov model process.

$P_D(d_i \mid S_i)$, and emissions with $P(O_t \mid S_i)$ over the duration of each segment.

Bringing these components together, the joint probability of the observed actions, hidden states, and durations is defined as:

$$\begin{aligned} P(O, \mathcal{S}, \{d_i\}) = P(S_1) P_D(d_1 \mid S_1) \prod_{t=1}^{d_1} P(O_t \mid S_1) \\ \cdot \prod_{i=2}^{N} \left[P(S_i \mid S_{i-1}) P_D(d_i \mid S_i) \prod_{t=t_i}^{t_i+d_i-1} P(O_t \mid S_i) \right] \end{aligned} \tag{7}$$

where $t_i = 1 + \sum_{j=1}^{i-1} d_j$ is the starting index of the ith segment. (Yu [89])

The model estimation consists of finding the state transition matrix $P(S_i \mid S_{i-1})$, duration distribution $P_D(d_i \mid S_i)$ and emission distribution $P(O_t \mid S_i)$ that maximize the likelihood of the observed user actions. In the case of multiple user sessions, the objective becomes maximizing the joint likelihood of the entire dataset under the model. The estimation algorithms are usually iterative and require a starting point of iteration, which needs to be chosen carefully.

The estimated models can be used to make classification decisions. The objective is to classify each user session into one of two classes: purchase (C_1) or no purchase (C_2), based on the observed action sequence O. This is done by computing the posterior probability:

$$\hat{C} = \arg \max_{k \in \{1,2\}} P(C_k \mid O), \tag{8}$$

where the posterior is estimated via Bayes' rule:

$$P(C_k \mid O) = \frac{P(O \mid C_k) \cdot P(C_k)}{P(O)}, \tag{9}$$

with $P(O \mid C_k)$ the class-conditional likelihood computed by estimating an HSMM for each class $k \in \{1, 2\}$, and $P(C_k)$ the class prior estimated from the empirical class distribution in the estimation data. The denominator $P(O)$ in Equation (9) is constant across classes and cancels out when maximizing the posterior in Equation (8). By estimating separate HSMMs for each class, the model captures both temporal and sequential dependencies in user behavior, including the varying durations users spend in different hidden intent states. This enables more accurate classification of sessions as leading to a purchase or not.

4.7 Model implementation

In this section, we describe the process of estimating hidden semi-Markov models for purchase and non-purchase sessions using their respective class-labeled datasets. To implement the models, we use Python 3.11.9 and the `hsmmlearn` library (Vankerschaver [84]). We estimate the models with 80% of the data and use the remaining 20% for validating the classification accuracy of the estimated models. The models are used to predict purchase decisions and interpret user behavior. Model performance is assessed on the held-out 20% validation set using several classification metrics: accuracy, balanced accuracy, recall, and area under the ROC curve. These metrics are defined and reported in Section 5.1.

4.7.1 Initializing the models

Table 7: Event category pair integer representations.

Integer token	Event name, category
0	`add_to_cart`, CART
1	`add_to_cart`, CHECKOUT
2	`add_to_cart`, INFO
3	`add_to_cart`, PRODUCT
4	`page_view`, CART
5	`page_view`, CHECKOUT
6	`page_view`, COLLECTION
7	`page_view`, HOME
8	`page_view`, INFO
9	`page_view`, PRODUCT
10	`page_view`, SEARCH
11	`remove_from_cart`, CART
12	`remove_from_cart`, INFO

Before we can fit the HSMMs to our data, we must initialize them. We use the class `MultinomialHSMM` from `hsmmlearn` to initialize HSMMs which requires us to specify the number of states K to form and provide initial guesses for the emission probabilities, transition matrices and duration distributions. We estimate the models using $K = 3$. This choice is justified in Section 5.5.

`MultinomialHSMM` expects the input data to be a sequence of sequences, where each inner sequence represents a user session composed of (event name, category) pairs represented by integers. The pair (`page_view`, Order) and all actions after it are removed from the modeling data. Viewing an Order page indicates a completed purchase so leaving it in would make the classification trivial and the actions after the purchase can not be used for making predictions. The complete list of integer representations of (event name, category) pairs is given in Table 7.

4.7.2 Estimating the models

Algorithm 1: Expectation-maximization algorithm implementation in `hsmmlearn`.

```
Input: Observed sequence of user actions $O$, maximum iterations max_iter,
convergence threshold $\epsilon$
Initialize:
    • Transition probabilities $A$
    • Emission probabilities $B$
    • Duration distributions $P_D$
    • Initial state probabilities $P(S_1)$
    • Previous log-likelihood $\ell_{\text{prev}} \leftarrow -\infty$
for $t = 1$ to max_iter do
    Estimate how likely the observed data $O$ is under the current parameters
    Update:
    • Initial state probabilities $P(S_1)$
    • Transition probabilities $A$
    • Duration distributions $P_D$
    • Emission probabilities $B$
    Compute current log-likelihood $\ell_{\text{curr}}$
    if $|\ell_{\text{curr}} - \ell_{\text{prev}}| < \epsilon$ then
        break
    end if
    $\ell_{\text{prev}} \leftarrow \ell_{\text{curr}}$
end for
Output: Estimated parameters $A$, $B$, $P_D$, and $P(S_1)$
```

The transition matrix A, emission matrix B, and duration distributions are iteratively updated to maximize the joint likelihood of the data under the model using the expectation-maximization (EM) algorithm (see Dempster et al. [19]). The algorithm alternates between two steps: an expectation step (E-step) and a maximization step (M-step). In the E-step, it computes the expected value of the log of the full joint probability in Equation (7), using the current guesses for the model parameters, i.e., the transition probabilities, emission probabilities, and duration distributions. In the M-step, the model parameters are updated to make this expected value as large as possible, by calculating new values based on how often each parameter is estimated to have contributed to generating the observed data. These estimates are calculated using the probabilities of being in each state and of transitioning between states at each point in the sequence, given the current parameter values. In this study, we use the EM algorithm implemented in `hsmmlearn`. The procedure is described in Algorithm 1.

We run the EM procedure for a maximum of 5000 iterations or until convergence below a tolerance threshold of $\epsilon < 10^{-3}$, where ϵ is the absolute difference between the log-likelihood of the observed data in the previous step and the current step. Using a relative difference instead could introduce instability when the log-likelihood values are small or close to zero, since dividing by very small numbers can lead to large values. This means that the algorithm terminates when changes to the model parameters result in a change in the log-likelihood smaller than the specified tolerance threshold ϵ. The stopping criteria were selected based on empirical observations. Since the use of an absolute difference means that an appropriate value for ϵ depends on the scale of the log-likelihood values, we evaluate different values to balance estimation time and model performance on validation data. The maximum iterations is set as an upper limit to prevent infinite loops in cases where the algorithm fails to converge. The convergence diagnostics for different values of ϵ are reported in Section 5.5.

4.7.3 Initial guesses for the parameters

As mentioned above, we need to provide initial guesses for the `MultinomialHSMM` class to initialize the models. The initial values for the parameters affect the convergence of the model so they have to be chosen carefully.

We use k-means clustering for initializing the emission probabilities by grouping similar actions together based on how they appear across all user sessions. The k-means clustering is implemented with the `KMeans` class from the `scikit-learn` library (version 1.6.1) (Pedregosa et al. [70]). Each cluster is then treated as an initial approximation of a hidden state, and the emission probabilities are calculated by counting how often each action appears within each cluster. This gives the model a reasonable starting point for estimating the relationship between hidden intent states and observed behaviors. While k-means is not ideal for categorical data, since it relies on numeric distances that may not fully reflect the structure of discrete events, it offers a simple, fast, and practical method for initializing the model in the absence of labeled state information. More specialized methods could improve this step, but k-means provides a useful balance between simplicity and effectiveness for our current setup.

For the durations, we use the discrete Weibull distribution described in Section 4.5. We fit the Weibull parameters for each class based on observed state durations from a preliminary model. For the purchase model, the best-fitting parameters were shape

$c_p = 1.436$ and scale $\theta_p = 0.328$ and for the non-purchase model $c_{np} = 0.805$ and scale $\theta_{np} = 0.601$. Figure 9 illustrates the resulting Weibull distributions in comparison with a geometric distribution ($p = 0.2$), which corresponds to the implicit duration distribution of a standard HMM with self-transition probabilities of 0.80. Self-transition probabilities indicate the likelihood that the model remains in the same state across consecutive time steps.

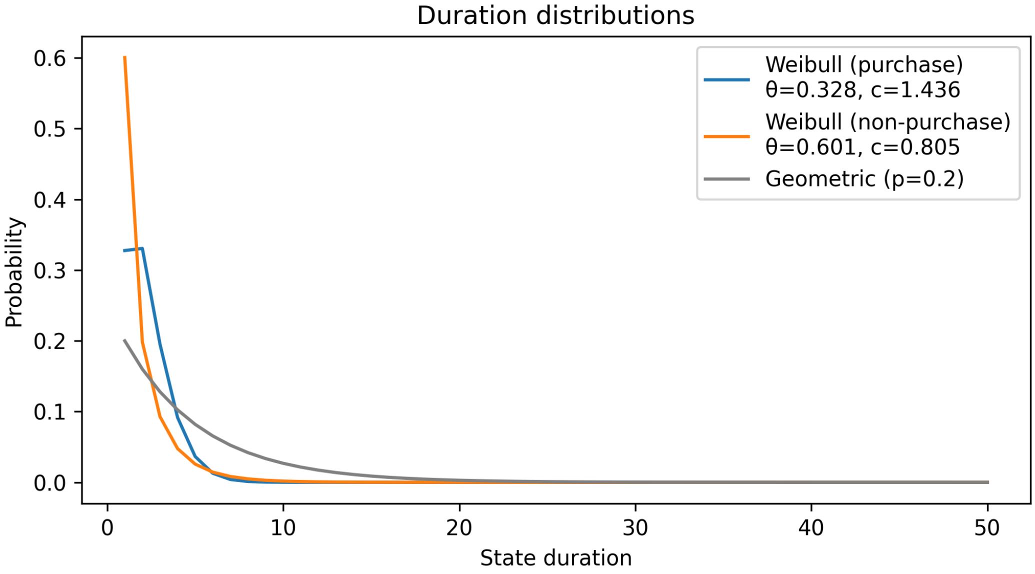


Figure 9: Weibull distributions for the HSMMs and a geometric distribution.

We initialize the transition matrix with the assumption that users tend to stay in the same behavioral state for multiple steps before switching, such as continuing to browse or compare products, rather than frequently alternating between different states. To reflect this, we assign a high probability (0.80) to remaining in the same state and distribute the remaining probability equally across transitions to other states. Each row is then normalized so that the transition probabilities sum to one. This creates a bias towards state persistence while still allowing for transitions.

4.7.4 Session classification and interpretation

After estimating the hidden semi-Markov models for both classes, we apply them to make classification decisions for the remaining 20% of the data evaluating their classification accuracy. Specifically, we compute the log-likelihood of each validation sequence $O = \{O_1, \ldots, O_T\}$ under each class-specific HSMM. We use the log-likelihood to avoid small probabilities being rounded to zero, which can happen when multiplying many such values over long sequences. Computing this log-likelihood involves first evaluating the marginal likelihood $P(O \mid C_k)$, that is, the total probability of observing O under class C_k, as defined by the generative model in Equation (7). Because the hidden state sequence and durations are not observed, this requires summing the joint probability over all possible combinations of hidden states $\mathcal{S}$ and durations d_i that could have produced the observations:

$$P(O \mid C_k) = \sum_{\mathcal{S},\{d_i\}} P(O, \mathcal{S}, \{d_i\} \mid C_k), \tag{10}$$

where $P(O, \mathcal{S}, \{d_i\} \mid C_k)$ is structured as defined in Equation (7), incorporating the initial state distribution $P(S_1)$, the transition probabilities $P(S_{i+1} \mid S_i)$, the state-dependent duration distributions $P_D(d_i \mid S_i)$, and the emission probabilities $P(O_t \mid S_i)$.

In the implementation, the marginal likelihood in Equation (10) is computed by recursively summing, in log-space, over all valid state transitions and duration segments that could produce the observation sequence O, corresponding to the summing over hidden states and durations described in the previous paragraph.

The resulting log-likelihoods $\log P(O \mid C_1)$ and $\log P(O \mid C_2)$ are then combined with empirical class priors using Bayes' rule:

$$\log P(C_k \mid O) = \log P(O \mid C_k) + \log P(C_k), \tag{11}$$

which corresponds to applying a maximum a posteriori (MAP) classifier over the generative likelihoods conditioned on class C_k. The priors $P(C_1)$ and $P(C_2)$ are calculated empirically from the estimating data as the relative frequencies of purchase and non-purchase sessions, respectively. To obtain normalized posteriors for classification, the log-posteriors are exponentiated and normalized:

$$P(C_1 \mid O) = \frac{\exp\left(\log P(C_1 \mid O)\right)}{\exp\left(\log P(C_1 \mid O)\right) + \exp\left(\log P(C_2 \mid O)\right)}, \tag{12}$$

$$P(C_2 \mid O) = \frac{\exp\left(\log P(C_2 \mid O)\right)}{\exp\left(\log P(C_2 \mid O)\right) + \exp\left(\log P(C_1 \mid O)\right)}, \tag{13}$$

which allows for a direct probabilistic interpretation. A final classification is then made by comparing the posterior $P(C_1 \mid O)$ to an empirically chosen classification threshold and assigning the session to class C_1 if the threshold is exceeded.

We select the classification threshold by examining the model's ROC curve. The ROC (receiver operating characteristic) curve displays the available combinations of hit rate and false positive rate for the model that can be achieved with different classification thresholds (Marzban [58]). It reflects how well the model separates the positive and negative classes as the decision threshold varies. The diagonal line corresponds to random classification, and curves above this line represent increasingly better performance (Marzban [58]). A perfect classifier would produce a curve that rises vertically to a hit rate of 1.0 at a false positive rate of 0.0, and then extends horizontally across the top (Marzban [58]). Because the ROC curve is monotonically increasing, selecting a threshold involves identifying the point with the highest true positive rate that still maintains an acceptable false positive rate. In this case, we select the threshold by locating the first region along the curve where the marginal gain in true positive rate begins to slow, that is, where the curve flattens or its slope first drops below 0.5 and remains low.

To gain insight into the hidden structure inferred by the estimated models, we decode the most likely sequence of hidden states for each observed session. This allows us to analyze how inferred user intent evolves over time. Decoding is performed using the Viterbi algorithm (see Viterbi [85]), which identifies the sequence of hidden states that maximizes the joint probability $P(\mathcal{O}, \mathcal{S}, \{d_i\})$ given the observed actions. The algorithm is adapted within the HSMM framework to account for variable-duration state segments. By recovering the most probable state path, we can qualitatively interpret patterns of behavior associated with purchase and non-purchase sessions and evaluate the model's internal representations. The hidden state interpretation is discussed in more detail in Section 5.2.

5 Results

5.1 Classification performance

We evaluate the prediction performance of the model using several common classification metrics (Hatt and Feuerriegel [40], Montgomery et al. [65]). **Accuracy** measures the overall proportion of correctly classified sessions and is defined as

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}, \tag{14}$$

where TP, TN, FP, and FN denote the number of true positives, true negatives, false positives, and false negatives, respectively. To account for class imbalance, we report **balanced accuracy**, calculated as

$$\text{Balanced accuracy} = \frac{1}{2}\left(\frac{TP}{TP + FN} + \frac{TN}{TN + FP}\right), \tag{15}$$

which averages the proportion of correctly classified sessions for each class. We also report the hit rate, defined as

$$\text{Hit rate} = \frac{TP}{TP + FN}, \tag{16}$$

which captures the proportion of purchase sessions correctly identified. **ROC AUC** measures how well the model distinguishes between purchase and non-purchase sessions across all possible classification thresholds. It is computed as the area under the ROC curve, which plots the hit rate (true positive rate) against the false positive rate ($FPR = \frac{FP}{FP+TN}$). A ROC AUC of 1.0 indicates perfect separation between the two classes, while 0.5 corresponds to random guessing. We also present a **confusion matrix**, a 2×2 table that displays the proportion of correct and incorrect predictions for each class. It provides a straightforward visual summary of classification results, helping to identify specific types of misclassification, such as whether the model tends to mislabel purchase sessions as non-purchases or vice versa. Finally, we include the **false positive rate** and the **false negative rate**, corresponding to the proportion of non-purchase sessions misclassified as purchases and the proportion of purchase sessions misclassified as non-purchases, respectively.

Table 8 shows the prediction performance of the HSMMs for a classification threshold of 0.2797 selected as described in Section 4.7.4. The model achieves an overall accuracy of 0.9544 which is largely explained by class imbalance: a naive model that classifies all sessions as non-purchase would still achieve approximately 99% accuracy, since only around 1% of sessions end in a purchase. The balanced accuracy of 0.8897 provides a more meaningful measure in this context, showing that the model performs well across both classes. The ROC AUC of 0.9335 suggests that the model performs well in distinguishing between classes across various classification thresholds. The hit rate of 0.8235 indicates that the model correctly classifies a large proportion of purchase sessions. The confusion matrix in Figure 11 visualizes the

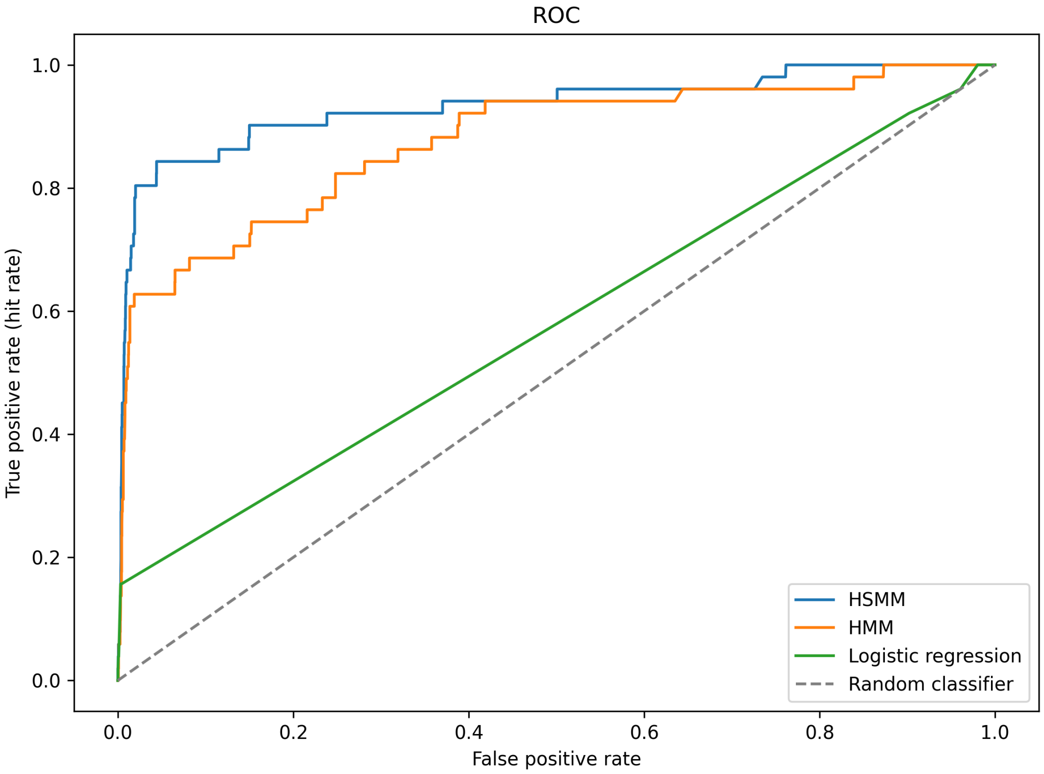


Figure 10: The ROC curves for the selected models.

proportion of correct and incorrect predictions for each class. We see that the model classifies 96% of non-purchase sessions and 82% of purchase sessions correctly.

To further assess the performance of the hidden semi-Markov models, we compare them to two commonly used alternatives: logistic regression and hidden Markov models. Logistic regression, implemented with the `LogisticRegression` class from the `scikit-learn` library (version 1.6.1) (Pedregosa et al. [70]), is often employed as a baseline due to its interpretability and efficiency (Hendriksen et al. [41], Sismeiro and Bucklin [78], Hatt and Feuerriegel [40]). In our case, we use an ℓ_2-regularized logistic regression model with the same input data and default settings except for two modifications: we set the maximum number of iterations to 500, and we enable `class_weight="balanced"` to correct for the substantial class imbalance in the data. ℓ_2-regularization adds a penalty term to the model's loss function proportional to the square of the model coefficients. This discourages the model from assigning overly large weights to any single feature, helping to prevent overfitting and improve generalization to unseen data. The `class_weight="balanced"` option automatically adjusts the weight assigned to each class during estimation based on its frequency in the data. Specifically, the weight for class i is set to $w_i = \frac{n_{\text{samples}}}{n_{\text{classes}} \cdot n_i}$, where n_i is the number of samples in class i. This means the minority class (purchases) is given more influence during estimation, helping the model detect patterns for both classes more effectively and reducing the tendency to predict only the majority class.

Table 8 summarizes classification results across the multiple metrics defined above for all three models. Logistic regression underperforms relative to both sequential models, particularly in identifying purchase sessions. We use the default classification threshold 0.5 for logistic regression. It yields an accuracy of 0.9870, balanced accuracy of 0.5763 and a hit rate of 0.1569 indicating limited effectiveness in capturing session-level behavioral patterns in this setting.

Table 8: Model comparison on classification metrics

Metric	HSMM	logistic regression	HMM
Accuracy	0.9544	**0.9870**	0.9722
Balanced accuracy	**0.8897**	0.5763	0.8016
ROC AUC	**0.9335**	0.5787	0.8761
Hit rate	**0.8235**	0.1569	0.6275
False positive rate	0.0442	**0.0042**	0.0242
False negative rate	**0.1765**	0.8431	0.3725

The HMMs follow the same structure as the HSMMs, with the key distinction that state durations are modeled using geometric distributions. Apart from this, the implementation is kept identical to that of the HSMMs to ensure that any observed differences in performance can be attributed to the choice of duration modeling. We use a classification threshold of 0.5745 for the HMM, again selected as described in Section 4.7.4. Its performance is slightly lower across most metrics, with the exceptions of overall accuracy and the number of false positives. The most notable difference lies in the hit rate, which is 0.6275 for the HMM compared to 0.8235 for the HSMM. As illustrated in the ROC curve in Figure 10, the HMM would require a false positive rate of approximately 30% to achieve a similar hit rate that the HSMM achieves at around 4% false positive rate. This suggests that the explicit duration modeling in the HSMMs more effectively supports the identification of purchase sessions.

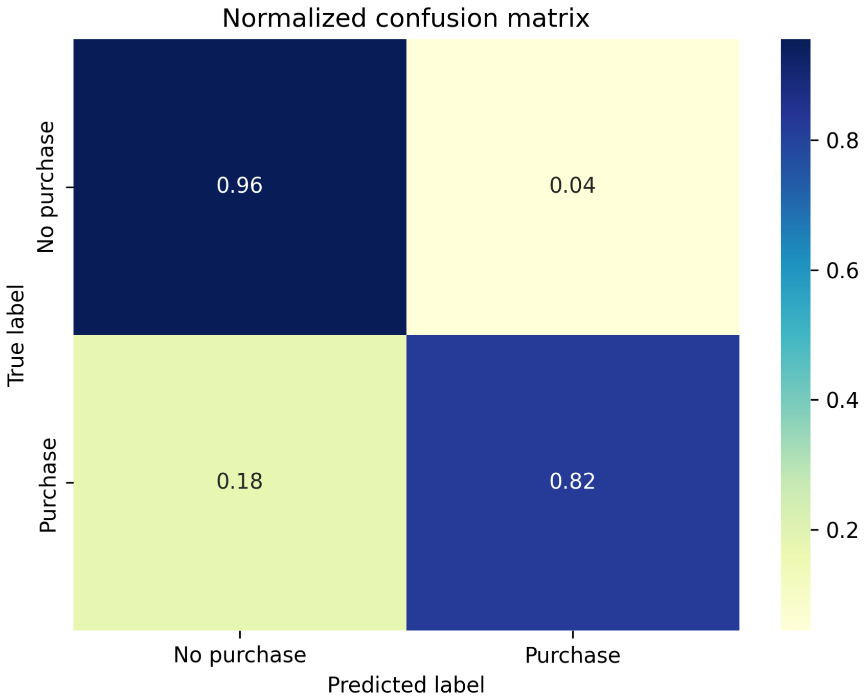


Figure 11: Confusion matrix of the true and predicted labels.

5.2 Hidden state interpretation

To understand the internal structure of the HSMMs, we examine the emission probabilities and inferred state sequences for selected sessions. These analyses allow us to interpret the estimated hidden states in terms of user behavior. The emission distributions for the purchase and non-purchase models are shown in Figures 12 and 13, respectively. Recall that the emission probabilities describe how likely each user action is under a given hidden state. These distributions provide insight into the behavioral patterns associated with each hidden state.

The decision-making process of online consumers can be described as a sequence of stages reflecting the progression of behavior during a shopping session. This includes need recognition, information search, evaluation of alternatives, purchase decision, and post-purchase evaluation (Han [38]). Another perspective describes online consumer decision-making as a process beginning with browsing, followed by information search and evaluation of alternatives, and finally the purchase decision (Awasthi [5]). These views are largely consistent, though the latter omits need recognition and post-purchase evaluation, which typically occur outside the online store and are therefore not reflected in our state interpretations.

The structure of the hidden states inferred by the models align with the stages discussed above. In the purchase model, state 0 has high emission probability for actions such as `page_view:Checkout` and `page_view:Cart`, which suggests that it corresponds to the decision or transaction stage. State 1 is dominated by `page_view:Product` and page_view:Collection actions, indicating a browsing phase that reflects the user's initial exploration of available products after arriving at the store. In contrast, state 2, with high probabilities for add_to_cart:Product and page_view:Cart, appears to reflect a consideration or evaluation stage where the user is comparing options and moving toward a purchase decision.

Similarly, for the non-purchase model, the distinctions between states also reflect progression through decision stages, though the sequence ultimately does not end in a purchase. State 0 is associated with collection page views and may correspond to users passively browsing or not finding relevant products. State 1 shows a higher emission probability for product page views reflecting a more deliberate information search. State 2 has a high emission probability for the action page_view:Info which includes pages such as shipping and payment terms. There are also non-zero emission probabilities for cart related actions and checkout page views which suggests that this state could be explained by consideration behavior.

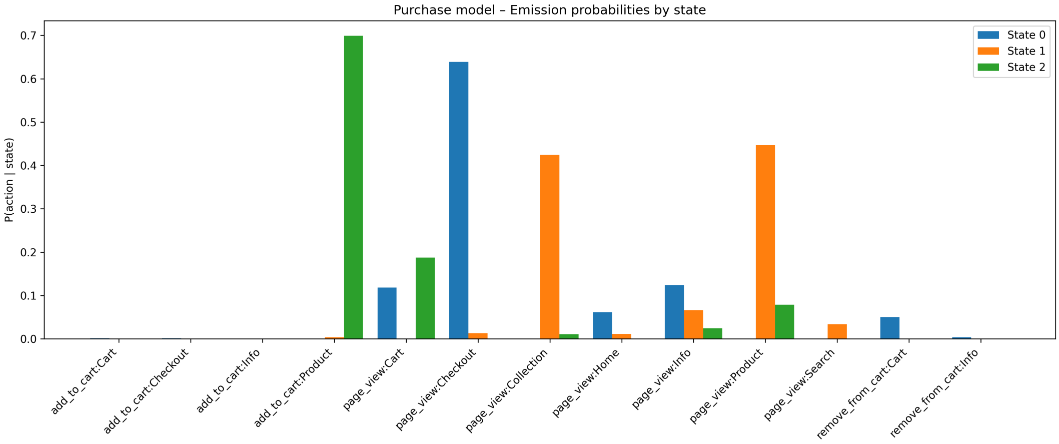


Figure 12: Emission distributions $P(O_t \mid S_i)$ for each hidden state in the purchase HSMM.

To further illustrate the behavior captured by the hidden states, Figure 14 presents two example sessions and their corresponding state sequences as inferred by the model. In the purchase session, the user initially occupies the consideration state (state 2) viewing a product and adding it to the shopping cart. They then transition to state 0 for the remainder of the session, during which they visit the cart, initiate checkout, briefly navigate to the home and information pages, and finally return to checkout to complete the purchase. In contrast, the non-purchase session begins in state 1, where the user browses different product pages. The user then shifts to state 0 for an extended period while exploring collection pages and eventually transitions to state 2, where they add an item to the cart and view additional information pages, without completing the purchase. Finally, the user briefly returns to state 0 before ending the session.

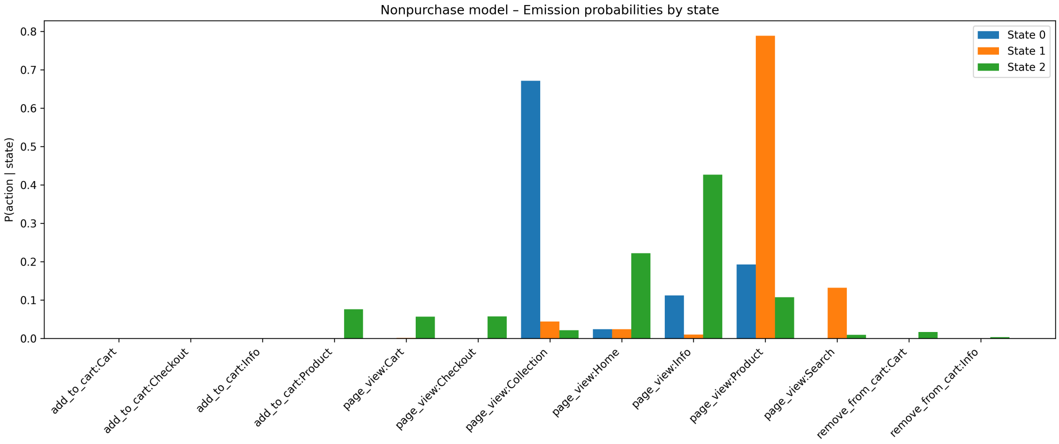
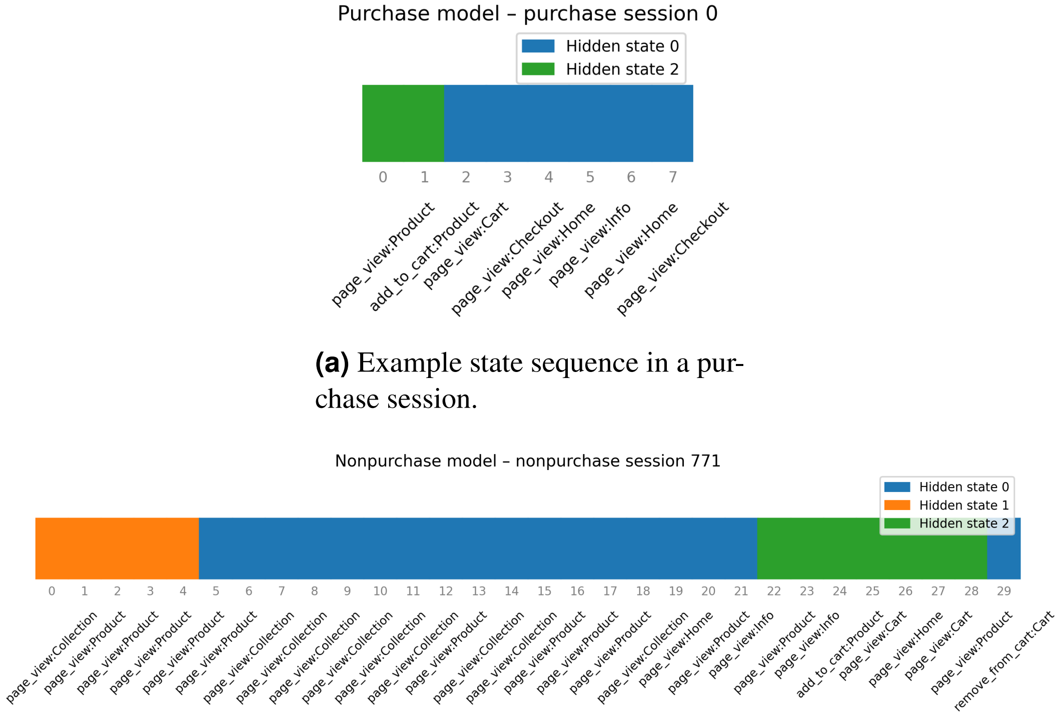


Figure 13: Emission distributions $P(O_t \mid S_i)$ for each hidden state in the non-purchase HSMM.



(a) Example state sequence in a purchase session.

(b) Example state sequence in a non-purchase session.

Figure 14: Hidden state sequences inferred by the HSMMs for example sessions. Each colored block represents one time step, with the assigned hidden state and the corresponding observed event shown below. These examples illustrate how the models segment user behavior into interpretable patterns, conditioned on the session class.

5.3 Duration modeling

To understand how long users tend to remain in a particular hidden state, we examine the distribution of state durations estimated by the HSMMs. Table 9 presents descriptive statistics for the state duration distributions, including the mean, standard deviation, mode, and median, for each state in both the purchase and non-purchase models.

In the purchase model, the mean duration for state 0 (associated with purchase-oriented activity) is 3.43 events, while state 1 (associated with browsing behavior) has a slightly longer mean duration of 4.23 events. State 2, which reflects consideration behavior has a notably brief mean duration of just 1.04 events. The relatively close means and medians for states 0 and 1 suggest that users tend to remain in each state for similar spans of time before switching, indicating comparable persistence in browsing and purchase-oriented behavior. However, the larger standard deviation in state 1 indicates greater variability in browsing patterns, some users browse briefly, while others spend longer periods exploring the store. This variation in how long users stay in the browsing state supports the idea of explicitly modeling state durations, as it allows the model to better account for differences in user behavior that simpler approaches may overlook.

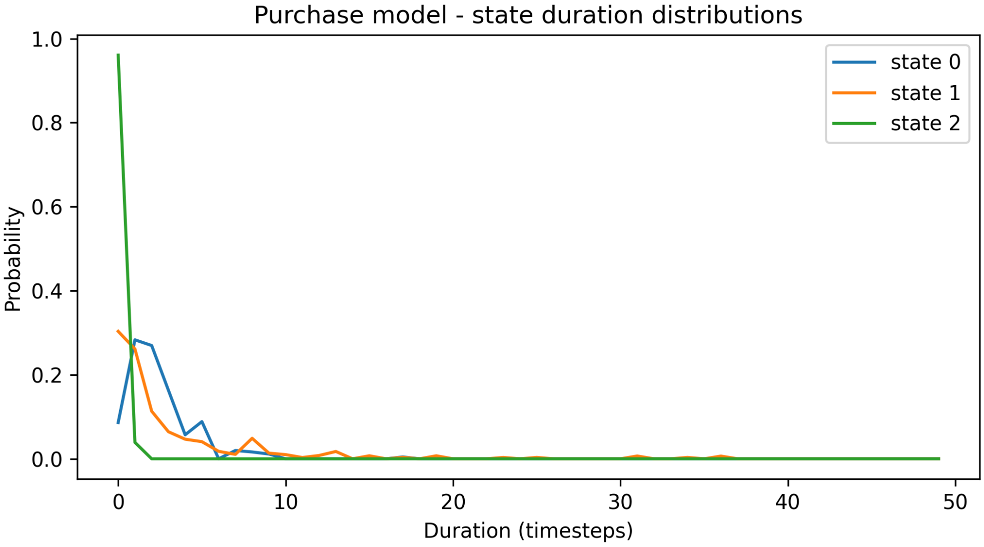
In contrast, the non-purchase model reveals a more pronounced difference in state durations. State 0, which primarily reflects repetitive browsing of collection pages, has a considerably longer mean duration of 7.76 events, with a relatively high standard deviation of 10.13. This suggests that users who do not make purchases tend to remain in this browsing state for extended sequences. State 1 in the non-purchase model has a shorter mean duration of 2.11 events, and state 2 shows the briefest durations, with a mean of 1.90 events.

These patterns indicate that purchase sessions are characterized by more frequent transitions between exploratory and goal-oriented behavior, while non-purchase sessions often involve long, uninterrupted stretches of low-engagement activity. This prolonged browsing may offer an opportunity to proactively influence user behavior, for instance, by presenting a targeted offer or discount to users who appear unlikely to complete a purchase.

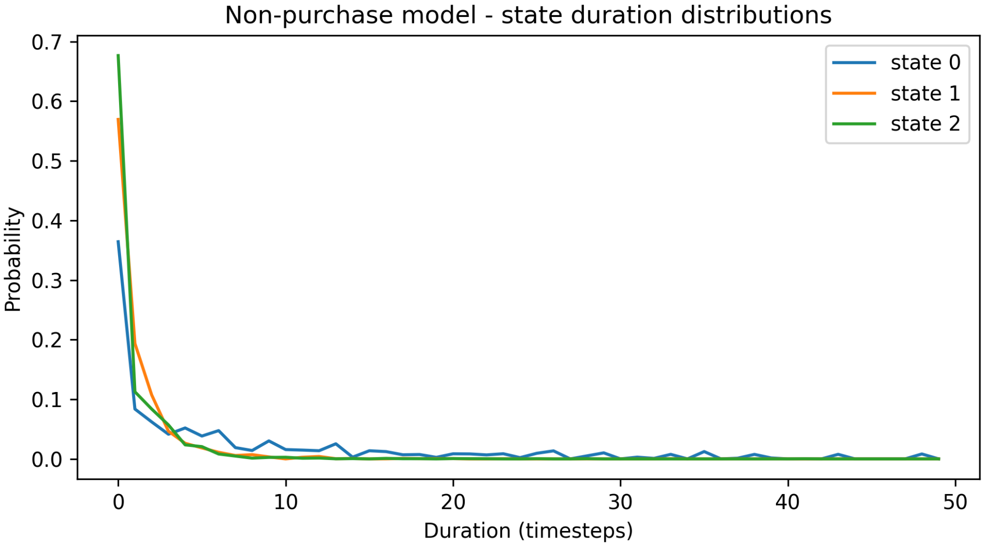
Table 9: Descriptive statistics of state duration distributions (in timesteps) for each hidden state in the HSMMs.

Model	State	Mean	Std	Mode	Median
Purchase	0	3.43	2.07	2	3
	1	4.23	5.58	1	2
	2	1.04	0.19	1	1
Non-purchase	0	7.76	10.13	1	3
	1	2.11	2.24	1	1
	2	1.90	2.06	1	1

Figure 15 visualizes the empirical state duration distributions for each hidden state in the purchase and non-purchase models. These plots offer more detail on the shape of the duration distributions beyond the summary statistics.



(a) State duration distributions for the purchase model.



(b) State duration distributions for the non-purchase model.

Figure 15: Estimated duration distributions $P_D(d \mid S_i)$ for each hidden state in the two HSMMs. The line shows the probability of remaining in a given state for a specified number of time steps. These distributions reflect different behavioral persistence patterns in purchase and non-purchase sessions.

For the purchase model, the duration distributions for all states are relatively concentrated in the short-duration range. State 0, associated with purchase-oriented actions, shows a peak at duration 2, indicating that once users enter this state, they tend to proceed through purchase-related steps relatively quickly. State 1, representing browsing, is also skewed toward shorter durations, peaking at duration 1. State 2, capturing consideration behavior, displays a sharp peak at duration 1, suggesting brief and consistent visits. This suggests that even browsing behavior is fairly dynamic, with users switching between pages or states at a relatively high rate.

The non-purchase model, on the other hand, exhibits a markedly different pattern. State 0 has a flatter and more dispersed distribution, with multiple peaks at the longer durations. The shape of this distribution indicates that users in this state often persist in repetitive, low-engagement activity for extended sequences. State 1 peaks sharply at duration 2 and declines steeply, pointing to brief visits. State 2 also peaks sharply at duration 2, suggesting short consideration behavior that interrupts longer periods in other states.

Taken together, these duration patterns reinforce the behavioral interpretation of the states: purchase behavior involves purposeful, often short-lived transitions between meaningful stages, while non-purchase behavior is more likely to involve long periods of low-variety activity with occasional, brief shifts. The variety in distributional shapes reflects the flexibility of the Weibull distribution to capture temporal dynamics, supporting its use as an initial model choice.

5.4 Real-time purchase prediction

One of the key motivations for using HSMMs in this context is their suitability for real-time prediction. Unlike models that only make decisions after observing the entire sequence, the HSMM should be capable of making intermediate predictions as user sessions unfold. In practice, this means the model can indicate whether a session is likely to end in a purchase before the session has actually concluded, potentially enabling timely interventions such as targeted offers or interface changes.

To evaluate this capability, we simulate a real-time setting by sequentially truncating each session in the validation data and making a prediction at every timestep. This corresponds to the dynamic setting described in Section 3.2.3. Given a session with observed event sequence $O = \{O_1, \ldots, O_T\}$, we perform classification at each partial sequence $\{O_1\}, \{O_1, O_2\}, \ldots, \{O_1, \ldots, O_T\}$. We then identify the earliest point $t \in \{1, \ldots, T\}$ at which the model makes a correct purchase prediction that remains unchanged for the remainder of the session. We refer to this as the point of stable and correct prediction. For comparison, we do the exact same simulation for the estimated HMMs.

Table 10 summarizes the real-time prediction performance of the HSMM and HMM models. The proportions reported in the table refer to session length, measured as the number of events observed so far divided by the total number of events in the session. When comparing the two, we observe that the HSMM achieves correct and stable predictions in a greater proportion of sessions, 82.4% versus 62.7% for the HMM. These values match the overall purchase classification accuracy for full sessions, since the final iteration of the real-time simulation includes the entire session. Therefore, it is more informative to examine at which point in the session a stable and correct prediction is reached. In those sessions where stable predictions are

Table 10: Real-time purchase prediction performance for the HSMM and HMM models.

Statistic	HSMM	HMM
Proportion of correct and stable predictions	0.824	0.627
Mean proportion	0.637	0.804
Median proportion	0.707	0.833
Standard deviation	0.280	0.225
Minimum	0.080	0.120
Maximum	1.000	1.000

achieved, the HSMM tends to reach this point earlier. The mean proportion of the session required for stability is 63.7% for the HSMM, compared to 80.4% for the HMM. The median and minimum values for the models follow the same pattern. The statistics suggest that the HSMM is able to make correct and stable predictions earlier on average.

The histogram in Figure 16 shows the distribution of these proportions for the HSMM and HMM across all purchase sessions. The HSMM achieves stable and correct predictions on a significant number of sessions before observing 50% of them. Further inspection reveals that these early predictions typically correspond to straightforward purchase sessions, where the user visits a product page, adds the item to the cart, and proceeds directly to checkout. The HMM only predicts two purchase sessions correctly before 50% completion which suggests that it is not able to recognize the straightforward purchase behavior. Predictions after 50% are achieved at similar success between the models and for both models there are many sessions that are only predicted just before completion.

The cumulative distribution functions (Figure 17) for the HSMM and HMM provide a complementary view. It shows that for the HSMM, approximately 40% of purchases are correctly predicted around 50% completion. At that point, the HMM has only achieved stable and correct predictions in approximately 10% of purchase sessions. After 70% completion, the cumulative shares for both models start increasing faster likely because of checkout related actions near the end of the sessions.

From a practical standpoint, these results indicate that hidden semi-Markov models are effective at detecting early purchase intent in straightforward sessions, but may struggle with longer sessions involving more exploration or hesitation. This insight can be valuable in several real-world applications. For example, during periods of high traffic, such as Black Friday or Christmas sales, server resources could be dynamically prioritized for users exhibiting clear purchase signals, ensuring they encounter minimal friction when completing their transactions. It also enables targeted discounts or reminders for users who are not exhibiting this type of behavior; if a user is already on track to make a purchase, offering a discount or distracting them with a reminder may be counterproductive.

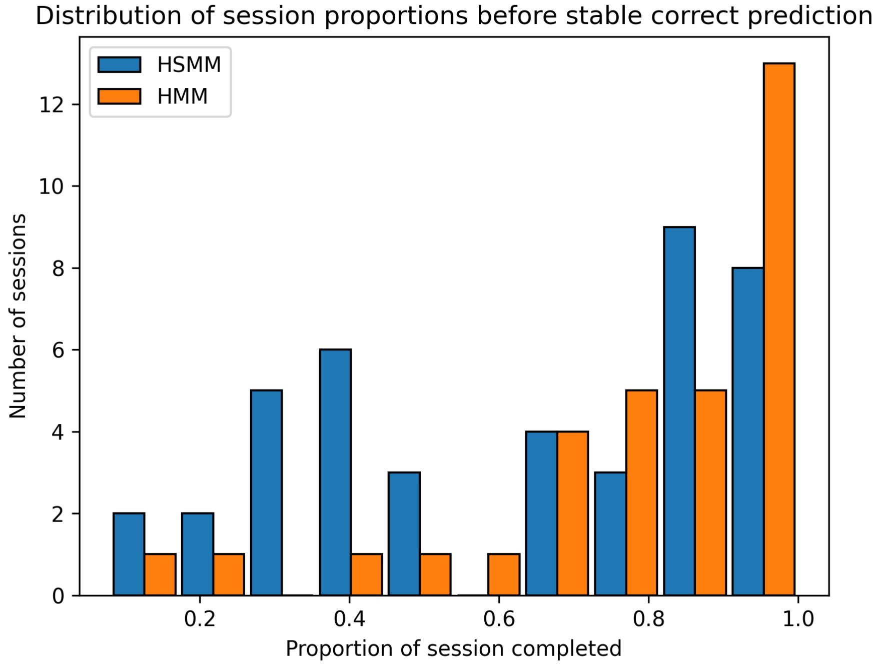


Figure 16: Histogram of session proportions before stable correct prediction.

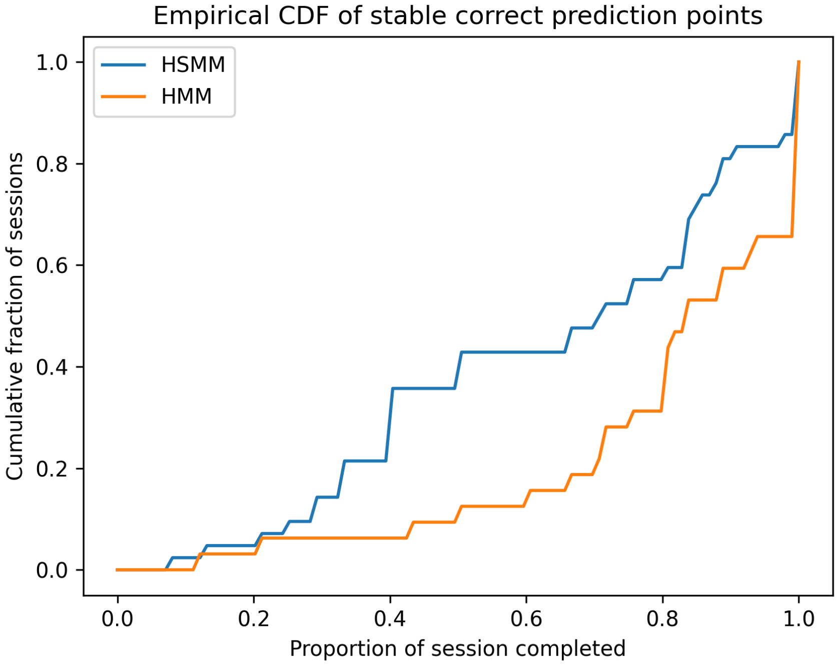


Figure 17: Empirical CDF of session proportions before stable correct prediction.

5.5 Sensitivity analysis

In this section, we investigate how the model performs with different input variables. We examine how the number of states K affects classification performance, evaluate the convergence of the three-state model by running the EM algorithm with different tolerance thresholds ϵ and show how undersampling the set of non-purchase sessions or oversampling the set of purchase sessions affects the results.

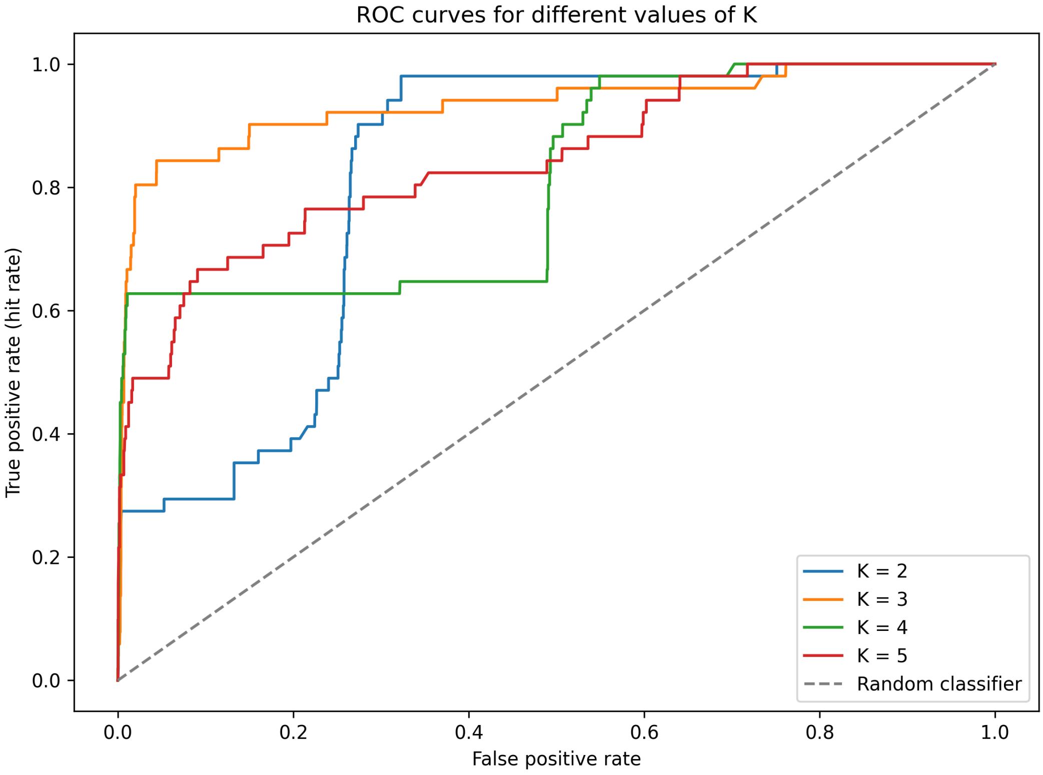


Figure 18: The ROC curves for the models with different values of K.

To examine how the choice of K affects classification performance, we estimate the models with $K \in \{2, 3, 4, 5\}$ and evaluate the performance of each model with the ROC curves. All other input variables are kept identical. The ROC curves presented in Figure 18 indicate which combinations of hit rate and false positive rate are available for each model with different classification thresholds. We can see that for the low false positive rates, the three-state model has a hit rate well above the rest. For a larger false positive rate, around 30%, the two-state model outperforms the three-state model but performs worse than any other model for lower false positive rates. This may reflect limitations in the two-state model's ability to distinguish between purchase and non-purchase behavior when strict classification is required. However, with higher false positives, it compensates by making broader generalizations that increase sensitivity. The four and five-state models perform worse up until very high false positive rates, which may indicate that these models capture patterns specific to the estimation data that do not generalize well to the validation data.

We evaluate the convergence of the three-state model by changing the tolerance ϵ in the EM algorithm used to fit the model parameters to the estimation data, where the tolerance defines the minimum improvement in log-likelihood required for the algorithm to continue iterating. We use absolute tolerance in this case because the log-likelihood values are on a consistent scale across runs. The maximum number of iterations is kept at 5000 to avoid extensively long runtimes. We report the number of iterations the algorithm ran for, the total runtime, and whether it converged in Table 11.

The results show that the purchase model converges quickly across all values of ϵ, with low iteration counts and runtimes. The non-purchase model, on the other hand, requires significantly more iterations and time to converge at every tolerance level. This difference can be explained by the much larger number of sequences used to fit the non-purchase model, which increases the computational load of each EM step. However, while the gains in fit precision from lowering ϵ diminish after 10^{-3}, the runtime continues to grow, particularly for the non-purchase model, which takes nearly 28 minutes at $\epsilon = 10^{-5}$. These results suggest that $\epsilon = 10^{-3}$ remains a reasonable choice for balancing convergence accuracy and computational efficiency, especially when runtime is a concern.

Table 11: Convergence diagnostics for different values of ϵ for the purchase and non-purchase models.

Model	ϵ	Iterations	Time (s)	Converged?
Purchase	10^{-1}	77	0.3	Yes
	10^{-2}	115	0.5	Yes
	10^{-3}	186	0.7	Yes
	10^{-4}	273	1.1	Yes
	10^{-5}	360	1.4	Yes
Non-purchase	10^{-1}	213	91.5	Yes
	10^{-2}	2463	1107.3	Yes
	10^{-3}	2814	1260.3	Yes
	10^{-4}	2821	1309.2	Yes
	10^{-5}	3713	1677.7	Yes

We undersample the non-purchase sessions by randomly selecting the same number of sessions as in the purchase set, without replacement. We evaluate the resulting model by plotting its ROC curve which is presented in Figure 19. The curve shows that the model struggles to separate the classes at low false positive rates, suggesting limited confidence in its predictions when strict classification is required. As the threshold is relaxed, the hit rate increases rapidly, indicating that the model can still detect purchases effectively when a higher false positive rate is acceptable.

The oversampling is done by randomly duplicating purchase sessions with replacement until their number matches the number of non-purchase sessions in the estimation data. While this approach introduces some redundancy, it is a simple way to balance the estimation data without altering the original session structure.

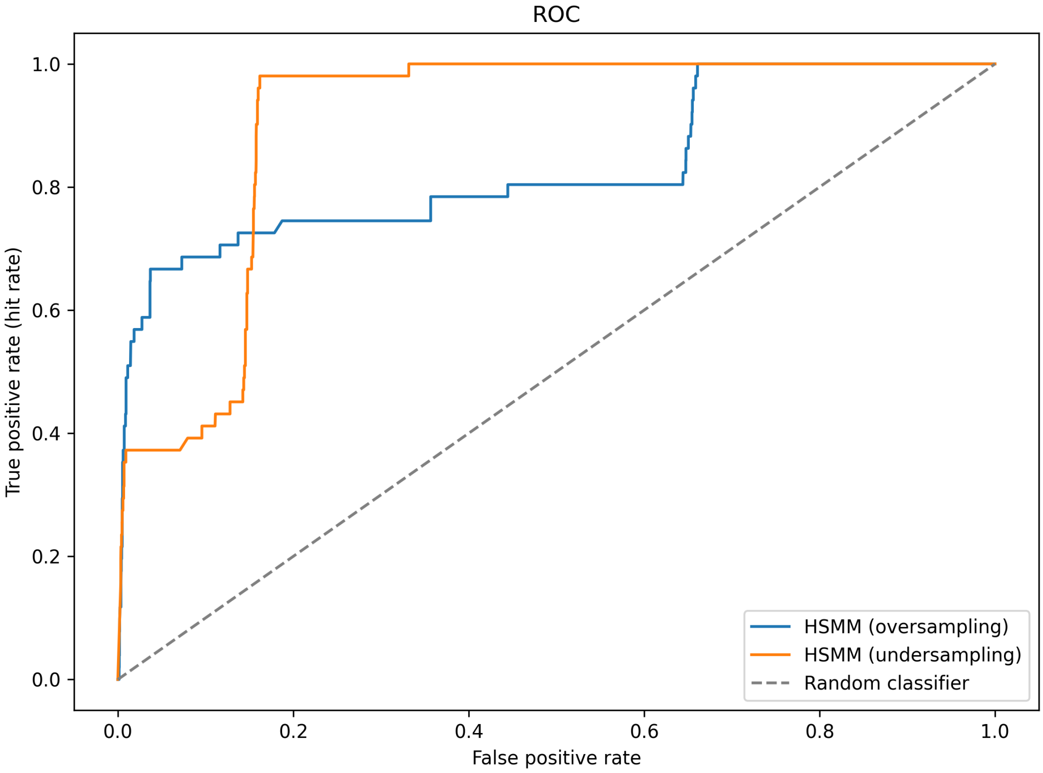


Figure 19: The ROC curves for the HSMMs with oversampled purchase sessions and undersampled non-purchase sessions.

An alternative would be to generate new artificial purchase sessions, but this would require significantly more effort and may result in sequences that do not reflect real user behavior. The ROC curve for the resulting model, shown in Figure 19, suggests that oversampling yields better performance at low false positive rates compared to undersampling, but still falls significantly short of the performance achieved with the original, unbalanced estimation data. This may be due to the limited diversity introduced by duplicating sessions, which can cause the model to be overly reliant repeated patterns and reduce its ability to generalize.

6 Discussion

The motivation for using HSMMs in this study was the assumption that explicit modeling of state durations would capture important behavioral patterns that HMMs might miss. The idea was that session behavior is not only defined by the sequence of actions, but also by how long users remain in particular modes of engagement, for instance, prolonged browsing versus a straightforward checkout path. The results support this hypothesis in this dataset. We managed to achieve solid classification performance with the HSMMs on both full sessions and truncated sessions simulating a real-time scenario, consistently outperforming the HMMs. While many of the sessions in the real-time setting were predicted correctly only towards the end of the session, the model was able to identify early purchase signals in a substantial number of cases.

A key limitation of this study is that the models were estimated and evaluated using data from a single online clothing store. While this focused setting allowed for a detailed analysis of user behavior within a consistent environment, it also raises questions about generalizability. Online stores can differ significantly in their structure, product types, navigation paths and traffic sources. Additionally, customer demographics, browsing habits, and purchase behavior may vary widely across industries and target audiences. As a result, the predictive performance and behavioral patterns identified in this study may not fully translate to other online stores. Future work could address this limitation by applying the models to data from a broader set of stores or domains, potentially uncovering commonalities or domain-specific adaptations that influence model effectiveness.

Another factor that may influence the interpretability and predictive capacity of the models is the abstraction level used in representing user actions. Before modeling, we opted to group similar page types under broader labels in order to reduce the dimensionality, for example, consolidating all product page views under a single "product" category. While this simplification helped reduce sparsity and manage model complexity, it also meant that potentially meaningful distinctions between different products, categories, or content types were not captured. As a result, the models were not able to account for behavioral differences tied to specific product interest, pricing levels, or browsing intent at a more granular level. Including more detailed event representations could potentially reveal richer patterns, such as preferences for certain product types or engagement with promotional content, that are currently collapsed under generalized labels. Future work could explore the trade-off between model tractability and behavioral resolution by systematically evaluating the effects of different levels of event granularity.

The data used in this study was collected over approximately four months, which limited the overall volume of available sessions, particularly purchase sessions, of which there were 252. While this was sufficient for conducting the main experiments, the relatively small sample size may restrict the robustness of the findings, especially for more complex models like the HSMM. Limited data can increase the risk of overfitting, hinder generalization, and reduce the reliability of estimated parameters, such as emission and duration distributions. Furthermore, the restricted time span may not capture seasonal trends, promotional effects, or other temporal dynamics that influence online shopping behavior. Extending the data collection period or incorporating data from multiple time windows could help provide a more comprehensive view of user behavior and enhance the stability and generalizability of the models.

An additional consideration in interpreting the results is the potential bias in the data collection process introduced by cookie consent requirements. The data used in this study was collected exclusively from users who accepted tracking cookies, which means that sessions from users who declined consent were not captured or analyzed. While there is no clear evidence suggesting that cookie rejection is systematically associated with different shopping behavior, it remains a possibility. Users who decline cookies might differ in terms of privacy preferences, engagement levels, or device usage patterns, all of which could subtly affect browsing or purchase behavior. This introduces a potential selection bias that is difficult to quantify but should be acknowledged. It is also worth noting that cookie consent is not required for this type of behavioral tracking in all jurisdictions. Studies conducted in regions with less restrictive consent frameworks could help validate whether models estimated on data from users who agreed to tracking generalize to the broader user population.

Although the models developed in this study show potential for predicting user purchase behavior with high accuracy, their real-world applicability poses practical challenges. Implementing a hidden semi-Markov model in a live e-commerce environment would likely require substantial engineering effort, real-time data infrastructure, and computational resources. In particular, the ability to make accurate, session-level predictions as user activity unfolds requires low-latency event processing and continuous model evaluation. These technical requirements may translate into high implementation costs, both in terms of initial setup and ongoing maintenance. As such, the question arises whether the potential benefits, such as improved targeting, personalization, or intervention strategies, are sufficient to justify these investments. While the value of real-time purchase prediction could be significant in high-traffic or high-margin settings, a cost-benefit analysis should be conducted before deploying such models in practice.

7 Conclusions

In recent years, an increasing share of global retail has shifted to online channels, creating a highly competitive landscape where converting visitors into buyers is crucial for business success. Despite significant investment in website optimization, recommendation systems, and targeted marketing, conversion rates remain modest. As a result, accurately predicting purchase decisions based on user behavior has become a key research focus in e-commerce.

Previous work in this area has explored both machine learning and Markovian modeling approaches. While machine learning models can achieve high predictive accuracy, they typically rely on carefully prepared input data or large estimation datasets. Markov models, in contrast, offer an interpretable probabilistic framework that naturally incorporates sequential behavior. Among these, hidden Markov models (HMMs) have proven effective in capturing hidden user intent. However, a key limitation of HMMs is their assumption of memoryless state transitions, which implicitly imposes a geometric duration distribution. Hidden semi-Markov models (HSMMs) address this limitation by explicitly modeling how long the system remains in each hidden state, offering a potentially richer temporal representation of user behavior.

This thesis investigated the use of HSMMs to classify user sessions based on whether they would end in a purchase, using behavioral data from a Finnish online clothing store. While prior work has demonstrated the utility of HSMMs in modeling user exits, our goal was to evaluate their effectiveness in predicting purchases and assess their performance under real-time constraints, where the full session may not be available. To assess the predictive capability, the model was tested on both complete sessions and truncated sessions that simulate real-time use.

The results suggest that HSMMs are effective for classifying user sessions in our dataset based on purchase intent, particularly due to their ability to model how long users remain in different behavioral states. The approach performed well on both full sessions and truncated sessions simulating real-time conditions. In many cases, the model was able to detect purchase intent early in the session, indicating potential for real-time applications where timely predictions are valuable.

Future research could explore applying these models across a broader set of online retail environments to assess their generalizability and identify domain-specific patterns. Increasing the granularity of user action representations may also uncover richer behavioral signals that are currently abstracted away. In addition, collecting larger and more temporally diverse datasets could improve the robustness of the models and allow for the exploration of seasonal or campaign-driven shopping behaviors. These directions may further clarify the strengths and limitations of sequence-based models in predicting online purchase decisions.

References

[1] Albert, J. H., and Chib, S., "Bayesian Analysis of Binary and Polychotomous Response Data." *Journal of the American Statistical Association*, 1993, vol. 88, no. 422, pp. 669-–679. DOI: 10.1080/01621459.1993.10476321.

[2] Analytics Market, *Google Analytics vs Log Files: A Comprehensive Comparison*. [Online]. https://www.analyticsmarket.com/blog/google-analytics-vs-log-files/ (accessed 28 February 2025).

[3] Ang, S. L., Ong, H. C., and Low, H. C., "Classification using the general Bayesian network." *Pertanika J. Sci. Technol*, 2016, vol. 24, no. 1, pp. 205–211.

[4] Arkes, H. R., and Blumer, C., "The psychology of sunk cost." *Organizational Behavior and Human Decision Processes*, 1985, vol. 35, no. 1, pp. 124–140. DOI: 10.1016/0749-5978(85)90049-4.

[5] Awasthi, K. S., "Consumer behavior analysis in the e-commerce." *In Proceedings of International Conference*, 2022, pp. 29–47.

[6] Baye, M. R., De los Santos, B., and Wildenbeest, M. R., "Search engine optimization: what drives organic traffic to retail sites?." *Journal of Economics & Management Strategy*, 2016, vol. 25, no. 1, pp. 6–31. DOI: 10.1111/jems.12141.

[7] Bekavac, I., and Garbin Praničević, D., "Web analytics tools and web metrics tools: An overview and comparative analysis." *Croatian Operational Research Review*, 2015, vol. 6, no. 2, pp. 373–386. DOI: 10.17535/crorr.2015.0029.

[8] Beyer, K., Goldstein, J., Ramakrishnan, R., and Shaft, U., "When is “nearest neighbor” meaningful?." *Database Theory—ICDT’99*, 1999, pp. 217–235. DOI: 10.1007/3-540-49257-7_15.

[9] Bigon, L., Cassani, G., Greco, C., Lacasa, L., Pavoni, M., Polonioli, A., and Tagliabue, J., "Prediction is very hard, especially about conversion. Predicting user purchases from clickstream data in fashion e-commerce." *arXiv preprint arXiv:1907.00400*, 2019. DOI: 10.48550/arXiv.1907.00400.

[10] Bishop, C. M., "Pattern recognition and machine learning." *New York: springer*, 2006, vol. 4, no. 4.

[11] Bradley, P. S., Fayyad, U. M., and Reina, C., "Scaling clustering algorithms to large databases." *Knowledge Discovery and Data Mining*, 1998, pp. 9–15. DOI: 10.5555/3000292.3000295.

[12] Bucklin, R. E., Lattin, J. M., Ansari, A., Bell, D., Coupey, E., Gupta, S., Little, J. D. C., Mela, C., Montgomery, A. and Steckel, J., "Choice and the Internet: From clickstream to research stream." *Marketing letters*, 2002, vol. 13, pp. 245–258. DOI: 10.1023/A:1020231107662.

[13] Chang, H. J., Hung, L. P., and Ho, C. L., "An anticipation model of potential customers' purchasing behavior based on clustering analysis and association rules analysis." *Expert systems with applications*, 2007, vol. 32, no. 3, pp. 753–764. DOI: 10.1016/j.eswa.2006.01.049.

[14] Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y., "Learning phrase representations using RNN encoder-decoder for statistical machine translation." *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1724–1734.

[15] Cirqueira, D., Hofer, M., Nedbal, D., Helfert, M., and Bezbradica, M., "Customer purchase behavior prediction in e-commerce: A conceptual framework and research agenda." *New Frontiers in Mining Complex Patterns*, Springer, Cham, 2019, vol. 11948, pp. 119–136. DOI: 10.1007/978-3-030-48861-1_8.

[16] Coppola, D., *E-commerce as percentage of total retail sales worldwide from 2021 to 2027*. [Online]. https://www.statista.com/statistics/534123/e-commerce-share-of-retail-sales-worldwide/ (accessed 31 January 2025).

[17] Cover, T., and Hart, P., "Nearest neighbor pattern classification." *IEEE transactions on information theory*, 1967, vol. 13, no. 1, pp. 21–27. DOI: 10.1109/TIT.1967.1053964.

[18] Darwiche, A., "Modeling and Reasoning with Bayesian Networks." *Cambridge University Press*, 2009. DOI: 10.5555/1534901.

[19] Dempster, A. P., Laird, N. M., and Rubin, D. B., "Maximum likelihood from incomplete data via the EM algorithm." *Journal of the Royal Statistical Society: Series B*, 1977, vol. 39, no. 1, pp. 1–22. DOI: 10.1111/j.2517-6161.1977.tb01600.x.

[20] Ding, A. W., Li, S., and Chatterjee, P., "Learning user real-time intent for optimal dynamic web page transformation." *Information Systems Research*, 2015, vol. 26, no. 2, pp. 339–359. DOI: 10.1287/isre.2015.0568.

[21] Dushnitsky, G. and Stroube, B., "Low-code entrepreneurship: Shopify and the alternative path to growth." *Journal of Business Venturing Insights*, 2021, vol. 16. DOI: 10.1016/j.jbvi.2021.e00251.

[22] Esmeli, R., and Gokce, A., "An Analysis of Consumer Purchase Behavior Following Cart Addition in E-Commerce Utilizing Explainable Artificial Intelligence." *Journal of Theoretical and Applied Electronic Commerce Research*, 2025, vol. 20, no. 1, pp. 28. DOI: 10.3390/jtaer20010028.

[23] European Union, *General Data Protection Regulation (GDPR)*. [Online]. https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A32016R0679 (accessed 17 February 2025).

[24] Ferrini, A. and Mohr, J., "Uses, limitations, and trends in web analytics." *Handbook of research on Web log analysis*, IGI Global, 2009, pp. 124–142. DOI: 10.4018/978-1-59904-974-8.ch007.

[25] Fedorovicius, J., and Krajcir, K., *Install Google Tag Manager and GA4 on Shopify (with Custom Pixel)*. [Online]. https://www.analyticsmania.com/post/install-google-tag-manager-and-ga4-on-shopify/ (accessed 20 January 2025).

[26] Filip, P., and Čegan, L., "Comparing Tools for Web-session Recording and Replaying." *International Conference on Sustainable Information Engineering and Technology (SIET)*, IEEE, 2019, pp. 257–260. DOI: 10.1109/SIET48054.2019.8986134.

[27] Fisher, R. A., "The Use of Multiple Measurements in Taxonomic Problems." *Annals of Eugenics*, 1936, vol. 7, pp. 179–188. DOI: 10.1111/j.1469-1809.1936.tb02137.x.

[28] Garimella, K., Kostakis, O., and Mathioudakis, M., "Ad-blocking: A study on performance, privacy and counter-measures." *Proceedings of the 2017 ACM on Web Science Conference*, 2017, pp. 259–262. DOI: 10.48550/arXiv.1705.03193.

[29] George, D. A. S., and George, A. H., "The evolution of content delivery network: How it enhances video services, streaming, games, ecommerce, and advertising." *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering (IJAREEIE)*, 2021, vol. 10, no. 7, pp. 10435–10442. DOI: 10.15662/IJAREEIE.2021.1007040.

[30] Ghahramani, Z., "An introduction to hidden Markov models and Bayesian networks." *International journal of pattern recognition and artificial intelligence*, 2001, vol. 15, no. 1, pp. 9–42. DOI: 10.5555/505741.505743.

[31] Goodfellow, I., Bengio, Y. and Courville, A., "Deep Learning." *MIT Press*, 2016. DOI: 10.5555/3086952.

[32] Google, *Google Analytics*. [Online]. https://developers.google.com/analytics.

[33] Google, *Google Tag Manager*. [Online]. https://developers.google.com/tag-platform/tag-manager.

[34] Google, *BigQuery*. [Online]. https://cloud.google.com/bigquery/docs.

[35] Google, *Use nested and repeated fields* https://cloud.google.com/bigquery/docs/best-practices-performance-nested

[36] Gudigantala, N., Bicen, P. and Eom, M., "An examination of antecedents of conversion rates of e-commerce retailers." *Management Research Review*, 2016, vol. 39, no. 1, pp. 82–114. DOI: 10.1108/MRR-05-2014-0112.

[37] Habel, J., Alavi, S. and Heinitz, N., "A theory of predictive sales analytics adoption." *AMS Rev 13*, 2023, pp. 34-–54. DOI: 10.1007/s13162-022-00252-0.

[38] Han, W., "Purchasing decision-making process of online consumers." *In 2021 international conference on public relations and social sciences*, 2021, pp. 545–548. DOI: 10.2991/assehr.k.211020.214.

[39] Hastie, T., Tibshirani, R., Friedman, J. H. and Friedman, J. H., "The elements of statistical learning: data mining, inference, and prediction." *Springer*, 2009, vol. 2, pp. 1–758. DOI: 10.1007/978-0-387-84858-7.

[40] Hatt, T. and Feuerriegel, S. "Detecting user exits from online behavior: A duration-dependent latent state model." *arXiv preprint arXiv:2208.03937*, 2022. DOI: 10.48550/arXiv.2208.03937.

[41] Hendriksen, M., Kuiper, E., Nauts, P., Schelter, S. and de Rijke, M., "Analyzing and predicting purchase intent in e-commerce: anonymous vs. identified customers." *arXiv preprint arXiv:2012.08777*, 2020, DOI: 10.48550/arXiv.2012.08777.

[42] Hochreiter, S., and Schmidhuber, J., "Long short-term memory." *Neural computation*, 1997, vol. 9, no. 8, pp. 1735–1780. DOI: 10.1162/neco.1997.9.8.1735.

[43] Hornik, K., Stinchcombe, M. and White, H., "Multilayer feedforward networks are universal approximators." *Neural Networks*, 1989, vol. 2, no. 5, pp. 359–366. DOI: 10.1016/0893-6080(89)90020-8.

[44] Hosmer, D. W., Lemeshow, S., Sturdivant, R. X., "Applied Logistic Regression." *Wiley*, 2013. DOI: 10.1002/9781118548387.

[45] Huang, T., and Van Mieghem, J. A., "Clickstream data and inventory management: Model and empirical analysis." *Production and Operations Management*, 2014, vol. 23, no. 3, pp. 333–347. DOI: 10.1111/poms.12046.

[46] Jain, A. K., Murty, M. N., and Flynn, P. J., "Data clustering: a review." *ACM Computing Surveys*, 1999, vol. 31, no. 3, pp. 264–323. DOI: 10.1145/331499.331504.

[47] Jain, A. K., "Data clustering: 50 years beyond K-means." *Pattern Recognition Letters*, 2010, vol. 31, no. 8, pp. 651–666. DOI: 10.1016/j.patrec.2009.09.011.

[48] Khalid, S. and Shukairy, A., "Conversion optimization: The art and science of converting prospects to customers." *O'Reilly Media, Inc.*, 2010.

[49] King, R., Churchill, E.F. and Tan, C., "Designing with data: Improving the user experience with A/B testing." *O'Reilly Media, Inc.*, 2017.

[50] Koller, D. and Friedman, N., "Probabilistic Graphical Models: Principles and Techniques." *MIT Press*, 2009. DOI: 10.5555/1795555.

[51] Kotsiantis, S. B., Kanellopoulos, D., and Pintelas, P. E., "Data preprocessing for supervised leaning." *International journal of computer science*, 2006, vol. 1, no. 2, pp. 111–117. DOI: 10.5281/zenodo.1082415.

[52] LeCun, Y., Bengio, Y. and Hinton, G., "Deep Learning." *Nature*, 2015, vol. 521, no. 7553, pp. 436–444. DOI: 10.1038/nature14539.

[53] Lee, H. G., Lee, S. C., Kim, H. Y., and Lee, R. H., "Is the internet making retail transactions more efficient?: Comparison of online and offline CD retail markets." *Electronic Commerce Research and Applications*, 2003, vol. 2, no. 3, pp. 266-–277. ISSN 1567-4223. DOI: 10.1016/S1567-4223(03)00030-9.

[54] Liu, C., "Robit regression: a simple robust alternative to logistic and probit regression." *Applied Bayesian Modeling and Casual Inference from Incomplete-Data Perspectives*, 2004, pp. 227–238. DOI: 10.1002/0470090456.ch21.

[55] MacKay, D. J. C., "A practical Bayesian framework for backpropagation networks." *Neural Computation*, 1992, vol. 4, no. 3, pp. 448–472. DOI: 10.1162/neco.1992.4.3.448.

[56] Madhuri, A., Shireesha, M., Reddy, S. M., and Kumar, B. R., "Exploring the Role of Personalization in E-commerce: Impacts on Consumer Trust and Purchase Intentions." *European Economics Letters*, 2024, vol. 14, no. 3, pp. 907–919. DOI: 10.52783/eel.v14i3.1845.

[57] Markov, A. A., "Rasprostranenie zakona bol'shih chisel na velichiny, zavisyaschie drug ot druga." *Izvestiya Fiziko-matematicheskogo obschestva pri Kazanskom universitete*, 1906, vol. 15, pp. 135—156.

[58] Marzban, C., "The ROC curve and the area under it as performance measures." *Weather and Forecasting*, 2004, vol. 19, no. 6, pp. 1106–1114. DOI: 10.1175/825.1.

[59] McKinney, W., "Data Structures for Statistical Computing in Python." *Proceedings of the 9th Python in Science Conference*, 2010, pp. 51–56.

[60] McLachlan, G. and Peel, D., "Finite Mixture Models." *Wiley*, 2000. DOI: 10.1002/0471721182.

[61] Menon, A., Narasimhan, H., Agarwal, S., and Chawla, S., "On the statistical consistency of algorithms for binary classification under class imbalance." *In International Conference on Machine Learning*, 2013, vol. 28, no. 3, pp. 603–611. DOI: 10.5555/3042817.3043004.

[62] Moe, W. and Fader, P., "Dynamic conversion behavior at e-commerce sites." *Management Science*, 2004, vol. 50, no. 3, pp. 326–335. DOI: 10.1287/mnsc.1040.0153.

[63] Mokryn, O., Bogina, V., and Kuflik, T., "Will this session end with a purchase? Inferring current purchase intent of anonymous visitors." *Electronic Commerce Research and Applications*, 2019, vol. 34. DOI: 10.1016/j.elerap.2019.100836.

[64] Montgomery, A. L., "Applying Quantitative Marketing Techniques to the Internet." *Interfaces*, 2001, vol. 31, no. 2, pp. 90–108. DOI: 10.1287/inte.31.2.90.10630.

[65] Montgomery, A. L., Li, S., Srinivasan, K., and Liechty, J. C., "Modeling online browsing and path analysis using clickstream data." *Marketing science*, 2004, vol. 23, no. 4, pp. 579–595. DOI: 10.1287/mksc.1040.0073.

[66] Murphy, K. P., "Machine Learning: A Probabilistic Perspective." *MIT Press*, 2012.

[67] Nakagawa, T., and Osaki, S., "The Discrete Weibull Distribution." *IEEE Transactions on Reliability*, 1975, vol. 24, no. 5, pp. 300-–301. DOI: 10.1109/TR.1975.5214915.

[68] Nishimura, N., Sukegawa, N., Takano, Y., and Iwanaga, J., "A latent-class model for estimating product-choice probabilities from clickstream data." *Information Sciences*, 2018, vol. 429, pp. 406–420. DOI: 10.48550/arXiv.1612.06589.

[69] Pearl, J., "Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference." *Morgan Kaufmann*, 1988. DOI: 10.5555/534975.

[70] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M. and Duchesnay, E., "Scikit-learn: Machine Learning in Python." *Journal of Machine Learning Research*, 2011. vol. 12, pp. 2825–2830.

[71] Prasath, V. B., Alfeilat, H. A. A., Lasassmeh, O., and Hassanat, A. B. A., "Distance and Similarity Measures Effect on the Performance of K-Nearest Neighbor Classifier–A." *arXiv preprint arXiv:1708.04321*, 2017. DOI: 10.1089/BIG.2018.0175.

[72] Pushkar, O., Hrabovskyi, Y., and Gordyeyev, A., "Development of a method for optimizing the site loading speed." *Eastern-European Journal of Enterprise Technologies*, 2020, vol. 2, no. 2, pp. 21–29. DOI: 10.15587/1729-4061.2020.216993.

[73] Rabiner, L. R., "A tutorial on hidden Markov models and selected applications in speech recognition." *Proceedings of the IEEE*, 1989, vol. 77, no. 2, pp. 257–286. DOI: 10.1109/5.18626.

[74] Raina, R., Shen, Y., Mccallum, A., and Ng, A., "Classification with hybrid generative/discriminative models." *Advances in neural information processing systems*, 2003, vol. 16. DOI: 10.5555/2981345.2981414.

[75] Rumelhart, D. E., Hinton, G. E. and Williams, R. J., "Learning representations by back-propagating errors" *Nature*, 1986, vol. 323, no. 6088, pp. 533–536. DOI: 10.1038/323533a0.

[76] Seckler, M., Heinz, S., Forde, S., Tuch, A. N., and Opwis, K., "Trust and distrust on the web: User experiences and website characteristics." *Computers in human behavior*, 2015, vol. 45, pp. 39–50. DOI: 10.1016/j.chb.2014.11.064.

[77] Sheil, H., Rana, O., and Reilly, R., "Predicting purchasing intent: automatic feature learning using recurrent neural networks." *arXiv preprint arXiv:1807.08207*, 2018. DOI: 10.48550/arXiv.1807.08207.

[78] Sismeiro, C., and Bucklin, R. E., "Modeling purchase behavior at an e-commerce web site: A task-completion approach." *Journal of marketing research*, 2004, vol. 41, no. 3, pp. 306–323. DOI: 10.1509/jmkr.41.3.306.35985.

[79] Șoavă, G., and Raduteanu, M., "Optimizing Ecommerce sites through the use heat map." *European International Journal of Science and Technology*, 2013, vol. 2, no. 4, pp. 53–64.

[80] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. and Salakhutdinov, R., "Dropout: A Simple Way to Prevent Neural Networks from Overfitting." *Journal of Machine Learning Research*, 2014, vol. 15 , no. 56 , pp. 1929–1958. DOI: 10.5555/2627435.2670313.

[81] Suchacka, G., and Chodak, G., "Using association rules to assess purchase probability in online stores." *Information Systems and e-Business Management*, 2017, vol. 15, no. 3, pp. 751–780. DOI: 10.1007/s10257-016-0329-4.

[82] Suh, E., Lim, S., Hwang, H., and Kim, S., "A prediction model for the purchase probability of anonymous customers to support real time web marketing: a case study." *Expert Systems with Applications*, 2004, vol. 27, no. 2, pp. 245–255. DOI: 10.1016/j.eswa.2004.01.008.

[83] Tang, J., and Wang, K. "Personalized top-n sequential recommendation via convolutional sequence embedding." *In Proceedings of the eleventh ACM international conference on web search and data mining*, 2018, pp. 565–573. DOI: 10.48550/arXiv.1809.07426.

[84] Vankerschaver, J., *hsmmlearn*. [Online]. https://github.com/jvkersch/hsmmlearn (accessed 4 June 2025).

[85] Viterbi, A., "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm." *In IEEE Transactions on Information Theory*, 1967, vol. 13, no. 2, pp. 260–269. DOI: 10.1109/TIT.1967.1054010.

[86] Werbos, P. J., "Backpropagation through time: what it does and how to do it." *Proceedings of the IEEE*, 2002, vol. 78, no. 10, pp. 1550–1560. DOI: 10.1109/5.58337.

[87] Xu, H., Li, Z., Chu, C., Chen, Y., Yang, Y., Lu, H., Wang, H. and Stavrou, A., "Detecting and characterizing web bot traffic in a large e-commerce marketplace." *Computer Security*, 2018, pp. 143–163. DOI: 10.1007/978-3-319-98989-1_8.

[88] Yeo, J., Kim, S., Koh, E., Hwang, S. W., and Lipka, N., "Predicting online purchase conversion for retargeting." *In Proceedings of the Tenth ACM international conference on web search and data mining*, 2017, pp. 591–600. DOI: 10.1145/3018661.3018715.

[89] Yu, S., "Hidden semi-Markov models." *Artificial intelligence*, 2010, pp. 215–243. DOI: 10.1016/j.artint.2009.11.011.

[90] Zumstein, D. and Kotowski, W., "Success factors of e-commerce-drivers of the conversion rate and basket value." *18th International Conference e-Society 2020*, 2020, pp. 43–50. DOI: 10.33965/es2020_202005L006.