VALUE OF INCORPORATING CUSTOMER PURCHASE BEHAVIOUR IN PREDICTING ONLINE RETURNS: AN INTEGRATED ANOMALY DETECTION APPROACH AND COUPON DISTRIBUTION

A Master's Thesis

by
RANA KAYA

Department of
Management
İhsan Doğramacı Bilkent University
Ankara
May 2025

VALUE OF INCORPORATING CUSTOMER PURCHASE BEHAVIOUR IN PREDICTING ONLINE RETURNS: AN INTEGRATED ANOMALY DETECTION APPROACH AND COUPON DISTRIBUTION
RANA KAYA
Bilkent University 2025

To My Beloved Grandma, Resmiye Savaş

VALUE OF INCORPORATING CUSTOMER PURCHASE BEHAVIOR IN PREDICTING ONLINE RETURNS: AN INTEGRATED ANOMALY DETECTION APPROACH AND COUPON DISTRIBUTION

The Graduate School of Economics and Social Sciences
of
İhsan Doğramacı Bilkent University

by

RANA KAYA

In Partial Fulfillment of the Requirements for the Degree of
MASTER OF SCIENCE

THE DEPARTMENT OF
MANAGEMENT
İHSAN DOĞRAMACI BİLKENT UNIVERSITY
ANKARA

May 2025

VALUE OF INCORPORATING CUSTOMER PURHCASE BEHAVIOUR IN PREDICTING ONLINE RETURNS: AN INTEGRATED ANOMALY DETECTION APPROACH AND COUPON DISTRIBUTION
By Rana Kaya

I certify that I have read this thesis and have found that it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science in Management.

Fehmi Tanrısever
Advisor

I certify that I have read this thesis and have found that it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science in Management.

Yasemin Limon
Examining Committee Member

I certify that I have read this thesis and have found that it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science in Management.

Eda Yücel
Examining Committee Member

Approval of the Graduate School of Economics and Social Sciences

---
Refet S. Gürkaynak
Director

ABSTRACT

VALUE OF INCORPORATING CUSTOMER PURCHASE BEHAVIOR IN PREDICTING ONLINE RESULTS: AN INTEGRATED ANOMALY DETECTION APPROACH AND COUPON DISTRIBUTION

Kaya, Rana

M.S., Department of Management

Advisor: Assoc. Prof. Fehmi Tanrısever

May 2025

This study explores the influence of customer behavior on product return rates by applying anomaly detection techniques to customer-level transaction data. Using data from a European e-commerce company, various algorithms are used to identify anomalous transactions, which are then incorporated into a logistic regression model to assess their predictive value for returns, with performance measured by AUC. The best-performing model is used in a second phase to guide coupon distribution strategies, estimating return probabilities under both coupon and non-coupon conditions. These estimates inform heuristics policies within a dynamic programming framework to optimize coupon allocation and evaluate revenue outcomes.

Keywords: Product Returns, Anomaly Detection, Customer Behavior, Coupon Distribution, Predictive Modeling

ÖZET

ÇEVRİMİÇİ İADELERİN TAHMİNİNDE MÜŞTERİ SATIN ALMA DAVRANIŞLARININ DEĞERİNİN ENTEGRE BİR ANOMALİ TESPİT YAKLAŞIMI VE KUPON DAĞITIMI İLE İNCELENMESİ

Kaya, Rana

Yüksek Lisans, İşletme

Tez Danışmanı: Doç. Dr. Fehmi Tanrısever

Mayıs 2025

Bu çalışma, müşteri davranışlarının ürün iade oranları üzerindeki etkisini, müşteri düzeyindeki işlem verilerine uygulanan anomali tespiti teknikleri aracılığıyla incelemektedir. Bir Avrupa e-ticaret şirketinden elde edilen veriler kullanılarak çeşitli algoritmalarla işlemler belirlenmiş, ve bu anomaliler, iade davranışını tahmin etmek amacıyla AUC ile değerlendirilen bir lojistik regresyon modeline entegre edilmiştir. En iyi performans gösteren model, çalışmanın ikinci aşamasında kupon dağıtım stratejilerini yönlendirmek için kullanılmıştır. Bu aşamada, her işlem için kuponlu ve kuponsuz senaryolarda iade olasılıkları tahmin edilmiş, ve bu tahminler, kupon tahsisini optimize etmeye yönelik dinamik programlama temelli sezgisel politikalara girdi sağlamıştır. Modelin etkinliği, gelir sonuçları üzerinden değerlendirilmiştir.

Anahtar Kelimeler: Ürün İadeleri, Anomali Tespiti, Müşteri Davranışı, Kupon Dağıtımı, Tahmine Dayalı Modelleme

ACKNOWLEDGMENTS

Pursuing a master's degree at Bilkent University has been both a valuable and challenging journey. As someone who had just graduated from undergraduate studies and started a new job, this path demanded a great deal. There are many people who stood by me throughout this journey and whom I owe thanks to—but for the first time, I would like to give priority to myself and express gratitude to myself. Rana, despite all the difficulties, you completed this degree, and I am most proud of you.

First and foremost, I would like to thank my advisor, Assoc. Prof. Fehmi Tanrısever, whose unwavering support and belief in me I deeply felt throughout the process. I am also sincerely grateful to my co-advisor, Asst. Prof. Yasemin Limon, who was always there to help whenever I needed guidance. In addition, I extend my heartfelt thanks to Asst. Prof. Feray Tunçalp, who supported me as much as my advisors and broadened my perspective in areas unfamiliar to me.

In a family where nearly everyone holds a Ph.D. or a master's degree, it was of course unthinkable to remain with just a bachelor's degree. I would like to thank my family, who supported me endlessly throughout my entire academic life and guided me toward success—especially my dearest grandmother, Resmiye Savaş, who, despite not having had the opportunity to study due to financial limitations, guided not only her own daughters but also me, and opened the way for our education. I wish you had written this thesis many years ago and fulfilled your dreams of pursuing education.

I also owe immense gratitude to my future husband, Berk Türk, who never left my side throughout these academically intense times. Your support, including spending weekends working with me, kept me motivated. Without it, I likely wouldn't have been able to accomplish this.

Finally, I would like to thank Prof. Dr. Savaş Dayanık, who helped me discover my passion for data.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

The share of online shopping within overall retail consumption has increased significantly in recent years. Although in 2000, e-Commerce constituted only about 1% of the total retail sales in the United States, this number has risen to 15% in 2023, marking a notable transformation in customer purchasing behavior (Statista, 2024b) . Until the COVID-19 outbreak, the online shopping rate increased slowly yearly, not even close to 1%. However, the pandemic caused a significant increase; from 2019 to 2020, the annual growth rate was 4%. In global, the user penetration rate of e-commerce is expected to hit 42% in 2025 , which is estimated to hit 50% by 2029 (Statista, 2024a). According to the Technology Acceptance Model, the reasons why the e-commerce rate is increasing other than the COVID-19 effect are usefulness (S.-C. Liu & Liao, 2001), ease of use (Moon & Kim, 2001), and enjoyment (Perea y Monsuwe´ et al., 2004). Lots of other reasons could be counted, i.e., product range and not wasting time in malls. Even though e-commerce is still not preferred as much as brick-and-mortar stores, by looking at total retail sales rates, 20 % of those sales are returned, a sharp contrast to the 9% return rate observed in brick-and-mortar stores (Rakuten, 2025). In 2022 alone, retail returns in the United States cost $817 billion,, with online retail responsible for about one-quarter of that total. Of the $ 1.3 trillion in online sales recorded that year, roughly 16% were returned (Statista,

2025). According to Robertson et al. (2020), two primary causes contribute to this high return rate, which we could not see in the brick and mortar stores: (i) consumers may be uncertain about product quality, fit, or size, and (ii) lenient return policies can inadvertently encourage bulk purchasing and subsequent returns.

The operational and financial costs of those returns are substantial. In 2022, it costs $33 for retailers to process a return (Narvar Team, 2023). Many items experience multiple shipping cycles before finally being sold to their final user or discarded, which leads to margin erosion. These challenges are especially experienced in the fashion sector, where European online return rates exceed 35% (Statista, 2023). Beyond these monetary losses, the environmental impact is also considerable. Online shopping produces 4.8 times more waste. The reason is the excessive packaging waste and carbon emissions involved in transporting multiple restocking and disposing of returned goods in multiple shipping cycles. Therefore, e-commerce returns cause pressure on retailers due to the increasing impact of sustainability on businesses and corporate social responsibility regulations (CSR). On the other hand, since lenient return policies increase sales (Janakiraman et al., 2016) , increase loyalty and have a positive correlation with perceived fairness, online retailers often shy away from imposing restrictions on return policies such as reimbursing half of the price or strict right of withdrawal periods. As a result, the lenient return policy had a ”halo” effect on the other factors (Wang et al., 2020). Consequently, the challenge is to find a way to address the issue without decreasing the halo effect, and it is imperative to develop sophisticated methods for predicting and managing returns without undermining customer satisfaction. However, there is a significant gap in the literature about how a consumer’s holistic shopping behaviour—beyond the attributes of a single transaction—drives return decisions. The traditional return management literature focuses on return policies, supply chain planning, and reverse logistics. Although these studies offer valuable operational insights, they often neglect the behavioural dimension, specifically how an individual’s past purchasing patterns and anomalies therein can forecast future returns. To bridge this gap, we propose an integrated, two-stage approach that unites consumer behaviour analytics with operational decision-making. Our primary goal is to develop a prediction model that combines conventional logistic regression with anomaly detection techniques. These methods, commonly employed for fraud detection in banking, enable us to identify atypical purchase behaviours—such as unusually high-value or inconsistent brand preferences—that may correlate with a higher likelihood of returns. By integrating this anomaly attribute into the prediction process, we capture abusive or uncharacteristic shopping patterns that conventional models might overlook. Our empirical findings are based

on transaction-level data from one of Europe's leading e-commerce firms. Through analyzing the influence of anomaly detection, we use machine learning techniques- specifically, Association Rule Mining, Local Outlier Factor (LOF), Cluster-Based Local Outlier Factor (CBLOF), Isolation Forest and Density-Based Spatial Clustering of Applications with Noise (DBSCAN) for anomaly detection, K-Means for clustering to integrate Association Rule Mining, and Logistic Regression for modelling. Secondly, we incorporate these improved return predictions into a multi-stage stochastic dynamic program designed to allocate coupons to shoppers in real time to decrease the customer's return possibility. Because high-dimensional dynamic programming is computationally intractable, we propose near-optimal heuristics. The remainder of the thesis is organized as follows: The literature review is provided in Chapter 2. Chapter 3 describes the data we have, its structure and the steps we follow for cleaning and preprocessing. Chapter 4 is dedicated to model selection and coupon distribution. Lastly, in Chapter 5, the conclusion of the study and suggestions for future work are presented.

CHAPTER 2

LITERATURE REVIEW

This study is grounded in operations management literature, particularly in the domains of predictive modelling and optimization, as well as in marketing literature related to e-commerce and product returns. Also, we mentioned cybersecurity literature since it extensively uses anomaly detection practices in the area. Our primary objective is to develop a data-driven predictive model by logistic regression that shows customer behaviour using anomaly detection as a first step. Therefore, this chapter includes a theoretical background of logistic regression, LOF, CBLOF, Isolation Forest, Association Rule Mining and DBSCAN. As a second step, we integrate these improved return predictions into a multi-stage dynamic program and apply heuristic methods that allocate coupons to customers in real-time, aiming to reduce the likelihood of returns. Therefore, this chapter also provides DP's theoretical background and the heuristics we use.

2.1 Operations Management

In the context of operations management, consumer return management focuses commonly on four areas: return policy design, planning and execution (P&E), consumer behavior and

return management.

2.1.1 Return Policy Design

Return policies are categorized based on their leniency levels. Some retailers offer flexible return policies, which allow full refunds anytime for any reason. In contrast, less lenient policies apply restrictions, such as the deduction of refund or limitations on the allowable time to return. Research on return policy design policies to analyze which return policy could function better with given constraints or analyze the operational consequences of current return policies and generate managerial strategies accordingly. X. Su (2009)studied the impact of full and partial returns policies on supply chain performance and proposed strategies to coordinate the supply chain in the presence of consumer returns. Shang et al. (2017) used the data collected from eBay, where identical products were sold with different return policies, to investigate whether consumers value full or partial returns policies.

2.1.2 Planning and Execution

The Planning and Execution (P&E) domain captures how returns affect forward supply chain management and managerial actions. Literature consists of decision-making in the supply chains, such as pricing, inventory, channel, assortment, and quality, as well as in managerial actions, such as competition and coordination. Bernstein et al. (2019) conveyed a study that clusters customers into segments in terms of their demographics, applied a multiarmed bandit model and learned customer preferences of that cluster to create a better assortment.

2.1.3 Return Management

While (P&E) focuses on decision-making in the forward supply chain, the return management (RM) domain considers the returns management in the reverse supply chain. Therefore, RM is considered a sub-field of a closed-loop supply chain, responsible for the disposition of returns, recycling, processing and acquisition. RM field has several types of research related to return prediction. However, they investigated forecasting the returns at the factory level. Toktay et al. (2004) and Tsiliyannis (2018) considered predicting returns

in manufacturing to find out how a company can adjust their remanufacturing line with the returned products.

2.1.4 Customer Behavior

The Customer Behavior area focuses on consumer decision-making in sales and return processes. Their studies generally test consumer behaviour empirically and classify them in terms of their behaviours (Petersen & Kumar, 2009). Rohm & Swaminathan (2004) studied on online shopping behaviours. They found out that there are four different online purchasing behaviours. Those behaviour groups have different product selection behaviours. Bechwati & Siegal (2005) studied customer decision processes with customer choice and behaviour, which would affect post-purchase behaviour. Therefore, it is crucial to understand customer behaviours and demographics to understand product return.

2.2 Prediction and Empirical Studies in Online Shopping

The operations management fields mentioned above emerged much earlier than prediction and empirical studies in online shopping. Even though it emerged later, forecasting returns is significant for operations management since it is an input for reverse supply chain processes, production planning, and inventory management. Therefore, researchers try to predict returns accurately.Zhou et al. (2016) predicted the quantity and probability of recycled parts of returned products. Petersen & Kumar (2009) tested the several circumstances as a hypothesis. Based on Sherry (1983), that a gift has a social meaning rather than economic value, they developed and tested a hypothesis suggesting that purchases made as gifts are less likely to be returned. Cui et al. (2020) developed a prediction model and estimated the return volume for an automotive accessories manufacturer. They used several prediction variables such as year, month, product type, retailer type and sales volume. They used model selection models to find which model and variables have the best prediction accuracy to predict return volume. Most of the literature in this area focuses on fitting prediction models and ignores customer behaviour. Therefore, there is a gap in the literature about combining customer behaviour insights with prediction models.

2.3 Logistic Regression

Logistic regression is a statistical method used to model the probability of a binary outcome as a function of one or more independent variables. It estimates the likelihood of a given input belonging to a particular category (typically coded as 0 or 1) by using the logistic (sigmoid) function, which maps it into the (0,1) interval. The coefficients are estimated using maximum likelihood estimation (MLE), and the model assumes that the outcome's log odds are linear combinations of the input variables.

Early developments in modelling binary outcomes began with Bliss (1934), who introduced the probit model in the context of toxicology and bioassay studies. The probability of survival of insects was derived from the inverse cumulative distribution function (CDF) of the standard normal distribution, which is probit. Logistic function, on the other hand, was invented in the 19th century to describe the population growth of young countries such as the United States at that time (Cramer, 2002). Up to the 1930s, the logistic function was not commonly used until Joshpe Berkson, who was a chief statistician at Mayo Clinic, offered to use the logit function in the statistical methodology bio-assay. Until the 1960s, the logistic function was used equally with the probit function, while researchers realized that both probit and logistic function could be used in fields beyond bio-assay. The first time logistic regression concept was mentioned in 1958 by Cox (1958). Cox mentioned that events could take one of two outcomes: 1 or 0; therefore, each trial series would give a sequence of binary outcomes. Hence, he claimed that the probability of occurrence of those binary outcomes could be depended on one or more independent variables, which is simply the definition of logistic regression.

Before building our model, we reviewed academic articles that discuss logistic regression as a concept. First, we reviewed articles that mentioned core concepts of logistic regression. Peng et al. (2002) suggested that core logistic regression concepts are overall model evaluation, statistical tests of individual predictors and goodness-of-fit statistics. There are tests that we could rely on to assess the overall model evaluation and the statistical importance of variables. While comparing two models, the general model and the nested one, the likelihood ratio test is used. It concludes whether the nested model or the general model is better to explain dependent variables. This test is also used with the null-only intercept - model to determine whether the added variables are logical. On the other hand, the statistical significance of a coefficient β is tested with Wald chi-square statistics. Often, the tested hypothesis is whether a coefficient is statistically different than zero. If the null

hypothesis is not rejected, the coefficient of that variable is not statistically different from zero, and it could be removed from the model. Goodness-of-fit statistics evaluate how well the logistic regression model aligns with the observed outcomes. The goodness-of-fit test for logistic regression is the Hosmer-Lemeshow (H-L) test. If the p-value for the test is not statistically significant, it indicates that the model adequately fits the data.

In addition to conceptual studies, we reviewed articles that apply logistic regression within operations management. These papers were examined with a specific focus on four key aspects: first, the methods used to assess predictive power; second, the evaluation of model fit; and third, the techniques employed to address unadjusted estimates. Kadiyala et al. (2021) tried to classify customer responses to the gift card promotion e-mail while they used six techniques, one of them being logistic regression. To compare the prediction performance of six models, they used the Kappa metric, which accounts for the uneven distribution of customer response variables. Hilafu et al. (2024) tested whether customized items have a lower return rate or not. They explained that the returns with different variables interacted with customization. Richardson et al. (2007) developed a logistic regression model that estimates the click-through rates (CTRs) of new advertisements based on the characteristics of the ads, the associated terms, and the advertisers. Author et al. (2025) studied textual data from online patient reviews to predict indicators of physicians' professional ethics with several models, one of them is logistic regression. Those ethical indicators are validated as predictors of adverse physician outcomes, including drug-related deaths, disciplinary actions, malpractice claims, and rent-seeking behaviours. They divided the dataset into an 80% training set and a 20% test set and employed five-fold cross-validation to optimize the hyperparameters of the machine learning models. To solve the sanction rarity issue, they applied class weighting and utilize the Synthetic Minority Over-sampling Technique (SMOTE). They used the area under the receiver operating characteristic (ROC) curve (AUC), along with precision, recall, and F1 score, to assess classification performance.

2.4 Anomaly Detection

In this research, we use anomaly detection techniques to model consumer behaviour. According Barnett & Lewis (1984), anomalies refer to observations inconsistent with the rest of the data. In anomaly detection, various methods are used, such as statistical tests, machine learning algorithms, deep learning models, clustering techniques, and hybrid

approaches, which integrate multiple methodologies to improve detection performance. According to Hodge & Austin (2004), anomaly detection methods could be categorized into three main approaches. The first approach involves identifying anomalies within datasets that lack prior labelling. In such cases, unsupervised learning methods are used, operating under the assumption that anomalous instances are rare and statistically distinct from data. Algorithms such as Isolation Forests, One-Class SVM, and clustering-based techniques (e.g., DBSCAN, CBLOF) are commonly used, aiming to partition the data in a way that highlights deviations without relying on pre-defined class labels. The second approach is based on supervised learning, where the dataset is pre-labelled as either normal or anomalous. This framework treats anomaly detection as a traditional classification task, where algorithms such as logistic regression, decision trees, support vector machines, and deep neural networks are trained to distinguish between normal and anomalous examples. Supervised methods generally offer high detection accuracy when sufficient labelled data is available, but it is the costly and time-consuming process of obtaining high-quality labelled datasets, especially in fields where anomalies are rare and labelling requires expert knowledge. The third approach is semi-supervised anomaly detection, which combines elements of both supervised and unsupervised learning. In this setting, models are typically trained on a dataset comprising predominantly normal observations, and anomalies are detected as deviations from the learned model of normality.

The late 1990s and early 2000s are significant eras for anomaly detection research. Before 1998, most studies relied only on known statistical models to catch outliers.Barnett & Lewis (1984) provided over 100 discordancy tests to identify outliers in experimental data. These tests were developed for various distributions, including normal, exponential, Poisson, and binomial, and divided into four categories according to whether the population parameters (such as mean and variance) are known or unknown, the expected number of outliers, and the suspected location of the outliers (upper, lower, or both tails). Even though there are many testing options, in the data mining context, most of the time, no standard distribution could model the observed data. Fitting the observed distributions into standard distributions and choosing one of the tests accordingly could take great effort. With this motivation, Knorr & Ng (1997) suggested it is better to use a point p in dataset D considered as an outlier if at least k points in the dataset are farther than a specified distance R from p. They proposed a distance-based outlier model, emphasizing a non-parametric definition of outliers, making it independent of any statistical distribution. Shortly after, Breunig et al. (2000) introduced the Local Outlier Factor (LOF), which allowed for the detection of local anomalies through density-based scoring rather than distance-based. Later on,He

et al. (2003) suggested that since those previous techniques define outliers with the full dimensional distances and densities of the points from one another, a model that uses both clustering and outlier detection could be better for computation cost and anomaly detection power. They proposed Cluster-Based Local Outlier Detection (CBLOF). On the other hand, the DBSCAN algorithm, although originally developed for clustering, contributed significantly by identifying noise points now widely interpreted as anomalies (Ester et al., 1996). Together, these contributions laid the theoretical and practical foundation for non-parametric anomaly detection methodologies. We used those models to compare which ones do better to understand customer anomalous transaction behaviours.

Anomaly detection techniques are mostly used in security-related fields. To understand how researchers use anomaly detection techniques, we select cyber-intrusion detection, credit card fraud detection and textual data anomaly detection fields are selected and nine papers were reviewed. Generally used techniques are classification-based models, which use neural networks, support vector machine (SVM) rule-based and clustering-based techniques. Kumar & Sangwan (2012), Patcha & Park (2007) and M.-Y. Su et al. (2011) are reviewed in cyber-intrusion detection; while, Aleskerov et al. (1997), Bolton & Hand (2002) and Srivastava et al. (2008) were reviewed in credit card fraud detection. Lastly, Allan et al. (1998) were reviewed in the textual data anomaly detection.

2.5 Clustering Techniques

We use clustering to cluster the products and customers and integrate them into the association rule mining model. Hence, literature on clustering has been reviewed as well. As mentioned in Saxena et al. (2017), clustering techniques are used in many fields, such as image segmentation, customer segmentation and anomaly detection. All the unsupervised anomaly detection techniques are also in the area of clustering as well. Saxena et al. (2017) mentioned the importance of classification in various data processing applications and provided an overview of various clustering methods, including hierarchical clustering, partitional clustering, k-means clustering, graph theoretic methods, mixture density-based methods and grid-based clustering techniques. Also, Saxena et al. (2017) highlights the importance of similarity measures in clustering objects by determining the groupings among data points.

Related papers are reviewed in the operations management field to understand how to

apply clustering algorithms. Kandula et al. (2024) studied to find optimal packaging box sizes to decrease inefficiency in the transportation of e-commerce goods. In this problem, they struggled with two extreme ways to design cartons for transportation. One extreme is to create a separate box for each SKU, perfectly matching its size — which leads to perfect space utilization but a huge number of different boxes. The other extreme is to design one large box that can fit all SKUs — which minimizes the number of box types but results in poor space efficiency. Grouping SKUs based on their sizes so that they are close together in three-dimensional space is an NP-hard problem. To solve it efficiently, they used a greedy heuristic from unsupervised learning — the k-means clustering algorithm as an initial solution.

2.6 Coupon Distribution

Coupon distribution in our study aims to decrease return probability by offering coupons to customers. Therefore, we will increase revenue and decrease environmental waste. Coupon distribution is one of the most fundamental tools in marketing and customer retention strategies. Their importance is derived from their ability to encourage purchases, increase customer retention and profit accordingly. However, the strategy behind coupon distribution is the most vital part of the process since poorly optimized couponing can lead to revenue loss and customer habituation (Blattberg & Neslin, 1990). As a result, the focus of modern research has shifted toward targeted, data-driven, and dynamically optimized coupon strategies.

The economic reasons behind couponing emerged from price discrimination and consumer behaviour theories. Narasimhan (1984) argued that coupons allow firms to price discriminate among heterogeneous consumers, which allows them to attract price-sensitive segments while not discounting the prices for inelastic buyers. Kahneman & Tversky (1979) introduced the term prospect theory. It has two key concepts to help explain why consumers may overvalue gains from coupons compared to equivalent price discounts. Regarding reference dependence and framing concepts, customers tend to evaluate outcomes relative to a reference point. The discounted price is the reference point when a customer sees the product, while if the customer has a coupon and there is no discount for the product, the reference point is the original price. Therefore, a coupon is seen as a gain, whereas a discount is usually silently incorporated into the price. The loss aversion concept suggests that losing 10 TL feels worse than gaining 10 TL feels good. Having a coupon may feel

like avoiding a loss, especially when the consumer sees the original price and can reduce it; it creates a better emotional reaction than seeing a lower price.

Coupon targeting based on customer characteristics is also vital in coupon distribution literature. Zhang & Wedel (2009) suggested three different levels of promotional campaigns which are mass market, which uniformly distributes the same coupon to all customers; segment-specific, which targets specific consumer segments based on shared characteristics; and individual specific, which is highly personalized promotions tailored to individual consumer based on their behaviours and preferences. These strategies were implemented and analyzed both online and offline. They found that loyalty promotions are more profitable in online stores while competitive promotions give better results offline. Generally, individual-specific promotions offer only slight profit increases over segment-specific or mass-market promotions, particularly in offline settings. However, in product categories that are highly sensitive to promotions, individual-specific customization in online stores can lead to meaningful profit improvements.

The coupon distribution problem could also be considered a sequential decision-making task under uncertainty. For a sequential decision-making task under uncertainty, Dynamic Programming (DP) is a well-established framework introduced by Bellman (1952). DP provides a recursive approach to solving multistage optimization problems by breaking them into smaller subproblems. In the context of coupon distribution, DP can dynamically allocate limited promotional resources based on evolving customer behaviour and state information. Our aim in coupon distribution is to figure out how to give coupons to customers based on their return probability, which our predictive models calculate. Our model resembles the class of dynamic stochastic knapsack problems (DSKPs), which extend the classical knapsack problem to sequential and uncertain environments. DSKP was introduced by Bellman (1952) and extended twice in 1998 and 2001. It utilizes decision-making over time where the arrival of opportunities is stochastic, and resource usage is optimized dynamically.

CHAPTER 3

EMPIRICAL SETTING

The data obtained for this study is from an online retail company in Europe. We aim to develop predictive models with anomaly detection techniques to model customer behaviour. To that end, it is crucial to predict the returns better and determine the likelihood of product returns. As a second step, we aim to increase revenue with coupon optimization while decreasing the likelihood of product returns with coupons. This chapter describes the dataset and the procedures followed for data cleaning and preprocessing.

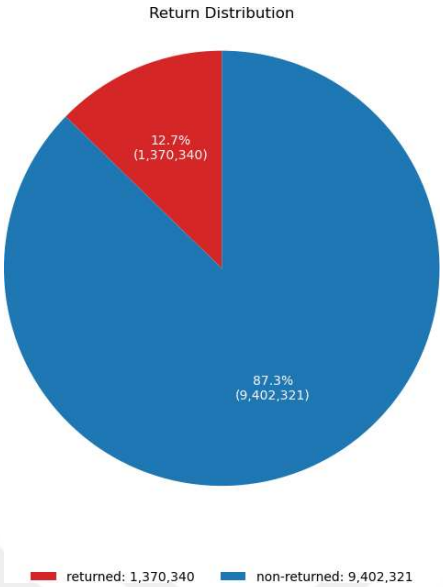

Figure 3.1: Return Distribution of Products

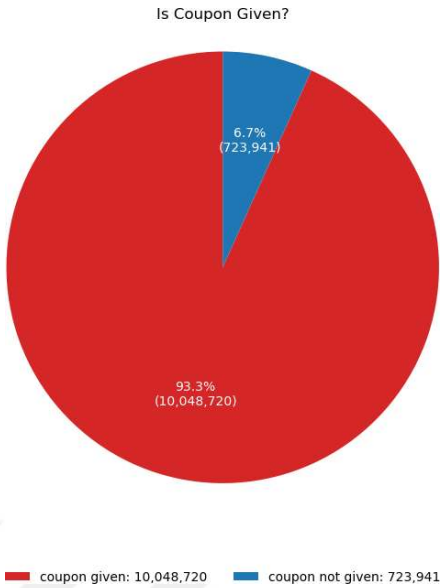

Figure 3.2: Coupon Distribution

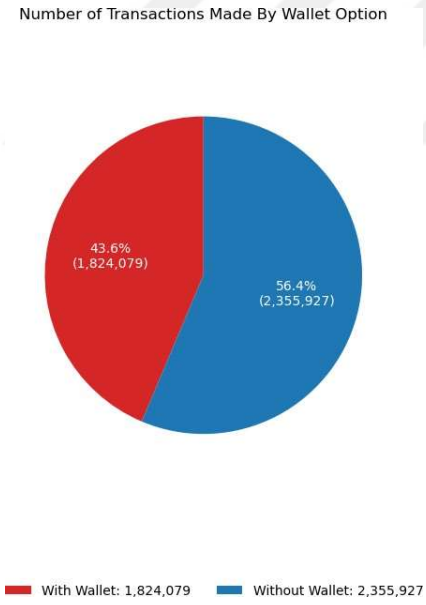

Figure 3.3: Number of Transactions Made Using Wallet Option

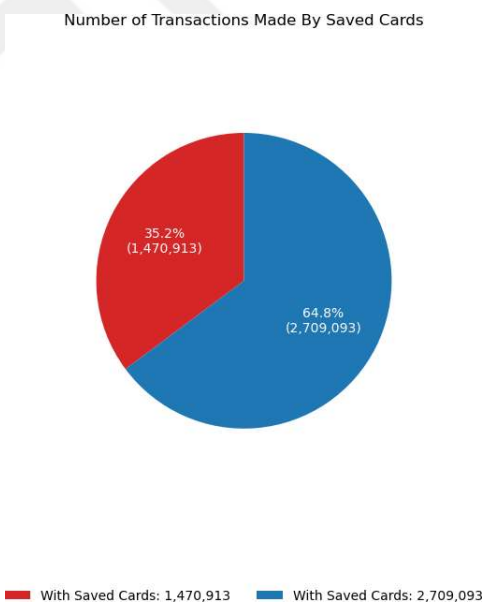

Figure 3.4: Number of Transactions Made Using Saved Cards

3.1 Data Description

The retailer provided eight transaction data files containing 4.180.006 unique transactions in total. Table 3.1 describes transaction data variables. The most critical variables in the dataset are *user_id_hashed* which provides uniqueness of the customers,*is_elite_user*, *original_price*, *ship_cost*, *coupon_discount* and *promotion_discount*. From those transactions, 10.772.661 units of products were sold and have been occurred between May 2021 and July 2021. Since we try to detect return and return probabilities in our first step of the study, the number of returns versus non-returns is significant to analyze. As shown in Figure 3.1, 9.402.321 products were not returned by the customers, while 1.370.340 products were returned to the retailer, which concludes that the return ratio is 12.72%. For the second step of the study, it is important to analyze the coupon distribution. Figure 3.2 demonstrates that only 6.7 % of the transactions were made with a coupon discount. Retailers provide different payment options, such as a digital wallet feature within both their mobile application and website, allowing customers to transfer money from their credit cards to their wallets for use in future transactions. According to transaction data, as seen in Figure 3.3, out of a total of 4,180,006 transactions, 1.824.079 were made with the wallet. Also, retailers allow customers to save their credit cards in the system to make transactions faster. According to transaction data, as is seen in Figure 3.4, 1.470.913 of the transactions were carried out with saved cards.

The retailer also provides the user demographics data, which contains 706.439 unique customers. Table 3.3 demonstrates variables of the user demographics dataset. From those variables, *gender* and *membership_date* could be significant to understand customer behaviour. As illustrated in Figure 3.5, 562.664 users are female, while 124.006 are male, and the rest of the customers did not disclose their gender information. Furthermore, as demonstrated in Figure 3.6, out of a total of 4,180,006 transactions, 3,571,479 were carried out by female customers and 549,837 by male customers. The rest of the transactions were done by customers who did not share their genders.

The dataset provided by the retailer contains detailed information about products, including information for 1.457.567 products. Among the variables, *gender_name* and *brand_id* are particularly significant, as they enable the analysis of whether customers tend to purchase products aligned with their gender and whether the brand can be considered as a reputable supplier. In Figure 3.7, it is observed that out of 1.457.567, 985.284 products are female products, 399.834 are male products, and 69.997 of the products are unisex.

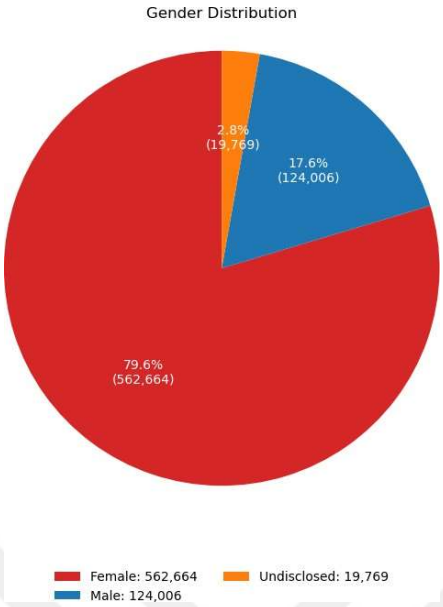

Figure 3.5: Gender Distribution of Users

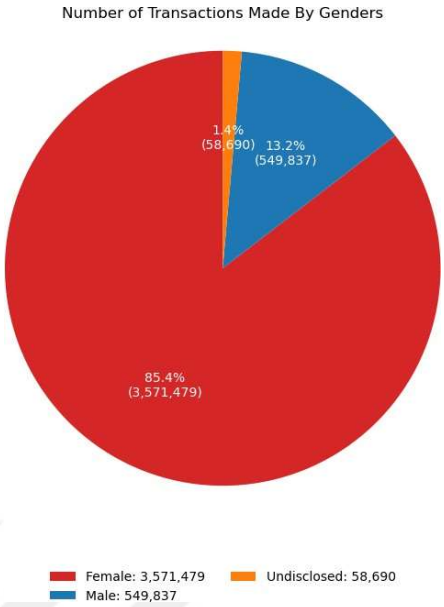

Figure 3.6: Gender Distribution Among Transactions

The rest of them are Unknown in terms of gender. Furthermore, Figure 3.8 shows user gender and product gender alignment. Out of 10.772.661 products bought, 7.923.997 of the products are aligned with the user's gender.

The data files described in this section were loaded into Python as data frames using the Pandas library.

3.2 Data Cleaning And Preprocessing

This section outlines the data cleaning and feature engineering processes, including creating new columns derived from existing features and removing uninformative rows and columns. Some columns are found to be uninformative since the information given is not related to our work or can be derived from other features. Some rows were excluded mainly due to containing misleading data or missing values. After excluding those rows and columns, we obtain our final dataset.

The variable *attribute_value*, which represents product sizes, was first standardized

Table 3.1: Transaction Dataset Variables

Variable Name	Definition
order_date	The date and time that transaction occurred
user_id_hashed	Unique User ID
is_elite_user	A customer could gain a elite user status after spending an amount specified by the retailer. This variable demonstrates whether the user is elite user
supplier_id_hashed	Unique supplier ID
order_line_item_id_hashed	Unique item ID assigned for every sold item within the basket
order_parent_id_hashed	Unique ID order, which is same for all the items within the transaction basket
product_content_id	Product ID which is independent of its size. Products of the same color but in different sizes share the same product_content_id
product_variant_id	Unique Product ID, each same colored product but in different sizes has different unique product_variant_id
original_price	List price of the product
discounted_price	The price paid for the product after the promotion discount is applied. Coupon discount is not included in this price
ship_cost	Shipping cost paid
coupon_id	ID of the coupon used for the transaction. If there is no coupon, this variable is NULL
coupon_discount	The Turkish Lira (TL) equivalent of coupon discount. If there is no coupon applied to this transaction, the variable is zero.
promotion_name	Name of the promotion applied if any
promotion_award_value	How much promotion discount applied. It can be a percentage or a flat discount
is_wallet_trx	Retailer allows its customer to use wallet within the app or website. This variable shows whether the transaction occurred using wallet
is_saved_card_trx	Customers could save their card information to retailer's web site or app. This variable shows whether this transaction occurred with saved card.
is_returned	Indicates whether the purchased product was returned by the customer.

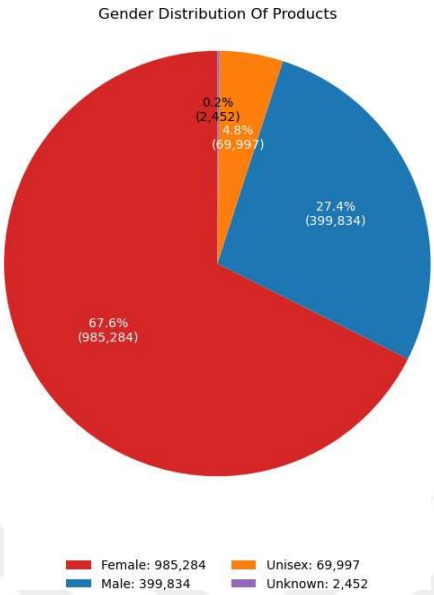

Figure 3.7: Gender Distribution of Products

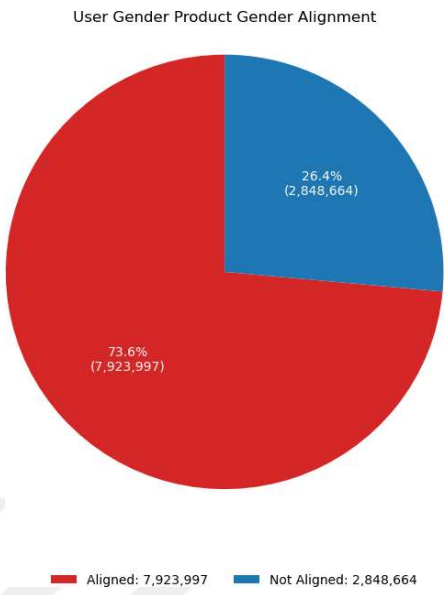

Figure 3.8: User-Gender Product Alignment

Table 3.2: Products Dataset Variables

Variable Name	Definition
product_id	Unique Product id independent of its color and size
product_content_id	Product id for the product where different sizes of the same colored products have the same product_content_id
product_variant_id	Product id of the product where Different sizes of the same colored products have different product_variant_id
product_name	Product name
brand_id	Unique brand id
brand_name	Brand name of the product
gender_id	Gender ID of the product
gender_name	Gender name of the product if any
category_id	Category ID of the product
category_name	Category name of the product
color_id	Color id of the product
color_name	Color of the product
supplier_color_name	Color given by the supplier
attribute_name	Name of the attribute, always equals to size in the dataset
attribute_value	Product size

Table 3.3: User Demographics Dataset Variables

Variable Name	Definition
user_id_hashed	Unique user id
birth_date	Birth date of the user if given
membership_date	Date that customer starts to be a member of the retailer's app or website
gender	User gender

during the data cleaning process. In the dataset, identical sizes have different representations. To ensure consistency across all transactions, a common representation is adopted. For example, sizes labelled as "3XL" are converted to "XXXL" to maintain uniformness. Since there is a presence of a wide variety of sizes across numerous product types, and in order to manage the dataset effectively and build a more stable model, only the standard sizes—XXS, XS, S, M, L, XL, and XXL—are selected for analysis, excluding less common or inconsistent size entries. Similar to this purpose, only transactions made in specific categories are selected since the size of the other products, such as jeans and trousers, varies a lot and could be two-sized (weight and height). Hence, only product categories T-Shirts, Shirts, Tank Tops, Blouses, Polo Shirts, Coats, Sweatshirts, Jackets, Sweaters, Cardigans, Dresses, Modest Dresses, and Evening and graduation Dresses are selected. In modelling, we use anomaly detection techniques at the user transaction level. To make a better analysis and not disturb the anomaly detection algorithms, customers that have more than 20 transactions in the mentioned categories are selected. It decreases the number of customers to 17.481. To assess the effect of both genders in the model, transaction data belonging to customers who did not disclose their gender was excluded from the data. After all the elimination, the total number of rows in our final dataset is 766.364.

From transaction, products and user demographics data, Table 3.4 presents the variables selected for our work and the rationale for their inclusion or justification of their exclusion. The new variables, which are the product of feature engineering, are mentioned in Table 3.5 in a detailed way. Table 3.6 shows the selected variables from the data used in the model. We also create interaction effects that could be important for the model. Two-way interaction variables were created for all selected features to capture potential interaction effects between them. Moreover, we add three-way interaction variables, as seen in Table 3.7. Lastly, squared and cubed terms of continuous variables *original_price*, *membership_duration*, *numeric_size*, *coupon_discount_ratio* and *promotion_discount_ratio* are added. With all those variables, the final dataset is created. In the final dataset, there are 24 variables, which are raw or feature-engineered, 276 variables are 2-way

interactions, nine variables are three-way interactions, and four variables are *coupon_id*, *coupon_id_coupon_ratio*, *promotion_name*, *promotion_name_promotion_ratio*.

Since we build logistic regression model, it is necessary to convert categorical variables into integers with one hot encoding. Specifically, *gender_name* is changed to *is_female*. *category_name* is also encoded and generated 12 new variables which are *category_name_TShirt*, *category_name_Shirt*, *category_name_TankTop*, *category_name_Blouse*, *category_name_PoloShirt*, *category_name_Coat*, *category_name_Sweatshirt*, *category_name_Jacket*, *category_name_Sweater*, *category_name_Cardigan*, *category_name_Dress*, *category_name_ModestDress* and *category_name_EveningGraduationDress*.

Since anomaly detection algorithms rely on identifying deviations of typical patterns, it is significant to understand each customer's past transactional behaviors to establish a representative behavioral baseline. To that end, we split the final data into two: past transactional data and new transactional data. Past transactional data is utilized to get customer behaviors and fit anomaly detection models so that we can predict whether the data coming from the new transactional data is anomalous or not. To create such dataset, we assume that if a customer has 21 transactions to 35 transactions, the previous 20 transaction of the customer is assumed as past transactions. If the customer has more than 35 transactions, its previous 60 % of the transactions are assumed as past transactions while rest of them is new transactions.

Before running logistic regression, correlations among raw and feature engineered predictive variables are checked with correlation matrix. Figure 3.9 shows that variables do not have any significant correlation between each other.

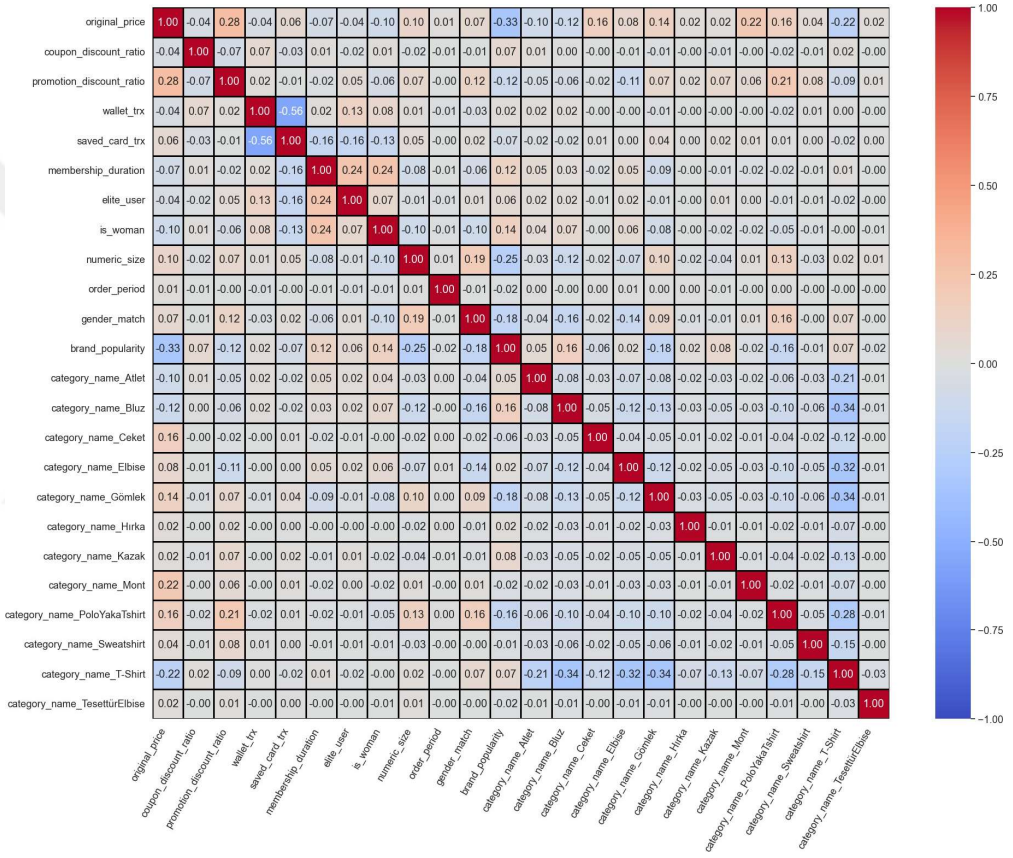

Figure 3.9: Correlation Matrix of Variables

Table 3.4: Comprehensive Dataset Variable Inclusion

Variable Name	Included	Reasoning
Transaction Dataset		
order_date	✗	Used to derive *order_period*, but original column excluded.
user_id_hashed	✓	Present in both the transaction and user demographics datasets; serves as a key for merging.
is_elite_user	✓	Indicates customer loyalty status; useful for behavioral analysis.
supplier_id_hashed	✗	Removed as the analysis does not target supplier behavior.
order_line_item_id_hashed	✗	Transaction level identifier which is not used in our work.
order_parent_id_hashed	✗	Groups items into baskets, but since the basket analysis out of scope, excluded
product_content_id	✗	Groups product variants, but it is out of our scope
product_variant_id	✗	Present in both the transaction and product datasets; serves as a key for merging.
original_price	✓	Demonstrate the listed price and is important customer behavior
discounted_price	✗	Original price is used. It is excluded to avoid multicollinearity with discount ratios and original price.
ship_cost	✗	Included for coupon distribution part but not used as a model feature.
coupon_id	✗	Excluded due to being textual
coupon_discount	✗	Used to derive *coupon_discount_ratio*.
promotion_name	✗	Excluded due to being textual.
promotion_award_value	✗	Used to calculate *promotion_discount_ratio*.
is_saved_card_trx	✓	Captures customer behavior for payment.
is_wallet_trx	✓	Captures customer behavior for wallet preference.
is_returned	✓	Target variable; indicates if the product was returned.
Products Dataset		
product_id	✗	Redundant with *product_variant_id*.
product_content_id	✗	Analysis of the content of the product is out of scope.
product_name	✗	Textual
brand_id	✗	Used to create variable named *brand_popularity*
brand_name	✗	Excluded in favor of *brand_id*.
gender_id	✗	Replaced by readable *gender_name*.
gender_name	✗	Used to create variable *gender_comparison* which shows product-gender alignment.
category_id	✗	Excluded due to redundancy with *category_name*.
category_name	✓	Useful for selecting specific categories. Also, given to the model with one-hot encoded versions
color_id	✗	Not informative for modeling.
color_name	✗	Not informative for modeling.
supplier_color_name	✗	Supplier-specific and textual; excluded.
attribute_name	✗	All column filled with 'size'
attribute_value	✗	Captures product size and is retained. It is manipulated to be in the same page. It is used to generate *numeric size* variable which shows numeric version of the sizes.
User Demographics Dataset		
user_id_hashed	✓	Required for joining with transaction data.
birth_date	✗	Low number of customer has birth date. Therefore, could not see effects on the model.
membership_date	✓	Proxy for tenure, important to see loyalty.
gender	✓	Used to generate gender alignment variable with products. Also, used to given to the model with one-hot encoded version.

Table 3.5: Feature Engineering Variables and Descriptions

Feature Engineering Variable	**Description**
coupon_discount_ratio	Ratio of coupon discount to original price. Derived from *coupon_discount* and *original_price* fields.
promotion_discount_ratio	Ratio of coupon discount to original price. Derived from *coupon_discount* and *original_price* fields.
membership_duration	Time duration between the membership date and the transaction date. Proxy for tenure and loyalty
is_woman	Binary indicator for female users. Used for gender-based segmentation.
numeric_size	Converted numeric version of product size. Helps standardize and model size effects.
order_period	Represents time period of day derived from order date. 1 if the transaction time is before 8 p.m
gender_match	Indicates whether the user's gender aligns with the product's target gender. Binary Variable that 1 indicates it is aligned
brand_popularity	Indicates how popular a brand is based on historical transaction counts.
category_name_TankTop	Indicates whether the product belongs to the 'Tank Top' category (one-hot encoded).
category_name_Blouse	Indicates whether the product belongs to the 'Blouse' category (one-hot encoded).
category_name_Jacket	Indicates whether the product belongs to the 'Jacket' category (one-hot encoded).
category_name_Dress	Indicates whether the product belongs to the 'Dress' category (one-hot encoded).
category_name_Shirt	Indicates whether the product belongs to the 'Shirt' category (one-hot encoded).
category_name_Cardigan	Indicates whether the product belongs to the 'Cardigan' category (one-hot encoded).
category_name_Sweater	Indicates whether the product belongs to the 'Sweater' category (one-hot encoded).
category_name_Coat	Indicates whether the product belongs to the 'Coat' category (one-hot encoded).
category_name_PoloShirts	Indicates whether the product belongs to the 'Polo Neck T-shirt' category (one-hot encoded).
category_name_Sweatshirt	Indicates whether the product belongs to the 'Sweatshirt' category (one-hot encoded).
category_name_TShirt	Indicates whether the product belongs to the 'T-Shirt' category (one-hot encoded).
category_name_ModestDress	Indicates whether the product belongs to the 'Modest Dress' category (one-hot encoded).

Table 3.6: Selected Variables and Their Types

Selected Variable	**Variable Type**
original_price	Raw
coupon_discount_ratio	Feature Engineered
promotion_discount_ratio	Feature Engineered
wallet_trx	Raw
saved_card_trx	Raw
membership_duration	Feature Engineered
elite_user	Raw
is_woman	Feature Engineered
numeric_size	Feature Engineered
order_period	Feature Engineered
gender_match	Feature Engineered
brand_popularity	Feature Engineered
category_name_TankTop	Feature Engineered
category_name_Blouse	Feature Engineered
category_name_Jacket	Feature Engineered
category_name_Dress	Feature Engineered
category_name_Shirt	Feature Engineered
category_name_Cardigan	Feature Engineered
category_name_Sweater	Feature Engineered
category_name_Coat	Feature Engineered
category_name_PoloTshirt	Feature Engineered
category_name_Sweatshirt	Feature Engineered
category_name_TShirt	Feature Engineered
category_name_ModestDress	Feature Engineered

Table 3.7: Selected Three-Way Interaction Terms

Interaction Term	**Reason**
coupon_discount_ratio × original_price × membership_duration	Explores how price and discount effects vary with tenure
coupon_discount_ratio × original_price × elite_user	Explores how price and discount effects vary by user status
coupon_discount_ratio × promotion_discount_ratio × membership_duration	How combined discount affects over tenure
coupon_discount_ratio × promotion_discount_ratio × elite_user	Tests if elite users react differently to overlapping discounts
original_price × numeric_size × order_period	Evaluates price-size sensitivity in daytime
original_price × numeric_size × gender_match	Assesses price-size relevance with gender alignment
original_price × promotion_discount_ratio × membership_duration	Investigates how promotional pricing varies over tenure
original_price × promotion_discount_ratio × elite_user	Measures effect of promotion on elite customers
saved_card_trx × coupon_discount_ratio × elite_user	Captures how saved card behavior interacts with discount usage and user type

CHAPTER 4

ANOMALY DETECTION MODEL SELECTION AND COUPON DISTRIBUTION

In the first part of the model selection section, we aim to build a baseline model that uses the variables in the final prepared data mentioned in the previous section. As a second step, we apply L2 regularization (Ridge Regression) to retain only the most statistically significant variables by shrinking the less important coefficients to zero. Then, we use the Area Under the Curve metric for each model to evaluate how well a logistic regression model classifies positive and negative outcomes. By doing so, we aim to show, without using any anomaly detection technique, how well we could predict the returns with those variables and create a benchmark for our models with anomaly detection techniques. Then, we apply anomaly detection models: Cluster-Based Local Outlier Factor (CBLOF), Isolation Forest, Association Rule Mining, Local Outlier Factor and Density-Based Spatial Clustering of Applications with Noise (DBSCAN) for transactions of each customer to flag any anomalous behaviour. Finally, we add those anomalous behaviour flags to our baseline model to compare the AUC of each model with the baseline model. In the coupon optimization part, we explain the dynamic programming that fits our model and why we cannot use dynamic programming. Lastly, we mentioned the optimization model, which is used for the upper bound, and the heuristics that we proposed.

4.1 Predictive Modeling Methods

4.1.1 Logistic Regression

In statistics, linear models are used to predict or analyze the dependent variable of the model if the variable follows a continuous distribution.

$$P(Y = 1 \mid X) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_k X_k \tag{4.1}$$

However, if the outcome variable is a binary variable which follows a Bernoulli distribution (dichotomous), the probability of occurrence of the outcome could be computed rather than the dependent variable itself. Even though the combination of dependent variables, the right-hand side of the equation, is linear, this probability of occurrence cannot be predicted with the linear models since probability cannot exceed one and cannot be smaller than zero. On that point, non-linear link functions, such as logit and probit functions, are used to map the linear combination of predictors onto the [0,1] interval. Both link functions can transform the linear predictor into a probability but differ. While the probit function uses the inverse standard normal cumulative distribution function (CDF) to model the probability as a linear combination of the independent variables, the logit model uses the logistic (sigmoid) function, which is the inverse of the logit link.

$$P(Y = 1 \mid X) = \frac{1}{1 + e^{(X\boldsymbol{\beta})}} \tag{4.2}$$

is the formulation of logistic function and

$$P(Y = 1 \mid X) = \Phi(X\boldsymbol{\beta})). \tag{4.3}$$

where it is assumed that

$$\Phi(z) = P(Z \leq z), \quad Z \sim \mathsf{N}(0, 1),$$

is the probit function Where:

- $\boldsymbol{\beta}$ is the coefficient vector of the logistic regression

- X is predictor variable vector
- $P(Y = 1 \mid X)$ is the probability of the outcome being 1
- e is the base of the natural logarithm.

Our work uses the logit function as the link function, thereby employing logistic regression to model binary outcomes. Logistic regression is a commonly used statistical technique for modelling the relationship between a binary outcome variable and one or more predictor variables.

The coefficients $\beta_0, \beta_1, \beta_2, \ldots, \beta_k$ are estimated using maximum likelihood estimation (MLE), and the model assumes that the log-odds of the outcome are a linear combination of the input variables. These coefficients demonstrate how a one-unit increase in each predictor variable affects the log odds of the outcome, assuming that all other variables remain the same. After fitting the model, logistic regression can be used to estimate the probability of the outcome variable being 1 for new data points once we give their corresponding predictor variable values.

To assess how well our model is fitted, we use Area Under The Curve. To assess accuracy, we decided not to use Youden's Index, chose a threshold, looked at True Positive and False Negative ratios, and maximised the F1 score.

4.1.2 Ridge Regression

Ridge regression, also called L2-regularized linear regression, reduces multicollinearity and overfitting by adding a penalty to large coefficient values. Ridge regression was first introduced in Hoerl & Kennard (1970). They introduced ridge regression as a solution to the instability of ordinary least squares (OLS) estimates when there is a multicollinearity between predictor variables. Although multicollinearity among predictor variables is not significant in our dataset, we apply ridge regression against overfitting to enhance the stability and generalizability of our estimates. Ridge regression helps mitigate overfitting by shrinking coefficients toward zero, especially in models involving many predictors or complex interactions, which is our case. The ridge regression objective function can be expressed as:

$$\hat{\boldsymbol{\beta}}^{ridge} = \arg\min_{\boldsymbol{\beta}} \left\{ \sum_{i=1}^{n} \left(y_i - \mathbf{x}_i^\top \boldsymbol{\beta}\right)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 \right\} \tag{4.4}$$

where

- β is the coefficient vector of the logistic regression
- y_i is observed value of the dependent variable for the *i*-th observation
- $\mathbf{x}_i$ is vector of predictor values for the *i*-th observation
- $\lambda \geq 0$ is the regularization parameter controlling the penalty's strength. Higher values lead to greater shrinkage of the coefficients
- $\sum_{i=1}^{n} \left(y_i - \mathbf{x}_i^\top \boldsymbol{\beta}\right)^2$ is the residual sum of squares, representing the total squared differences between observed and predicted values
- $\sum_{j=1}^{p} \beta_j^2$ is the L2 penalty term, which is the sum of squared values of the regression coefficients without intercept

4.2 Anomaly Detection Techniques

4.2.1 Association Rule Mining

Our rule-based model is built upon association rule mining, an unsupervised learning tech-nique used to uncover significant relationships among itemsets by analyzing co-occurrence patterns within the data. It was introduced by Agrawal et al. (1993), and its first application is on market basket analysis, where it sought to find associations between products that are frequently purchased together. After that, it is used in various domains, including anomaly detection, bioinformatics, web usage mining, and recommendation systems. To apply association rule mining, it is essential to have data in a transactional format. Given a dataset D consisting of transactions, where each transaction $T_i \in$ D and $T_i \subseteq$ I, we define I as the universal itemset, which contains all the possible items that may appear in any transaction in D. For example, in a retailer's transaction dataset, each transaction represents the set of products purchased by a customer in a single visit. The universal itemset, in this context, includes all product sets ever purchased across all transactions.

To illustrate, consider an itemset A that consists of a green blouse and pink trousers. If at least one customer has purchased this combination in a single transaction, then A is a valid itemset in the dataset. If A appears in 15 out of 100 total transactions, the support of itemset A is 0.15. This metric indicates how frequently itemset A appears in the dataset. To derive meaningful patterns, association rule mining identifies relationships between disjoint itemsets. Suppose A and B are two such itemsets, where $A, B \subseteq \mathbf{I}$ and $A \cap B = \emptyset$. An association rule takes the form:

$$A \Rightarrow B, \quad \text{where } A, B \subseteq \mathbf{I}, \; A \cap B = \emptyset \tag{4.5}$$

There are two significant measures that is used to demonstrate quality of the given rule. Support metric shows the ratio of transactions in dataset that contains both itemsets.

$$\text{Support}(A \Rightarrow B) = \frac{\text{Support}(A \cup B)}{|\mathbf{D}|} \tag{4.6}$$

Support quantifies how frequently the itemset $A \cup B$ appears in the dataset $\mathbf{D}$. Confidence, nevertheless, indicates how often items in A appear in transactions that contain B.

$$\text{Confidence}(A \Rightarrow B) = \frac{\text{Support}(A \cup B)}{\text{Support}(A)} \tag{4.7}$$

Association rules can model normal behavior in a dataset. Rules that has lower confidence or support are considered as anomalies.

4.2.2 Local Outlier Factor

Local Outlier Factor, which measures the local density deviation of a data point concerning its neighbours, is a density-based unsupervised anomaly detection algorithm and was first mentioned by Breunig et al. (2000). Unlike global methods, LOF can detect local anomalies, points that are outliers only in their local context. There are some notions to follow to compute the outlier score that the local outlier factor produces. Given the dataset $X = \{x_1, x_2, \ldots, x_n\}$, the Local Outlier Factor (LOF) score computes k-distance. For each point, x, k-distance is defined as the distance to its k-th nearest neighbour, which could be

shown as:

$$k\text{-distance}(x) = \max_{x' \in N_k(x)} d(x, x') \tag{4.8}$$

where $N_k(x)$ denotes the set of k nearest neighbors of point x, and $d(x, x')$ is the distance between x and a neighbor $x' \in N_k(x)$. The k-distance of x is the distance between x and its k-th nearest neighbor. Deciding whether the point x is an outlier based on its distance to the k-th neighbour could be misleading since its k-th neighbour could be very close to x. At that point, it is significant to demonstrate that the kth neighbour does not lie in a sparse region where its k-th n neighbour is large. To reflect this sparseness, the local outlier factor calculates the reachability distance for all k neighbours as follows:

$$\text{reach-dist}_k(x, x') = \max\{k\text{-distance}(x'),\ d(x, x')\} \tag{4.9}$$

and the local reachability density of x is the inverse of the average reachability distance from its neighbors:

$$\text{LRD}_k(x) = \left(\frac{1}{|N_k(x)|} \sum_{x' \in N_k(x)} \text{reach-dist}_k(x, x') \right)^{-1} \tag{4.10}$$

and the LOF score is computed as:

$$\text{LOF}_k(x) = \frac{1}{|N_k(x)|} \sum_{x' \in N_k(x)} \frac{\text{LRD}_k(x')}{\text{LRD}_k(x)} \tag{4.11}$$

The LOF score of a data point x is essentially the average ratio of the local reachability density (LRD) of x's neighbours to the LRD of x itself. In other words, it compares how densely packed the area around x is relative to how densely its neighbours are packed. If the LRD of x is low and its neighbours' LRDs are high, the LOF score will be high. It indicates that x lies in a region that is less dense than its neighbours; it indicates that x is an outlier. On the other hand, if x has a similar density to its neighbours, the LOF score will be close to 1, meaning that it is not an outlier.

4.2.3 Cluster Based Local Outlier Factor

Cluster-Based Local Outlier Factor (CBLOF) is suggested by He et al. (2003) due to possible disadvantages that full distance-based models create. It captures the structural characteristics of data through clustering, typically k-means, Gaussian Mixture Models, or hierarchical clustering. Each point is then assigned a score based on two criteria: distance to the cluster centroid and size of the cluster data point belongs to. Assume the results of the clustering algorithm CBLOF used (i.e. KMeans is applied) on dataset D and the number of clusters is k, is denoted as $C = \{C_1, C_2, \ldots, C_k\}$ where $C_i \cap C_j = \emptyset$ and $C_1 \cup C_2 \cup \cdots \cup C_k = D$. The CBLOF score is calculated as:

$$\text{CBLOF}(t) = \begin{cases} |C_i| \cdot \min\left(\text{distance}(t, C_j)\right) & \text{where } t \in C_i,\ C_i \in SC,\ C_j \in LC \\ |C_i| \cdot \text{distance}(t, C_i) & \text{where } t \in C_i,\ C_i \in LC \end{cases} \tag{4.12}$$

where

- SC denotes set of small clusters and LC denotes set of large clusters.
- $|C\,i|$ denotes the size of the cluster it belongs to.

Therefore, CBLOF is calculated with the size of its cluster and the distance between the data point and its closest large cluster if this record lies in a small cluster or the distance between the data point and the cluster it belongs to if this record belongs to a large cluster.

4.2.4 Isolation Forest

Isolation Forest (IForest) is introduced by F. T. Liu et al. (2008). The model is different from other anomaly detection techniques we discuss. It is an ensemble-based anomaly detection algorithm constructed to isolate anomalies instead of profiling normal points. It relies on the idea that anomalies are rare and different and thus require fewer random splits to isolate from the rest of the data. IForest randomly selects and randomly chooses a split value concerning the minimum and maximum value of that feature. It repeats the same process until an isolated forest is built. The overall model consists of an ensemble of such

trees, where the hyperparameter n_estimators determine the number of trees. The anomaly scoring, on the other hand, is as follows:

$$s(x, n) = 2^{-\frac{h(x)}{c(n)}} \tag{4.13}$$

where

- h(x) denotes the average path length required to isolate observation x across a collection of isolation trees.
- n is the subsample size used to build each tree.
- $c(n)$ is the average path length of unsuccessful searches in a Binary Search Tree of size n. Since trees have the same structure to Binary Search Tree, the estimation of average h(x) for external node terminations is the same as the unsuccessful search in BST. This normalization is important because, as the number of instances in the tree increases, the overall path lengths tend to grow as well. Hence, c(n) ensures comparability of path lengths. $c(n)$ is computed as

$$c(n) = 2 \cdot H(n-1) - \frac{2(n-1)}{n} \tag{4.14}$$

  where $H(i)$ is the i^{th} harmonic number, approximated as:

$$H(i) \approx \ln(i) + \gamma, \quad \text{with } \gamma \approx 0.5772 \tag{4.15}$$

- If $s(x) \to 1$, the point has a high probability to be anomaly.
- If $s(x) \approx 0.5$, the point is similar to the majority (average isolation depth).
- If $s(x) < 0.5$, the point is deeply embedded in the forest, which indicates a normal point.

4.2.5 Density-Based Spatial Clustering of Applications with Noise

DBSCAN, primarily a clustering algorithm, is introduced by Ester et al. (1996). However, detecting anomalies by labelling points that do not belong to any cluster is also effective.

Unlike other clustering techniques, DBSCAN does not require the number of clusters. It is based on the idea of separating high-density regions from low-density regions. The key idea is that a point belongs to a cluster if there are at least a minimum number of neighbouring points within a specified distance. Assume $\mathsf{X} = \{x_1, x_2, \ldots, x_n\}$ is the dataset of n observations. DBSCAN has several concepts to calculate. According to Ester et al. (1996) the eps-neighborhood of a point p, denoted by $N_\varepsilon(p)$, is described as:

$$N_\varepsilon(p) = \{q \in D \mid \text{dist}(p, q) \leq \varepsilon\} \tag{4.16}$$

Each point within a cluster's eps-neighborhood contains at least a minimum number of points (MinPts). However, some points are located within dense cluster regions, while others are not. The points are located in the high-density regions in clusters called core points, whereas others are border points. Therefore, the above given naive approach is not acceptable for better clustering. To ensure that all points belonging to the same cluster are included, MinPts should be selected as a relatively low threshold. This low threshold, however, should not hinder the density criterion, especially in noisy datasets. To that end, DBSCAN accepts a different condition: for each $p \in C$, there should exist another point $q \in C$, which is the core point, such that p lies within the eps-neighbourhood of q, and the neighbourhood $\mathsf{N}_\varepsilon(q)$ contains at least MinPts points. A point p is directly density reachable from a point q if

- $p \in \mathsf{N}_\varepsilon(q)$
- $|\mathsf{N}_\varepsilon(q)| \geq$ MinPts (core point condition)

This relation is **symmetric** in the special case where both p and q are core points; that is, if p is directly density-reachable from q, then q is also directly density-reachable from p. However, in the general case involving a core point and a border point, this relationship is **not symmetric**. Specifically, a border point may be directly density-reachable from a core point, but the reverse does not necessarily hold. A point y is *density-reachable* from a point x with respect to ε and *MinPts* if there exists a sequence of points

$$p = p_1, p_2, \ldots, p_k = y$$

such that p_{i+1} is directly density-reachable from p_i for all $i \in \{1, \ldots, k-1\}$.

Density-reachability is an extension of direct density-reachability. Note that two border points within the same cluster C may not be density-reachable from each other, as the core point condition might not be satisfied for both. However, there must exist at least one core point in C from which both border points are density-reachable. This motivates introducing the concept of density-connectivity, which captures the mutual relationship of such points within the same cluster. A point p is said to be density-connected to a point q with respect to the parameters eps and MinPts, if there exists a point $o \in D$ such that both p and q are density-reachable from o with respect to eps and MinPts. After establishing the concept of density-connectivity, a density-based cluster could be defined. Assume D is a dataset of points. A non-empty subset $C \subseteq D$ is called a *cluster* with respect to parameters eps and MinPts if the following conditions are satisfied:

- **Maximality:** For all $p \in C$ and $q \in D$, if q is density-reachable from p with respect to ε and *MinPts*, then $q \in C$.
- **Connectivity:** For all $p, q \in C$, p is density-connected to q with respect to ε and *MinPts*.

And since $C_1, C_2, \ldots, C_k$ be the clusters in the dataset D with respect to the parameters eps and $MinPts_i$, for $i = 1, \ldots, k$. The set of *noise points* is defined as follows:

$$\text{noise} = \{p \in D \mid \forall i \in \{1, \ldots, k\},\ p \notin C_i\} \tag{4.17}$$

4.3 Hyperparameter Optimization For Anomaly Detection Techniques

Anomaly detection techniques necessitate hyperparameter optimization. Those algorithms are sensitive to selected hyperparameters; therefore, an inappropriate choice may result in poor anomaly detection performance. In this section, we mentioned about the hyperparameter optimization heuristics. The optimization algorithms, such as grid search, are costly to implement since we apply anomaly detection techniques to each customer separately. In Table 4.1, the hyperparameters used in each anomaly detection technique can be seen.

Table 4.1: Hyperparameters of Anomaly Detection Algorithms

Algorithm	Hyperparameter	Description
CBLOF	n_clusters	Number of clusters used in the clustering step (e.g., k-means).
	contamination	Proportion of data assumed to be outliers.
	use_weights	Whether to weight the outlier score by cluster size.
LOF	n_neighbors	Number of neighbors to compute local reachability density.
	contamination	Proportion of data expected to be outliers.
Isolation Forest	n_estimators	Number of base estimators (trees) in the ensemble.
	max_samples	Number of samples to train each tree. Proportion
	contamination	of outliers in the dataset.
DBSCAN	eps	Maximum distance between two samples to be considered neighbors.
	min_samples	Minimum number of samples in a neighborhood for a point to be a core point.
	metric	Distance metric used to calculate the distance between points.

We use the silhouette to find an optimal number of clusters for CBLOF, eps and MinPts for DBSCAN, which is used for each customer separately. It is introduced by Rousseeuw (1987), which measures how similar observation is to its own cluster compared to other clusters. It balances intracluster cohesion and intercluster separation and quantifies it from -1 (poor clustering) to +1 (well-clustered). To quantify this balance between intra-cluster cohesion and inter-cluster separation, the two different values that constitute the Silhouette Score. $a(i)$ which is the average distance from point x_i to all other points within the same cluster C_i is as follows:

$$a(i) = \frac{1}{|C_i| - 1} \sum_{\substack{x_j \in C_i \\ j \neq i}} d(x_i, x_j) \tag{4.18}$$

where:

- $d(x_i, x_j)$ is the distance from point x_i to all other points within the same cluster C_i
- $|C_i|$ is the size of the cluster.

$b(i)$ which is the minimum average distance from point x_i to all points in the nearest neighboring cluster is as follows:

$$b(i) = \min_{C_k \neq C_i} \frac{1}{|C_k|} \sum_{x_j \in C_k} d(x_i, x_j) \tag{4.19}$$

- $d(x_i, x_j)$ is the distance (typically Euclidean) between points x_i and x_j, where x_j is a member of nearest neighboring cluster.

The Silhouette Score for the entire dataset is then calculated as the average of individual scores:

$$S = \frac{1}{n} \sum_{i=1}^{n} s(i) \tag{4.20}$$

where

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}} \tag{4.21}$$

n is the total number of data points. A silhouette score close to 1 indicates well-separated and cohesive clusters, while scores near 0 or negative values suggest overlapping or incor-rectly assigned clusters. A higher average silhouette score indicates a more meaningful and well-separated clustering structure. In CBLOF, K-Means is used as the clustering algorithm. Therefore, while determining the near-optimal number of clusters, we evalu-ate cluster quality using K-Means and compute the Silhouette Score based on its results. Algorithm **??** presents the proposed method in CBLOF, while Algorithm 2 presents it for DBSCAN in greater detail. We use the unique user id dataset and past transactions dataset. The Algorithm is looped for each customer for its selected features. In Algorithm 1, for each customer, we assign a minimum number of clusters and a maximum number of clusters. From minimum to maximum cluster number, we fit K-Means for each cluster number, and the silhouette score is calculated for each iteration. The cluster with the best silhouette score is selected as a number of clusters. Selected features are *orig-inal price*, *coupon discount ratio*, *promotion discount ratio*, *numeric size*, *order period*, *brand popularity* which is also used for main anomaly detection algorithm as well which will be mentioned in the later. Those features are transactional variables that vary from transaction to transaction. It is crucial to apply anomaly detection techniques to the trans-actional dataset where variables are only selected transactional variables. The Algorithm

2 also gets a unique user dataset and past transactions and uses the same variables as well. We assume is varied from 0.1 to 5.0 with a step size of 0.2, while min samples are tested across integer values from 3 to 10. The DBSCAN algorithm is fitted to the customer's transactions for each parameter combination, and the clustering quality is assessed using the silhouette score. The parameter pair yielding the highest silhouette score is selected as the optimal configuration for that customer. These best-fit values of eps and min samples are then recorded in a user-level parameter table for subsequent use in anomaly detection.

It is also critical to determine an optimal number of neighbours (k) for the Local Outlier Factor (LOF) Algorithm to enhance the reliability of anomaly detection. To find the near-optimal number of neighbours for LOF, which is used for each customer separately, we use (k) finder based on the neighbourhood consistency method (KFC) to find out. KFC method is introduced by Yang et al. (2023), which is based on the neighbourhood consistency method. For each observation x_i, Φ_i^k denotes its k nearest neighbors and $\Omega_i^k = \Phi_i^k \cup \{x_i\}$ represents the local region. Neighbourhood consistency assumes that the outlier score s_i of x_i should be close to the average outlier score of its neighbours, which is:

$$v_i^k = s_i, \quad u_i^k = \frac{1}{k} \sum_{x_j \in \Phi_i^k} s_j \tag{4.22}$$

The global consistency across all samples is then evaluated using cosine similarity:

$$c_k = 1 - \cos(\mathbf{V}^k, \mathbf{U}^k)$$

where $\mathbf{V}^k = [v_1^k, \ldots, v_n^k]$ and $\mathbf{U}^k = [u_1^k, \ldots, u_n^k]$ are vectors of individual and neighborhood-average outlier scores, respectively. A higher c_k indicates stronger consistency and implies a more appropriate choice of k for the detector. For any candidate, a set of candidate values of c_k is evaluated, and one that maximizes neighbourhood consistency is selected. Algorithm 3 demonstrates it in detail. The algorithm is looped for each customer for its selected features. In Algorithm 3, we give past transactions and unique user dataset. For each customer, the algorithm iterates through a range of neighbourhood sizes, from a minimum of 5 to a maximum of 50. LOF scores are computed for each value of k, and a neighbourhood consistency score c_k is calculated. This score is derived from the cosine similarity between the LOF scores of each transaction and the average LOF scores of their respective k nearest neighbours. The value of k that maximizes this consistency score is

selected as the optimal neighbourhood size for that customer.

In the Isolation Forest, on the other hand, there is no literature-base n_estimator optimization. Since applying grid search to our algorithm is time-consuming, we accept n_estimator as 100 as F. T. Liu et al. (2008) justified.

Contamination refers to the proportion of data that are assumed to be anomalous within a given dataset. It is a critical hyperparameter for distinguishing normal observations from outliers. This parameter plays a significant role in identifying anomalous points for unsupervised methods such as Isolation Forest, Local Outlier Factor (LOF), and CBLOF (Clustering-Based Local Outlier Factor). An inaccurate contamination estimation can lead to excessive falsely flagged normal points or missed outliers. Hence, it affects the overall detection performance. We use an algorithm within our main anomaly detection algorithms to find the best contamination for each anomaly detection technique. We propose a heuristic for contamination tuning that selects the contamination rate, which maximizes the mean gap between predicted anomaly scores of the given anomaly detection method for normal points and outlier points. The algorithm can be seen in a detailed way in Algorithm 4.

Algorithm 1 best_n_clusters_cblof

Input: $past\_transactions$ // full dataset
$user\_df$ // dataframe to store user-specific best cluster number
$min\_clusters$, $max\_clusters$ // range of clusters to try

Output: $user\_df$ updated with best cluster number for each user

Initialize customer index $i \leftarrow 1$
foreach $user \in past\_transactions[user\_id\_hashed].unique()$ **do**
    Extract $user\_data$ from $past\_transactions$
    Select features: original price, coupon discount ratio, promotion discount ratio, numeric size, order_period, brand popularity
    Standardize features → X_scaled
    Determine $n\_samples \leftarrow$ number of rows in X_scaled
    Adjust $max\_clusters \leftarrow \min(n\_samples - 1,\ n\_samples//2,\ max\_clusters)$
    Initialize $best\_score \leftarrow -1$, $best\_k \leftarrow min\_clusters$
    foreach k *from* $min\_clusters$ *to* $max\_clusters$ **do**
        Fit K-Means with k clusters on X_scaled Compute silhouette score **if** *score* > $best\_score$ **then**
            Update $best\_score \leftarrow$ score Update $best\_k \leftarrow k$
        Continue on error
    Print "i-th customer is OK"
    Update $user\_df$ with $best\_k$ for the current user
    $i \leftarrow i + 1$
return $user\_df$

Algorithm 2 best_dbscan_params

Input: $past\_transactions$ // full dataset
$user\_df$ // dataframe to store user-specific parameters
min_eps, max_eps, eps_step // range for DBSCAN ε
$min\_samples\_range$ // range of min samples for DBSCAN

Output: $user\_df$ updated with best ε and min_samples

Initialize customer index $i \leftarrow 1$
foreach $user \in past\_transactions[user\_id\_hashed].unique()$ **do**
    Extract $user\_data$ from $past\_transactions$
    Select features: original_price, coupon_discount_ratio, promotion_discount_ratio, numeric_size, order_period, brand_popularity
    Standardize features → X_scaled
    if X_scaled *has fewer rows than* $min\_samples\_range[0]$ **then**
        continue
    Initialize $best\_score \leftarrow -1$, $best\_eps \leftarrow$ None, $best\_min\_samples \leftarrow$ None
    foreach $min\_samples$ *in* $min\_samples\_range$ **do**
        foreach ε *from* min_eps *to* max_eps *with step* eps_step **do**
            Fit DBSCAN with $(\varepsilon, min\_samples)$ on X_scaled Compute silhouette score
            if *score* > $best\_score$ **then**
                Update $best\_score$, $best\_eps$, $best\_min\_samples$
            Continue on error
    Print "i-th customer is OK"
    Update $user\_df$ with $best\_eps$ and $best\_min\_samples$ for the current user
    $i \leftarrow i + 1$
return $user\_df$

Algorithm 3 best_n_neighbours_for_lof

Input: $past\_transactions$ // full dataset
$user\_df$ // dataframe for output
$min\_neighbors$, $max\_neighbors$ // bounds for neighbor search

Output: DataFrame with best number of neighbors per user

Initialize empty list $best\_n\_neighbors\_list$ Set user counter $i \leftarrow 0$
foreach $user \in past\_transactions[user\_id\_hashed].unique()$ **do**
    Extract $user\_data$ from $past\_transactions$
    if *number of transactions* $\leq min\_neighbors$ **then**
        continue
    Select features: original price, coupon discount ratio, promotion discount ratio, numeric size, order_period, brand popularity
    Standardize features $\rightarrow X\_scaled$
    Set $local\_max\_neighbors \leftarrow \min(max\_neighbors, n\_samples - 1)$
    Initialize $best\_k \leftarrow min\_neighbors$, $best\_ck \leftarrow -\infty$
    foreach k *in* $[min\_neighbors, local\_max\_neighbors]$ **do**
        Fit LOF with k neighbors and obtain scores V_k
        Fit NearestNeighbors and obtain indices $knn\_indices$
        Compute U_k: average LOF score of each sample's k neighbors
        if *standard deviation of* V_k *or* U_k *is zero* **then**
            continue
        Compute consistency $c_k \leftarrow 1 - \text{cosine}(V_k, U_k)$
        if $c_k > best\_ck$ **then**
            Update $best\_ck \leftarrow c_k$ Update $best\_k \leftarrow k$
        continue on error
    Increment counter i Print "i-th user is OK"
    Append {user_id_hashed : $user$, best_n_neighbors : $best\_k$} to $best\_n\_neighbors\_list$
return *DataFrame from* $best\_n\_neighbors\_list$

Algorithm 4 get_best_contamination

Input: $scores$ // anomaly scores
$min\_cont, max\_cont, step$ // range and step for contamination

Output: $best\_cont$ // contamination value maximizing score gap

Initialize $best\_cont \leftarrow min\_cont$ Initialize $max\_gap \leftarrow 0$
for $cont = min\_cont$ **to** $max\_cont\ step$ **do**
    $threshold \leftarrow$ percentile of $scores$ at $100 \cdot (1 - cont)$ $inliers \leftarrow \{s \in scores \mid s \leq threshold\}$ $outliers \leftarrow \{s \in scores \mid s > threshold\}$
    if $|inliers| == 0$ ***or*** $|outliers| == 0$ **then**
        continue
    $gap \leftarrow mean(outliers) - mean(inliers)$
    if $gap > max\_gap$ **then**
        $max\_gap \leftarrow gap$ $best\_cont \leftarrow cont$
return $best\_cont$

4.4 Coupon Distribution

In the modelling part, we aim to calculate the probability of returns for each transaction with logistic regression variables and anomaly detection model outputs, which is also given to logistic regression as an anomaly flag. The coupon distribution aims to decrease those returns by offering coupons, which is expected to decrease the probability of return of that transaction. Therefore, retailers obtain higher revenue and lower environmental waste. We have finite budget B, and coupons offered to customers consume a portion of this budget. In the coupon distribution environment, we can only see the customer who wants to check out in time t; we cannot see the other customers coming after. Therefore, we make our decisions for each customer sequentially. Similarly, we do not know the features of the next coming customer. Consequently, whether to offer a coupon must be made in real-time, without knowledge of future opportunities, and is irreversible. The rewards associated with each decision are stochastic and customer-specific, depending on transaction features and estimated return probabilities with and without a coupon. The characteristics of our environment are similar to the class of Dynamic Stochastic Knapsack Problems (DSKPs), which extend the classical knapsack formulation for sequential item arrivals, probabilistic outcomes, and budget trade-offs. In this setting, the retailer should allocate the budget over the defined time period while balancing the benefit of reducing the return probability of

the current customer and the opportunity cost of detaining the budget that could be used for higher-impact customers. Assume $V_t(s)$ represents the optimal expected revenue from period t (each customer checkout is a new period) to the terminal period T, conditional on the remaining coupon budget $s \in [0, B]$ at time t. The corresponding Bellman equation is:

$$V_t(s) = \mathbb{E}_X \left[H_t(s; X) \right], \tag{5'}$$

where:

- $X\,t$ represents the observed selected features of the product and customer in period t
- $H_t(s; \chi_t)$ is the period-specific maximization function over the decision to offer a coupon or not

Assume that p^c (χ_t) is the probability of the return of the transaction at time t while p^o (χ_t) is probability of not returning the product. The optimal action at period t is determined by:

$$H_t(s; \chi_t) = \max \begin{Bmatrix} p_t^c(-\ell) + (1 - p_t^c)(\alpha_t r_t) + V_{t+1}(s - \theta r_t), \\ p_t^o(-\ell) + (1 - p_t^o)(\alpha_t r_t) + V_{t+1}(s) \end{Bmatrix}, \quad \text{for } s - \theta r_t \geq 0, \tag{6}$$

where:

- ℓ is the loss incurred if the item is returned
- r_t is the transaction revenue in period t
- α_t is the net profit margin factor
- θr_t is the coupon cost, proportional to revenue

and where boundary conditions are:

$$V_{T+1}(s) = s, \quad \forall s \geq 0. \tag{4.23}$$

Lastly, the optimal total profit over the T periods is given by

$$V^* = V_1(B). \tag{4.24}$$

In this setup, the decision to issue a coupon depends solely on the observed customer-product features and remaining budget, as every transaction opportunity is realized. According to Kleywegt & Papastavrou (2001), for each arriving item customer period t, there exists a dynamic threshold k^* (t, s, θr_t) such that it is optimal to accept the item (i.e., offer a coupon) if and only if:

$$s - \theta r_t \geq 0 \quad \text{and} \quad (p_t^o - p_t^c)(\alpha_t r_t + \ell) \geq k^*(t, s, \theta r_t) \tag{4.25}$$

This condition reflects the idea that a coupon should be offered only if the marginal benefit of influencing customer behaviour, which is measured by the change in return probability in our case, exceeds the marginal cost of using up the limited budget.

The threshold is given by:

$$k^*(t, s, \theta r_t) = \begin{cases} V_{t+1}(s) - V_{t+1}(s - \theta r_t), & \text{if } \theta r_t \leq s, \\ \infty, & \text{otherwise.} \end{cases}$$

This formulation defines k^* as the opportunity cost of spending θr_t units of budget in the current period. If the budget is insufficient, the opportunity cost is infinite, and a coupon is never offered.

Although the coupon distribution problem can be formulated as dynamic programming, solving it is computationally intractable for realistic problems. As first noted by Bell-man (1966), DP suffers from the curse of dimensionality, where the size of space grows exponentially with the number of state variables. In our setting, each possible budget level $s \in [0, B]$, at each time t, combined with feature-dependent probability distribu-tions, leads to an explosion in the number of states the DP must handle. Moreover, future customers, their return propensities, and their transaction values are unknown and only revealed sequentially. To solve the DP optimally, we need to compute the expected value over all possible future customer sequences and decision paths, which is computationally

hindered. For implementations with large customer populations and long decision hori-zons, which is our case, computing DP becomes computationally infeasible. As a result, approximate solutions such as threshold-based heuristics are used to achieve near-optimal results with reduced computational burden. We compute an optimization model to assign an upper bound to our heuristic models, in which we assume each customer is entered at the same time. The objective maximizes the expected net profit across all periods while penalizing returns and incorporating unused budget as a gain.

$$\max_{\Pi(X)} \quad \sum_{t=1}^{T} \Big[\Pi(X) \big(p^c(X)(-\ell) + (1 - p^c(X))(\tilde{\alpha}\tilde{r}) \big) + (1 - \Pi(X)) \big(p^o(X)(-\ell) + (1 - p^o(X))(\tilde{\alpha}\tilde{r}) \big) \Big] + B - \sum_{t=1}^{T} [\Pi(X) \cdot \theta \cdot \tilde{r}]$$

$$\text{subject to:} \quad \sum_{t=1}^{T} [\Pi(X) \cdot \theta \cdot \tilde{r}] \leq B$$

$$\Pi(X) \in \{0, 1\} \quad \forall X$$

Where;

- T: Total number of periods, which actually the number of customers
- X: Feature vector for each transaction which contains the variables we selected.
- $\Pi(X)$: Decision variable; 1 if a coupon is offered for transaction with features X, 0 otherwise
- $p^c(X)$: Estimated return probability with a coupon
- $p^o(X)$: Estimated return probability without a coupon
- $\tilde{r}$: Transaction revenue
- $\tilde{\alpha}$: Margin coefficient (portion of price retained as profit)
- ℓ: Penalty cost associated with a return

- θ: Coupon value as a fraction of price
- B: Total coupon budget

We propose two heuristics to solve DP model. Myopic policy is a policy that for a given transaction in state s and period t with the return probability estimates p_t^c and p_t^o ,we offer a coupon if and only if:

$$s - \theta r_t \geq 0 \quad \text{and} \quad \frac{(p_t^o - p_t^c)(\alpha_t r_t + \ell) - \theta r_t}{\theta r_t} \geq 0 \tag{4.26}$$

In the above formulation, if the transaction is profitable, we give coupon to the customer. However for our second approach, we assume that for a given transaction in period t with return probability estimated as $p^c(X)$ and $p^o(X)$ there exists a unique threshold k_{SP} such that it is optimal to offer a coupon if and only if:

$$s - \theta r_t \geq 0 \quad \text{and} \quad \frac{(p_t^o - p_t^c)(\alpha_t r_t + \ell) - \theta r_t}{\theta r_t} \geq k_{\text{SP}}. \tag{4.27}$$

where k_{SP} is:

$$k_{\text{SP}} = \begin{cases} 0, & \text{if } \tau(0) \leq \dfrac{B}{T\lambda} \\ \psi, & \text{otherwise} \end{cases} \tag{4.28}$$

where ψ is the unique solution of $\tau(\psi) = \frac{B}{T\lambda}$, and $\tau(\psi)$ is given with

$$\tau(\psi) = \mathbb{E}_{\boldsymbol{X}}\left[\theta\tilde{r}\mathcal{I}\left\{\frac{(p^o(\boldsymbol{X}) - p^c(\boldsymbol{X}))(\tilde{\alpha}\tilde{r} + \tilde{\ell}) - \theta\tilde{r}}{\theta\tilde{r}} \geq \psi\right\}\right], \tag{4.29}$$

where $\mathrm{I}\{.\}$ is the indicator function.

We estimate $k^*(t, s, \theta r_t)$ with the from the test data using Algorithm 7 in Appendix A. We refer to the forecasted threshold as $\hat{k}_{\text{SP}}$. For a given transaction in period t with return probability estimates p_t^c and p_t^o, offer a coupon (valued at θr_t) if and only if:

$$s - \theta r_t \geq 0 \quad \text{and} \quad \frac{(p_t^o - p_t^c)(\alpha_t r_t + \ell) - \theta r_t}{\theta r_t} \geq \hat{k}_{\text{SP}}. \tag{4.30}$$

We calculate the revenue with all three ways and compare how much we could improve to from myopic policy to optimum model revenue with using static policy.

4.5 Models Proposed

This section introduces five models whose output is given to logistic regression for anomaly detection. The final dataset columns are used as predictor variables except user_id_hashed. The dataset is divided into two parts: past transactions and new transactions. We select only customers with more than 20 transactions. Each customer's first 20 transactions are assumed to be past transactions, which we used to fit anomaly detection models. Therefore, we use this dataset to model customer behaviour with its past transactions. The rest of the transactions are new transactions, where we use anomaly detection models to predict those behaviours. From new transactions to test model performance, 60% of the new transactions are added to the past transactions and used as train data, whereas 40%

4.5.1 Baseline Model

While building the baseline model, we try different variables, both raw and feature-engineered ones, to determine whether those variables are meaningful to modelling customer behaviour. After several trials, the selected variables are in the final data. Logistic regression is trained using the training dataset, which is used to optimize the model's coefficients. The test dataset then evaluates the model's performance by comparing its predictions with the actual labels. Since the number of predictor variables is high, we also apply ridge regression to determine which of the variables remained in the model. We select different max_features parameters on the ridge regression, and the one selected in terms of both AUC, which is one of our evaluation criteria, is 15. Those selected variables are *promotion_discount_ratio*, *wallet_trx*, *membership_duration*, *gender_match*, *category_name_Cardigan*, *original_price promotion_discount_ratio*, *coupon_discount_ratio category_name_Sweatshirt*, *promotion_discount_ratio brand_popularity*, *elite_user category_name_TShirt*, *is_woman gender_match*, *is_woman brand_popularity*, *brand_popularity category_name_Sweater*, *coupon_id*, *promotion_name* and *promo x promo disc*

AUC is 0.8955. We accept this model as our baseline model. All the other anomaly detection outputs are added to this model.

4.5.2 Association Rule Mining

The underlying assumption of this technique is that if two products are frequently bought together and not returned, then in future transactions, the likelihood of return for one item decreases if it is purchased alongside another product from the established rule set. This perspective offers a behavioural resemblance to more reliable purchasing patterns. As a first step, we cluster both products and customers from the rule generation dataset. Products are clustered in terms of several variables, which are *original_price*, *avg_promotion*, *avg_coupon*, *return_ratio*, *numeric_size*, *brand_popularity* while customers are clustered with *avg_spending*, *average_discount_benefit_promo*, *average_discount_benefit_coupon* and *return_ratio*. The reason why we cluster the products is that since the association rule only accepts one variable to create rules for it, we decide to generate clusters and apply the rule mining to them. The rationale behind why we apply clustering on customers is that the rule generation data set is relatively smaller and creates more information since buying two or more products could not be the case for each customer. Therefore, by clustering the customer, we obtain more rules and more data to compute confidence and support. Algorithm 5 explains how we created the rules in a detailed way. As an output, we get the dataset with the related confidence.

Algorithm 5 Generate Train Dataset with Association Rule Confidence

Input: df, rule_generation

Output: df

foreach $i \in \{0, 1, 2, 3, 4\}$ **do**

    Filter rule_generation for non-returned and CustomerCluster i Group by user to get previous product clusters Convert to transactions and apply TransactionEncoder Generate frequent itemsets using Apriori Generate rules with single-item consequents Remove unused metrics (e.g., lift, leverage)

    Create identity rule matrix with confidence 1 for fallback Merge actual rules and fallback rules, group and aggregate confidence

    Filter train_df by CustomerCluster i Merge with previous product clusters by user Merge with rule confidences on antecedents and consequents

    if *confidence is missing* **then**

        Explode cluster set and try re-matching with singletons Fill missing confidences with 1

    Convert cluster sets to integers Append result to final dataset list

Concatenate all sub-results into train_df_last

return *df*

4.5.3 Other Anomaly Detection Techniques

The anomaly detection algorithms and hyperparameter optimization algorithms are applied for each customer. The contamination selection algorithm is used within the main anomaly detection algorithms. In best parameter selection algorithms and main anomaly detection algorithm, we use *original_price*, *coupon_discount_ratio*, *promotion_discount_ratio*, *numeric_size*, *order_period* and *brand_popularity* variables to cluster transactions of each customer since those variables are the ones that changes with respect to transaction. The anomaly detection algorithms have a similar core for each technique except for a rule-based model. The algorithms necessitate the unique user dataset with the best parameters that are used in each different anomaly detection technique (i.e. best number of clusters for the CBLOF and best eps and minPtc for DBSCAN) and the final dataset. The anomaly detection technique is fitted with the best parameters selected with the best parameter selection algorithms, and contamination is 0.5 selected to get anomaly scores. After obtaining the score, we use Algorithm 4 to find the best contamination. At that point, we obtain all the hyperparameters necessary to fit anomaly detection algorithms to assess whether a given

customer transaction is flagged as anomalous behaviour based on the selected transactional variables. Here, we have another parameter, which is the customer-specific return rate. If the customer return ratio of the products is higher than 50 %, then it is assumed that the customer is the abuser in buying. We assume that the normal behaviour of the customer is returning rather than keeping the product. Therefore, if the prediction that the anomaly detection technique produces is 1 for these customers, we assume it is zero and vice versa since our variable needs to measure anomalous behaviours. Since CBLOF is a clustering-based algorithm, we apply this parameter at the clustering level since abusing behaviours could also be contained in one cluster. For instance, in CBLOF, we know which point is assigned to which cluster. Here, we look at each cluster's return rate, and if that cluster's return rate is higher than 0.3, it means this cluster contains the abused behaviours. The reason why we cannot apply this in DBSCAN even though it is a clustered based algorithm is in cblof, all the outliers are also belong to a one cluster at first. On the other hand, DBCAN directly flags the outliers without assigning them to a cluster. Algorithm 6 demonstrates the anomaly detection algorithm in a more detailed way. An anomaly flag variable is added to the logistic regression model for each of the four models. We assume this prediction as *anomaly_flag* and add it to our final dataset.

Algorithm 6 apply_anomaly_detection_all (Compact Version)

Input: $past, new$ // past and new transactions
$user\_params$ // tuning parameters per user
$threshold$ // return rate threshold

Output: new with anomaly flags and scores

Initialize flags and scores in new
foreach $user \in past[user\_id\_hashed].unique()$ **do**
  Extract X_{past}, X_{new} and standardize **if** $|X_{past}| < 2$ *or* $|X_{new}| = 0$ **then**
    continue
  Compute $r_{user} \leftarrow$ past return rate
  CBLOF: Fit with k from $user\ params$, get scores/preds/labels Compare cluster return rate → flag anomalies
  LOF: Fit with k, tune contamination, predict on X_{new} **foreach** $(pred, idx)$ **do**
    if $(pred = 1 \wedge r_{user} < threshold) \vee (pred = 0 \wedge r_{user} \geq threshold)$ **then**
      Flag anomaly
  IForest: Fit with $cont = 0.5$, predict on X_{new} Flag if prediction and r_{user} mismatch
  DBSCAN: Fit with $(\varepsilon, minPts)$, convert labels Flag if label and r_{user} mismatch
return *new with flags and scores*

4.6 Evaluation Criteria

4.6.1 Area Under Curve

The Area Under the Receiver Operating Characteristic Curve (AUC-ROC) is a metric that is used to assess the discriminative power of binary classification models (**?**). It measures a model's ability to correctly distinguish between positive and negative classes. AUC values range from 0 to 1, with higher values indicating superior performance. Specifically, an AUC of 0.5 implies the model performs no better than random guessing, while values below 0.5 indicate worse-than-random performance. An AUC of 1.0 denotes a perfect classifier. In this study, AIC is used to compare logistic regression models which are baseline model and four logistic regressions with anomaly flag. The results of AUC of the six model we discussed is in the Table 4.2.

Compared to the baseline model, which achieved an AUC of 0.9023, all models incorporating anomaly-based features showed slight to moderate improvements in discriminative performance. Among them, the Logistic Regression model with LOF (Local Outlier Factor) flags achieved the highest AUC of 0.9138, suggesting the most substantial enhancement over the baseline, with an absolute gain of 0.0115. This implies that LOF-based anomaly signals contribute meaningfully to the model's ability to distinguish between classes. Isolation Forest and DBSCAN-based models also outperformed the baseline, achieving AUCs of 0.9094 and 0.9116 respectively, indicating consistent improvements with unsupervised anomaly detection techniques. In contrast, models augmented with CBLOF and Association Rule Mining showed only marginal improvements, with AUCs of 0.9030 and 0.9032 respectively—suggesting limited added value in those anomaly features for this specific classification task. Overall, these results demonstrate that integrating anomaly detection mechanisms, particularly those based on neighborhood density or isolation principles, can enhance model performance beyond traditional logistic regression.

Table 4.2: Comparison of Anomaly Detection Models by AUC

Model	**AUC**
Baseline Model	0.9023
CBLOF	0.9030
LOF	0.9138
Isolation Forest	0.9094
DBSCAN	0.9116
Association Rule Mining	0.9032

4.7 Coupon Distribution

In this section, our aim is to distribute coupons to decrease probability of return of the transactions. In the previous section, we analyze the AUC to find which one of the models we work on gives the better prediction power. LOF has better AUC compared to other methods, therefore, we select LOF to use for our prediction work. We also use baseline model to create a benchmark. The reason why we use baseline model as a benchmark is that we know that the baseline model predictive power is lower than the other but not that much. To assign it as a benchmark shows the prediction power of the anomaly detection models. The significant part of coupon distribution section is to demonstrate the heuristics

we create converge to optimal solution where everything is known and all of the customer show up at the environment at the same time.

We fit our test data to the model and we predict the probability of return for each transaction in the test dataset. However in this dataset, we see transactions that coupon has been already given. Therefore, we clone those transactions as if they have no return and predict their probability of return accordingly. For our work to distribute coupons, we assume that we distribute coupons that allows 15% discount. Therefore, we also predict that if each of those transactions have 15% , what would be their probability of return. We add those probabilities to our test dataset. We apply the same procedure to train dataset to use in Algorithm 1 in Appendix A to find $k*\ (t, s, \theta r_t)$ and apply static policy. Then we apply our coupon distribution techniques with the test dataset and return probabilities generated by LOF and Baseline Model . Table 4.3 demonstrates the revenues that created with LOF and Baseline across two coupon distribution heuristics and optimal solution.

Table 4.3: Revenue Comparison Across Coupon Budgets and Policies

Coupon (TL)	**Optimization Model**		**Myopic Policy**		**Static Policy**	
	Baseline	LOF	Baseline	LOF	Baseline	LOF
No Budget = No Coupon	4,239,596	4,223,439	4,239,596	4,223,439	4,239,596	4,223,439
10,000	4,278,266	4,266,207	4,258,530	4,243,977	4,277,822	4,242,643
20,000	4,308,869	4,302,491	4,277,778	4,264,826	4,308,685	4,295,647
30,000	4,336,280	4,335,037	4,297,176	4,285,407	4,329,186	4,331,934
40,000	4,361,613	4,363,862	4,316,719	4,306,421	4,351,061	4,347,401
50,000	4,385,179	4,389,200	4,335,046	4,327,137	4,374,617	4,371,866
60,000	4,407,209	4,412,192	4,355,065	4,348,694	4,386,582	4,395,173
70,000	4,427,541	4,433,311	4,375,626	4,371,284	4,408,551	4,400,690
80,000	4,446,131	4,452,879	4,394,175	4,392,004	4,413,939	4,423,943
90,000	4,463,124	4,471,255	4,412,648	4,412,578	4,434,522	4,437,339
100,000	4,478,753	4,488,522	4,431,384	4,432,921	4,444,435	4,449,787
110,000	4,493,154	4,504,729	4,450,283	4,453,924	4,456,809	4,463,138
120,000	4,506,589	4,519,943	4,468,902	4,475,412	4,470,987	4,477,597

The results in Tables 4.3 and 4.4 highlight the performance differences between the coupon distribution strategies and models. The optimization model, representing the upper benchmark where full information and perfect foresight are assumed, consistently achieves the highest revenue across all coupon budgets. Within each policy type, the LOF-based predictions outperform the Baseline model, demonstrating the benefit of incorporating anomaly detection into return probability estimation. For example, at a 120,000

Table 4.4: Incremental Revenue Gains under Optimization and Myopic Policies

Coupon (TL)	**Optimization Model**		**Myopic Policy**			
	Baseline	LOF	Baseline	%	LOF	%
10,000	38,670	42,767	18,934	48.96%	20,538	48.02%
20,000	69,273	79,051	38,182	55.12%	41,387	52.35%
30,000	96,684	111,598	57,580	59.55%	61,968	55.53%
40,000	122,018	140,422	77,124	63.21%	82,981	59.09%
50,000	145,584	165,760	95,451	65.56%	103,698	62.56%
60,000	167,613	188,752	115,469	68.89%	125,255	66.36%
70,000	187,945	209,872	136,030	72.38%	147,845	70.45%
80,000	206,536	229,440	154,579	74.84%	168,564	73.47%
90,000	223,528	247,816	173,053	77.42%	189,138	76.32%
100,000	239,157	265,083	191,788	80.19%	209,482	79.03%
110,000	253,558	281,289	210,687	83.09%	230,485	81.94%
120,000	266,994	296,503	229,306	85.88%	251,972	84.98%

TL coupon budget, the LOF-based optimization yields 296,503 TL in incremental revenue—surpassing the Baseline by nearly 30,000 TL. Even in the myopic and static heuristics, which operate under more limited decision-making frameworks, the LOF model leads to higher revenue than the Baseline, showing its practical utility even outside optimal conditions. These findings confirm that data-driven anomaly detection methods like LOF can significantly enhance coupon targeting, especially when aligned with intelligent policy design.

CHAPTER 5

CONCLUSION

This thesis presents an integrated approach to improving online retail profitability by combining anomaly detection techniques with dynamic coupon distribution strategies. The study is motivated by the challenge of high return rates in e-commerce, which not only decrease revenue but also increase operational costs. To address this, we developed a predictive models to estimate the probability of return for each transaction and proposed a heuristic-based policy to distribute discount coupons effectively.

Our anomaly detection analysis demonstrated that the Local Outlier Factor (LOF) model outperforms other methods in predicting return probabilities, as indicated by its superior AUC value. This finding justified the selection of LOF for downstream optimization tasks. Furthermore, by constructing a logistic regression baseline model, we were able to benchmark the added value of incorporating anomaly flags, ensuring that the improvement in prediction performance was both measurable and statistically significant.

The coupon distribution module operationalizes these predictions to reduce the expected cost of returns. By modeling a scenario in which 15% discount coupons are selectively offered, we simulated a real-world decision environment where customer transactions arrive

sequentially and decisions must be made in real time. Recognizing the limitations of having incomplete future information, we proposed heuristics that adaptively allocate coupons based on profitability thresholds. These heuristics were shown to closely approximate the optimal solution derived under the unrealistic assumption of full future information.

A key contribution of this work lies in demonstrating that machine learning-driven coupon targeting, when grounded in robust anomaly detection and guided by dynamic optimization, can meaningfully reduce product return rates without excessive promotional expenditure. The integration of customer behavior signals and anomaly indicators into return prediction models offers a scalable and practical pathway for enhancing post-purchase outcomes in online retail.

5.1 Societal Impact

Beyond the direct business implications, the proposed framework also carries meaningful societal impact. By reducing the number of product returns, the model indirectly contributes to lowering the carbon footprint associated with reverse logistics, including packaging waste, fuel consumption, and warehouse processing. Minimizing returns can also lead to more conscious consumer behavior, particularly if incentives are aligned to encourage deliberate purchasing decisions.

Moreover, the responsible use of data-driven techniques in coupon distribution promotes fairer and more efficient allocation of promotional resources. Rather than blanket marketing, personalized targeting ensures that discounts are directed to transactions where they generate the most value—economically and environmentally. This aligns with sustainable retail practices and supports the broader goal of reducing waste in the fast-growing e-commerce industry.

As the digital economy continues to expand, leveraging intelligent systems not only for profitability but also for sustainable operations represents an important step toward balancing commercial innovation with societal responsibility.

5.2 Further Research

While the proposed study gives promising results, there are several areas for potential improvement and future investigation. First, although the Local Outlier Factor (LOF) model provided better AUC scores than other anomaly detection methods, the overall AUC values indicate room for improvement in prediction accuracy. Enhancing feature engineering—such as incorporating more behavioral variables to the anomaly detection model since we could only add limited number of transactional variables -may increase model performance and demonstrate customer behavior more. Additionally, exploring more advanced anomaly detection techniques like deep autoencoders and Variational Autoencoders (VAEs) improve the detection return patterns.

Secondly, this work assumes a fixed coupon value (15% discount) and focuses solely on return probability reduction. Future studies could incorporate multi-objective optimization frameworks that simultaneously consider return probability, customer lifetime value, and retention probability. Moreover, adaptive coupon values based on predicted elasticity could be tested to improve cost-efficiency.

Lastly, a broader application across different product categories, customer segments, and longer time horizons would help assess the generalizability and robustness of the model. Cross-validation with external datasets and real-world A/B testing would provide further empirical grounding.

By addressing these avenues, future research can build upon this foundation to create even more effective, scalable, and socially responsible coupon distribution systems in online retail.

REFERENCES

Agrawal, R., Imieliński, T., & Swami, A. (1993). Mining association rules between sets of items in large databases. In *Proceedings of the 1993 acm sigmod international conference on management of data* (pp. 207–216). Association for Computing Machinery. Retrieved from https://dl.acm.org/doi/10.1145/170036.170072 doi: 10.1145/170036.170072

Aleskerov, E., Freisleben, B., & Rao, B. (1997). Cardwatch: A neural network based database mining system for credit card fraud detection. In *Proceedings of the ieee/iafe 1997 conference on computational intelligence for financial engineering (cifer)* (pp. 220–226). doi: 10.1109/CIFER.1997.618940

Allan, J., Carbonell, J., Doddington, G., Yamron, J., & Yang, Y. (1998). Topic detection and tracking pilot study: Final report. In *Darpa broadcast news transcription and understanding workshop.*

Author, F., Author, S., & Author, T. (2025). Analyzing professional ethics of physicians using online patient reviews. *Production and Operations Management*. Retrieved from https://journals.sagepub.com/doi/full/10.1177/10591478251318885 doi: 10.1177/10591478251318885

Barnett, V., & Lewis, T. (1984). *Outliers in statistical data* (2nd ed.). Chichester, UK: John Wiley & Sons.

Bechwati, N. N., & Siegal, W. S. (2005). The impact of the prechoice process on product returns. *Journal of Marketing Research*, *42*(3), 358–367. Retrieved from https://www.jstor.org/stable/30162379 doi: 10.1509/jmkr.42.3.358.17189

Bellman, R. (1952). On the theory of dynamic programming. *Proceedings of the National Academy of Sciences*, *38*(8), 716–719. doi: 10.1073/pnas.38.8.716

Bellman, R. (1966). Dynamic programming. *Science*, *153*(3731), 34–37. Retrieved from https://www.science.org/doi/10.1126/science.153.3731.34 doi: 10.1126/science.153.3731.34

Bernstein, F., Modaresi, S., & Sauré, D. (2019). A dynamic clustering approach to data-driven assortment personalization. *Management Science*, *65*(5), 2095–2115. Retrieved from https://papers.ssrn.com/sol3/papers.cfm?abstract id=2983207 doi: 10.1287/mnsc.2018.3077

Blattberg, R. C., & Neslin, S. A. (1990). Sales promotion: The long and the short of it. *Marketing Letters*, *1*, 81–97. Retrieved from https://doi.org/10.1007/BF00436151 doi: 10.1007/BF00436151

Bliss, C. (1934). The method of probits. *Science*, *79*(2037), 38–39. doi: 10.1126/science.79.2037.38

Bolton, R. J., & Hand, D. J. (2002). Statistical fraud detection: A review. *Statistical Science*, *17*(3), 235–255. doi: 10.1214/ss/1042727940

Breunig, M. M., Kriegel, H.-P., Ng, R. T., & Sander, J. (2000). Lof: Identifying density-based local outliers. In *Proceedings of the 2000 acm sigmod international conference on management of data* (pp. 93–104). ACM. Retrieved from https://dl.acm.org/doi/10.1145/335191.335388 doi: 10.1145/335191.335388

Cox, D. (1958). The regression analysis of binary sequences. *Journal of the Royal Statistical Society: Series B (Methodological)*, *20*(2), 215–242. Retrieved from https://www.scirp.org/reference/referencespapers?referenceid=1680101

Cramer, J. S. (2002). *The origins of logistic regression* (Tinbergen Institute Discussion Paper No. 02-119/4). Tinbergen Institute. Retrieved from https://papers.tinbergen.nl/02119.pdf

Cui, H., Rajagopalan, S., & Ward, A. R. (2020). Predicting product return volume using machine learning methods. *European Journal of Operational Research*, *281*(3), 612–627. doi: 10.1016/j.ejor.2019.05.046

Ester, M., Kriegel, H.-P., Sander, J., & Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the second international conference on knowledge discovery and data mining (kdd-96)* (pp. 226–231). AAAI Press. Retrieved from https://dl.acm.org/doi/10.5555/3001460.3001507

He, Z., Xu, X., & Deng, S. (2003). Discovering cluster-based local outliers. *Pattern Recognition Letters*, *24*(9-10), 1641–1650. doi: 10.1016/S0167-8655(03)00003-5

Hilafu, H., Letizia, P., & Roma, P. (2024). Product customization and returns: The moderating role of national culture. *Production and Operations Management*. Retrieved from https://journals.sagepub.com/doi/10.1177/10591478241249477 doi: 10.1177/10591478241249477

Hodge, V. J., & Austin, J. (2004). A survey of outlier detection methodologies. *Artificial Intelligence Review*, *22*(2), 85–126. Retrieved from https://www.researchgate.net/publication/220638052_A_Survey_of_Outlier_Detection_Methodologies doi: 10.1023/B:AIRE.0000045502.10941.a9

Hoerl, A. E., & Kennard, R. W. (1970). Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, *12*(1), 55–67. Retrieved from https://www.jstor.org/stable/1267351 doi: 10.2307/1267351

Janakiraman, N., Syrdal, H. A., & Freling, T. (2016). The effect of return policy leniency on consumer purchase and return decisions: A meta-analytic review. *Journal of Retailing*, *92*(2), 226–235. Retrieved from https://www.scirp.org/reference/referencespapers?referenceid=2925428 doi: 10.1016/j.jretai.2015.11.002

Kadiyala, B., Ö zer, , & Simsek, A. S. (2021). Data-driven approaches to targeting promotion e-mails: The case of delayed incentives. *Production and Operations Management*, *30*(3), 766–782. doi: 10.1111/poms.13316

Kahneman, D., & Tversky, A. (1979). Prospect theory: An analysis of decision under risk. *Econometrica*, *47*(2), 263–292. Retrieved from https://www.jstor.org/stable/1914185 doi: 10.2307/1914185

Kandula, S., Roy, D., & Akartunalı, K. (2024). A machine learning approach to solve the e-commerce box-sizing problem. *Production and Operations Management*, 1–20. Retrieved from https://journals.sagepub.com/doi/full/10.1177/10591478241282249 doi: 10.1177/10591478241282249

Kleywegt, A. J., & Papastavrou, J. D. (2001). The dynamic and stochastic knapsack problem with random sized items. *Operations Research*, *49*(1), 26–41. Retrieved from https://www.jstor.org/stable/222952 doi: 10.1287/opre.49.1.26.11130

Knorr, E. M., & Ng, R. T. (1997). A unified notion of outliers: Properties and computation. In *Proceedings of the third international conference on knowledge discovery and data mining (kdd-97)* (pp. 219–222). Retrieved from https://www.researchgate.net/publication/2717250 A_Unified_Notion_of_Outliers Properties and Computation

Kumar, V., & Sangwan, O. P. (2012). Development and assessment of intrusion detection system using machine learning algorithm. *International Journal of Computer Applications*, *ICNICT*(6), 33–36. Retrieved from https://www.ijcaonline.org/specialissues/icnict/number6/9455-1079/

Liu, F. T., Ting, K. M., & Zhou, Z.-H. (2008). Isolation forest. In *2008 eighth ieee international conference on data mining* (pp. 413–422). IEEE. Retrieved from https://ieeexplore.ieee.org/document/4781136 doi: 10.1109/ICDM.2008.17

Liu, S.-C., & Liao, Z.-W. (2001). Integrating innovation diffusion theory and the technology acceptance model: The adoption of electronic commerce by small and medium enterprises. *Information & Management*, *38*(8), 507–521. doi: 10.1016/S0378-7206(01)00127-6

Moon, J.-W., & Kim, Y.-G. (2001). Extending the tam for a world-wide-web context. *Information & Management*, *38*(4), 217–230. Retrieved from https://www.scirp.org/reference/referencespapers?referenceid=1592385 doi: 10.1016/S0378-7206(00)00061-6

Narasimhan, C. (1984). A price discrimination theory of coupons. *Marketing Science*, *3*(2), 128–147. Retrieved from https://www.jstor.org/stable/183747 doi: 10.1287/mksc.3.2.128

Narvar Team. (2023). *The growing normalization of returns in ecommerce.* https://corp.narvar.com/blog/returns-are-the-new-normal. (Accessed: 2025-05-18)

Patcha, A., & Park, J.-M. (2007). An overview of anomaly detection techniques: Existing solutions and latest technological trends. *Computer Networks*, *51*(12), 3448–3470. doi: 10.1016/j.comnet.2007.02.001

Peng, C.-Y. J., Lee, K. L., & Ingersoll, G. M. (2002). An introduction to logistic regression analysis and reporting. *Journal of Educational Research*, *96*(1), 3–14. Retrieved from https://doi.org/10.1080/00220670209598786 doi: 10.1080/00220670209598786

Perea y Monsuwé, T., Dellaert, B. G., & de Ruyter, K. (2004). What drives consumers to shop online? a literature review. *International Journal of Service Industry Management*, *15*(1), 102–121. doi: 10.1108/09564230410523358

Petersen, J. A., & Kumar, V. (2009). Are product returns a necessary evil? antecedents and consequences. *Journal of Marketing*, *73*(3), 35–51. doi: 10.1509/jmkg.73.3.35

Rakuten. (2025). *The right fit: How ai is changing ecommerce apparel returns.* https://www.retaildive.com/library/rakuten-whitepaper-the-right-fit/. (Accessed: 2025-05-18)

Richardson, M., Dominowska, E., & Ragno, R. (2007). Predicting clicks: estimating the click-through rate for new ads. In *Proceedings of the 16th international conference on world wide web* (pp. 521–530). Retrieved from https://dl.acm.org/doi/10.1145/1242572.1242643 doi: 10.1145/1242572.1242643

Robertson, T. S., Hamilton, R., & Jap, S. D. (2020). Many (un)happy returns? the changing nature of retail product returns and future research directions. *Journal of Retailing*, *96*(2), 172–177. Retrieved from https://bakerretail.wharton.upenn.edu/wp-content/uploads/2020/08/Many-Unhappy-Returns-PUBLISHED.pdf doi: 10.1016/j.jretai.2020.04.001

Rohm, A. J., & Swaminathan, V. (2004). A typology of online shoppers based on shopping motivations. *Journal of Business Research*, *57*(7), 748–757. doi: 10.1016/S0148-2963(02)00351-X

Rousseeuw, P. J. (1987). Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, *20*, 53–65. doi: 10.1016/0377-0427(87)90125-7

Saxena, A., Prasad, M., Gupta, A., Bharill, N., Patel, O. P., Tiwari, A., . . . Lin, C.-T. (2017). A review of clustering techniques and developments. *Neurocomputing*. doi: https://doi.org/10.1016/j.neucom.2017.06.053

Shang, G., Pekgün, P., Ferguson, M., & Galbreth, M. (2017). How much do online consumers really value free product returns? evidence from ebay. *Journal of Operations Management*, *53–56*, 45–62. doi: 10.1016/j.jom.2017.07.001

Sherry, J. F. (1983). Gift giving in anthropological perspective. *Journal of Consumer Research*, *10*(2), 157–168. Retrieved from https://www.jstor.org/stable/2488921

Srivastava, A., Kundu, A., Sural, S., & Majumdar, A. K. (2008). Credit card fraud detection using hidden markov model. *IEEE Transactions on Dependable and Secure Computing*, *5*(1), 37–48. doi: 10.1109/TDSC.2007.70228

Statista. (2023). *E-commerce product return rate in europe.* https://www.statista.com/chart/16615/e-commerce-product-return-rate-in-europe/. (Accessed: 2025-05-18)

Statista. (2024a). *ecommerce - worldwide.* Retrieved from https://www.statista.com/outlook/emo/ecommerce/worldwide (Accessed: 2025-05-18)

Statista. (2024b). *Share of e-commerce in total value of u.s. retail wholesale trade sales.* Retrieved 2025-05-18, from https://www.statista.com/statistics/185351/share-of-e-commerce-in-total-value-of-us-retail-wholesale-trade-sales/ (Accessed on May 18, 2025)

Statista. (2025). *E-commerce returns in the united states.* https://www.statista.com/topics/10716/e-commerce-returns-in-the-united-states/. (Accessed: 2025-05-18)

Su, M.-Y., Horng, S.-J., Fan, P., Lin, J.-L., Kao, T.-W., Chen, R.-J., ... Run, R.-S. (2011). A hybrid model based on hierarchical clustering and support vector machines for intrusion detection. *Expert Systems with Applications*, *38*(1), 306–313. doi: 10.1016/j.eswa.2010.06.066

Su, X. (2009). Consumer returns policies and supply chain performance. *Manufacturing & Service Operations Management*, *11*(4), 595–612. doi: 10.1287/msom.1080.0240

Toktay, B., van der Laan, E. A., & de Brito, M. P. (2004). Managing product returns: The role of forecasting. In R. Dekker, M. Fleischmann, K. Inderfurth, & L. N. van Wassenhove (Eds.), *Reverse logistics: Quantitative models for closed-loop supply chains* (pp. 15–45). Springer. doi: 10.1007/978-3-540-24803-3 3

Tsiliyannis, C. (2018). Markov chain modeling and forecasting of product returns in remanufacturing based on stock mean-age. *European Journal of Operational Research*, *271*(2), 545–558. doi: 10.1016/j.ejor.2018.05.026

Wang, Y., Anderson, J., Joo, S.-J., & Huscroft, J. R. (2020). The leniency of return policy and consumers' repurchase intention in online retailing. *Industrial Management & Data Systems*, *120*(1), 21–39. doi: 10.1108/IMDS-01-2019-0016

Yang, J., Tan, X., & Rahardja, S. (2023). Outlier detection: How to select k for k-nearest-neighbors-based outlier detectors. *Pattern Recognition Letters*, *174*, 112–117. Retrieved from https://doi.org/10.1016/j.patrec.2023.08.020 doi: 10.1016/j.patrec.2023.08.020

Zhang, J., & Wedel, M. (2009). The effectiveness of customized promotions in online and offline stores. *Journal of Marketing Research*, *46*(2), 190–206. doi: 10.1509/jmkr.46.2.190

Zhou, L., Xie, J., Gu, X., Lin, Y., Ieromonachou, P., & Zhang, X. (2016). Forecasting return of used products for remanufacturing using graphical evaluation and review technique (gert). *International Journal of Production Economics*, *181*, 315–324. Retrieved from https://doi.org/10.1016/j.ijpe.2016.04.016 doi: 10.1016/j.ijpe.2016.04.016

APPENDICES

Appendix A: Estimation of k^{SP}

Here, we outline the estimation procedure for k^{SP}, in (4.30). First note that $\tau(\psi)$ in (4.29) could be rewritten as:

$$\tau(\psi) = \mathbb{E}_{\boldsymbol{X}}\left[\theta\tilde{r}\mathcal{I}\left\{(p^o(\boldsymbol{X}) - p^c(\boldsymbol{X}))\left(\frac{\tilde{\alpha}}{\theta} + \frac{\ell_t}{\theta\tilde{r}}\right) - 1 \geq \psi\right\}\right]. \tag{5.1}$$

Our historical data consists of a sample of K transactions. For each transaction k, we have information on the price and profit margin of the purchased product, and the estimated return probabilities with and without the coupon for the customer. We will first provide an algorithm to estimate $\tau(\psi)$ in (5.1) for a given $\psi \geq 0$ and past transaction, $\tau(\psi)$ could be estimated as:

$$\hat{\tau}(\psi) = \frac{1}{K}\sum_{k=1}^{K}\theta r_k \mathcal{I}\left\{(p_k^o - p_k^c)\left(\frac{\alpha_k}{\theta} + \frac{\ell_t}{\theta r_k}\right) - 1 \geq \psi\right\}, \tag{5.2}$$

where $\mathrm{I}\{.\}$ is the indicator function.

Now we can present the algorithm for computing the threshold k^{SP}, which will be used for the upcoming T time periods. For a given B, T and the historical data, the threshold k^{SP} could be estimated with Figure 7.

Algorithm 7 Estimation of Threshold k^{SP}

Input: Budget B, time horizon T, historical data $\{(r_k, \alpha_k, p_k^c, p_k^o)\}_{k=1}^K$

Output: Estimated threshold k^{SP}

Estimate $\hat{\tau}(0)$ using equation 5.2 with historical data $\{(r_k, \alpha_k, p_k^c, p_k^o)\}_{k=1}^K$ **if** $\hat{\tau}(0) \leq \frac{B}{\Lambda(T)}$ **then**
    return 0
end
else
    Set $k_{\min} \leftarrow 0$ Set $\mathcal{E} \leftarrow 10^{-3}$ Set $k_{\max} \leftarrow \left(\frac{\Lambda(T)\ell_t}{B} + \frac{\max_k \alpha_k}{\theta}\right) \max_k (p_k^o - p_k^c) - 1$
    while *True* **do**
        Set $k \leftarrow \frac{k_{\min}+k_{\max}}{2}$ Estimate $\hat{\tau}(k)$ using equation 5.2
        if $\left|\hat{\tau}(k) - \frac{B}{\Lambda(T)}\right| \leq \mathcal{E}$ **then**
            return k
        end
        if $\hat{\tau}(k) < \frac{B}{\Lambda(T)}$ **then**
            $k_{\max} \leftarrow k$
        end
        else
            $k_{\min} \leftarrow k$
        end
    end
end