Integration of Machine Learning and Optimization Models for Data-Driven Decisions: Applications to Lot Sizing Problem with Random Yield

by

Bijan Bibak

A Dissertation Submitted to the
Graduate School of Sciences and Engineering
in Partial Fulfillment of the Requirements for
the Degree of

Doctor of Philosophy

in

Industrial Engineering and Operations Management

KOÇ ÜNİVERSİTESİ

January 13, 2025

Integration of Machine Learning and Optimization Models for Data-Driven Decisions: Applications to Lot Sizing Problem with Random Yield

Koç University

Graduate School of Sciences and Engineering

This is to certify that I have examined this copy of a doctoral dissertation by

Bijan Bibak

and have found that it is complete and satisfactory in all respects,
and that any and all revisions required by the final
examining committee have been made.

Committee Members:

_____

Prof. Fikri Karaesmen (Advisor)

_____

Prof. Barış Tan

_____

Assoc. Prof. Barış Yıldız

_____

Assoc. Prof. Hatice Tekiner Moğulkoç

_____

Prof. Mehmet Gönen

Date: _____

To my family.

ABSTRACT

Integration of Machine Learning and Optimization Models for Data-Driven Decisions: Applications to Lot Sizing Problem with Random Yield

Bijan Bibak

Doctor of Philosophy in Industrial Engineering and Operations Management

January 13, 2025

In recent years, significant advancements in data collection, analytical techniques, and data-driven methodologies have transformed the landscape of addressing complex decision-making and production planning challenges. Building on these innovations, this thesis introduces novel data-driven approaches to manage uncertainty and variability across diverse applications. The core focus of this research is to leverage big data in uncertain environments to derive effective decision-making rules for policymakers by integrating machine learning techniques with optimization methods. The thesis encompasses three primary studies: decision-making problems under uncertainty, a single-stage lot sizing problem, and a multi-period lot sizing problem with random yields.

First, we address a challenge in decision-making and classification problems where actions must be taken prior to observing an uncertain event, and incorrect decisions result in asymmetric costs. We propose some novel methods that integrate machine learning and optimization techniques to derive decision rules from limited observations, particularly in scenarios where frequent features have a significant impact on outcomes. Our methods aim to minimize the costs associated with incorrect decisions in both binary and multi class classification problems. Experimental results using publicly available data sets show that our approaches significantly outperform traditional benchmarks, reducing costs by up to 95% with respect to naive benchmarks.

Second, we investigate a data-driven lot sizing problem under random yield. Motivated by semi-conductor production, we focus on the case where the random

yield rate of a manufacturing process depends on a large number of features that can be observed before the lot sizing decision is made. Similarly, demand may also be random and may depend on a number of features. The lot sizing problem in this setting is challenging because the optimal decision depends on a large number of observed features for which there is limited data. To address this challenge, we propose estimation and optimization methods that combine tools from machine learning with tools from stochastic optimization. Using a publicly available data set for semi-conductor yield data and an additional synthetic data set, we compare the performance of different estimation and optimization approaches. We show that there is significant value of taking feature information into account for cost minimization. We also find that the best method for this problem combines tools from estimation with theoretical optimization properties of the random yield inventory problem.

Finally, we extend the single-period lot sizing problem to a multi-period setting. Given the realized variability in yields across periods, there are numerous potential scenarios by the end of the production horizon. The main objective is to develop a data-driven rule for determining optimal lot sizes under unknown yield distributions, unlike conventional approaches in the literature that assume known distributions. This rule aims to minimize the expected underage and overage costs at the end of the final period plus holding cost of intermediate periods and variable production cost while ensuring that total customer demand is met. This study advances the literature by introducing reinforcement learning to address the multi-period lot sizing problem with stochastic yields and unknown distribution for the first time. To assess the effectiveness of our approach, we compare results against established static and dynamic optimization methods. Our method provides a flexible and adaptive framework for decision-making in complex production systems characterized by stochastic yields.

ÖZETÇE

Veri Tabanlı Karar Alma için Bütünleşik Yapay Öğrenme ve Eniyileme Modelleri: Olasılıksal Üretim Randımanı Durumunda Lot Belirleme Problemi Uygulamaları

Bijan Bibak

Endüstri Mühendisliği ve İşletme Yönetimi, Doktora

13 Ocak 2025

Son yıllarda, veri toplama, analitik teknikler ve veri odaklı yöntemlerdeki önemli ilerlemeler, karmaşık karar alma ve üretim planlama zorluklarını ele alma biçimlerini dönüştürdü. Bu tez, bu yeniliklere dayanarak, çeşitli uygulamalarda belirsizliği ve değişkenliği yönetmek için yeni yaklaşımlar sunmaktadır. Bu araştırmanın temel odağı, makine öğrenimi tekniklerini, optimizasyon yöntemleriyle entegre ederek, belirsiz ortamlarda büyük verileri kullanarak karar vericiler için etkin karar alma kuralları türetmektir. Tez, üç temel çalışmayı kapsamaktadır: belirsizlik altındaki karar alma durumları, rassal randımanı olan üretim sistemleri için tek dönemli lot (parti) boyutlandırma ve son olarak çok dönemli lot boyutlandırma sorunu.

İlk olarak, belirsiz bir olayı gözlemlemeden önce eylemlerle ilgili kararların verilmesi gereken ve yanlış kararların asimetrik maliyetlerle sonuçlandığı karar alma ve sınıflandırma problemi incelenmektedir. Özellikle sık görülen özelliklerin sonuçlar üzerinde önemli bir etkiye sahip olduğu senaryolarda, sınırlı gözlemlerden karar kuralları türetmek için makine öğrenimi ve optimizasyon tekniklerini entegre eden dört yeni yöntem önermekteyiz. Yöntemlerimiz, hem ikili hem de çok sınıflı sorunlarda yanlış kararlarla ilişkili maliyetleri en aza indirmeyi amaçlamaktadır. Kamuya açık veri kümelerini kullanan deneysel sonuçlar, yaklaşımlarımızın geleneksel kıyaslama ölçütlerinden önemli ölçüde daha iyi performans gösterdiğini ve maliyetleri %95'e kadar azalttığını göstermektedir.

İkinci olarak, rastgele verim (üretim randımanı) altında veri odaklı bir lot boyutlandırma sorununu araştırıyoruz. Yarı iletken üretimindeki randıman sorunlarında yola çıkarak, bir üretim sürecinin rastgele verim oranının lot boyutlandırma kararı verilmeden önce gözlemlenebilen çok sayıda özniteliğe (feature) bağlı olduğu duruma

odaklanıyoruz. Benzer şekilde, talep de rastgele olabilir ve bir dizi özniteliğe bağlı olabilir. Bu ortamda en iyi lot belirleme kararı, randıman oranının sınırlı veri bulunan çok sayıda gözlemlenen özniteliğe bağlı olması nedeniyle zorlaşmaktadır. Bu zorluğun üstesinden gelmek için, makine öğreniminden araçları rassal optimizasyon yöntemleriyle birleştiren tahmin ve optimizasyon yöntemleri önerilmektedir. Yarı iletken verim verileri için kamuya açık bir veri kümesi ve ek bir sentetik veri kümesi kullanarak, farklı tahmin ve optimizasyon yaklaşımlarının performans karşılaştırması sunulmaktadır. Maliyet minimizasyonu için öznitelik bilgilerinin dikkate alınmasının önemli bir değeri olduğunu gösteriyoruz. Ayrıca, bu sorun için en etkin yöntemin, tahmin araçlarını rassal getirili envanter sorununun kuramsal optimizasyon özellikleriyle birleştirmek olduğunu göstermekteyiz.

Son olarak, randıman belirsizliği olan tek dönemli lot boyutlandırma problemini, çok dönemli duruma genişletiyoruz. Dönemler arasındaki randıman oranındaki değişkenlik göz önüne alındığında, üretim ufkunun sonunda çok sayıda olası senaryo bulunmaktadır. Ana amaç, literatürdeki bilinen olasılık dağıtımları varsayan geleneksel yaklaşımların aksine, bilinmeyen getiri dağıtımları altında optimum lot boyutlarını belirlemek için veri odaklı yöntemler geliştirmektir. Bu yöntemler, son dönemin sonunda beklenen eksik ve fazla maliyetleri artı ara dönemlerin stok tutma ve değişken üretim maliyetini en aza indirmeyi ve toplam müşteri talebinin karşılanmasını sağlamayı amaçlamaktadır. Bu çalışma, rassal randıman oranı ve bilinmeyen olasılık dağılımları ile çok dönemli lot boyutlandırma sorununu ilk kez ele almak için pekiştirmeli öğrenme yöntemlerini de önermekte ve sınamaktadır. Yaklaşımımızın etkinliğini değerlendirmek için sonuçları yerleşik statik ve dinamik optimizasyon yöntemleriyle karşılaştırıyoruz. Yöntemimiz, rassal randıman oranlarına sahip üretim sistemlerinde karar alma için esnek ve uyarlanabilir bir çerçeve sunmaktadır.

ACKNOWLEDGMENTS

First, I would like to express my heartfelt gratitude to my advisor, Prof. Fikri Karaesmen (Fikri Hocam), for his unwavering support, guidance, motivation, and immense knowledge throughout my Ph.D. studies and research. His expertise and mentorship have been instrumental in shaping my academic journey and guiding me toward my career aspirations. Without his dedicated involvement at every stage, this thesis would not have been possible. I am profoundly grateful for the trust and confidence he placed in my abilities.

I am also deeply thankful to my committee members for their insightful comments, encouragement, and active participation. I extend my gratitude to Assoc. Prof. Hatice Tekiner Moğulkoç, my advisor during my M.Sc. studies, whose continued support and collaboration have greatly enriched my research over the years. I am especially appreciative of Prof. Barış Tan, the president of Özyeğin University—it was an honor to join his AutoTwin project and benefit from his invaluable insights and guidance. My sincere thanks go to Assoc. Prof. Barış Yıldız for all his support and valuable comments, which have significantly improved the quality of my thesis. Finally, I extend my special thanks to Prof. Mehmet Gönen for his steadfast support and constructive feedback over the years, which have greatly impacted the validity of my research and enhanced the quality of my thesis.

I am truly grateful to my friends for their unwavering support and encouragement throughout my academic journey. My deepest appreciation goes to Amirreza, Ada, Ali^2, Arash, Ayşe Beyza, Ayşe, Damla, Duygu, Eda Nur, Fariborz, Fatemeh, Hadi, Hanife Kübra, Kazem, Mehmet Akif, Mina, Parinaz, Reza, and Zeynep (sorted alphabetically :D), as well as all my other friends, who continually inspired me to strive for excellence and to grow both personally and professionally. I carry

fond memories of Koç University in my heart, thanks to the wonderful friends and exceptional professors who made my time there truly unforgettable.

Most importantly, none of this would have been possible without the unconditional love and support of my family. My deepest gratitude goes to my dear and loving father, mother, sister, brother, grandfather, grandmother, uncles, aunts, and cousins. Words cannot express how thankful I am for the sacrifices you have made on my behalf. This thesis stands as a testament to your unwavering love, encouragement, and belief in me. I love you so much.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

ANN	Artificial Neural Network
CART	Classification and Regression Trees
CDE	Conditional Distribution Estimation
ERM	Empirical Risk Minimization
GBR	Gradient Boosting Regression
KPIs	Key Performance Indicators
LP	Linear Programming
PE	Prediction Error
RF	Random Forest
RFECV	Recursive Feature Elimination with Cross-Validation
SAA	Sample Average Approximation
VIEO	Value of Integrating Estimation and Optimization

Chapter 1

INTRODUCTION

In a world increasingly driven by data, effective decision-making and resource management are more critical than ever. From businesses aiming to minimize costs to factories optimizing their production processes, the need for intelligent, data-driven solutions has never been greater. Similarly, in healthcare, data-driven methods play a pivotal role in improving outcomes, such as optimizing treatment plans for patients or accurately diagnosing diseases to ensure appropriate and timely interventions. However, real-world decision-making is rarely straightforward. It often involves uncertainty, trade-offs, and the challenge of balancing competing objectives. This thesis focuses on addressing such challenges through innovative methods that combine advanced modeling, optimization, and machine learning techniques.

In recent years, machine learning techniques have become essential for decision-making under uncertainty. Classification models categorize data into predefined classes based on observed features, assisting decision-makers in identifying patterns and predicting outcomes with varying confidence levels. When a model estimates the probability of an instance belonging to specific classes, it conveys its uncertainty regarding the correct label. The decision rule that maps these probabilities to final classifications (e.g., through thresholding) resembles decision-making under uncertainty, (Duda et al., 2001).

In many decision-making problems, the costs associated with different types of errors are not equal. In this example, a false negative (failing to identify a disease) might lead to severe consequences, while a false positive (incorrectly diagnosing a disease) could result in unnecessary treatment. Traditional classification models that aim to minimize overall error may not adequately address these cost disparities. The

impact of misclassification errors can be significant, particularly when false positives and false negatives carry different costs. As Viaene and Dedene (2004) discuss, a challenge in traditional classification for decision-making contexts is prioritizing cost minimization over mere error reduction.

This thesis focuses on data-driven approaches for decision-making, combining machine learning and optimization to derive decision rules that map elements of high-dimensional feature spaces $(X_1, X_2, \ldots, X_p)$ to an element j of action space $\mathcal{J} = \{1, 2, 3, \ldots, J\}$. Despite the challenge of limited observations and the uniqueness of each instance, our methodology efficiently addresses this complexity by considering a probabilistic framework. For each of the J actions, there exists a chance node with K possible probabilities $(p_k|\mathbf{X})$, where each probability is a function of the features. Our approach uses these probabilities and the cost functions $(C(j, k))$ to estimate an expected cost and consequently optimize decision-making under uncertainty. As illustrated in Figure 1.1, the interplay between data, uncertainty, and optimization highlights the key properties of our framework in extracting decision rules that can be effectively applied to for cases whose features were not observed in the training data.

Chapter 2 of this thesis investigates decision-making problems where actions must be taken prior to the realization of uncertain events, and the cost of incorrect decisions varies based on the type of error. For instance, in healthcare, misclassifying a critical condition as benign may lead to significantly more severe consequences than the opposite error. Similarly, an incorrect diagnosis of a serious illness could result in either unnecessary treatments or a missed opportunity to save a life. To address these challenges, this thesis introduces four novel data-driven methods that explicitly incorporate asymmetric costs. These methods leverage data to derive optimal decision rules for various scenarios and integrate machine learning with optimization techniques to provide interpretable and actionable guidance to decision-makers.

Recent years have seen an increase in data-based inventory optimization models where additional feature data that provides information about demand is taken

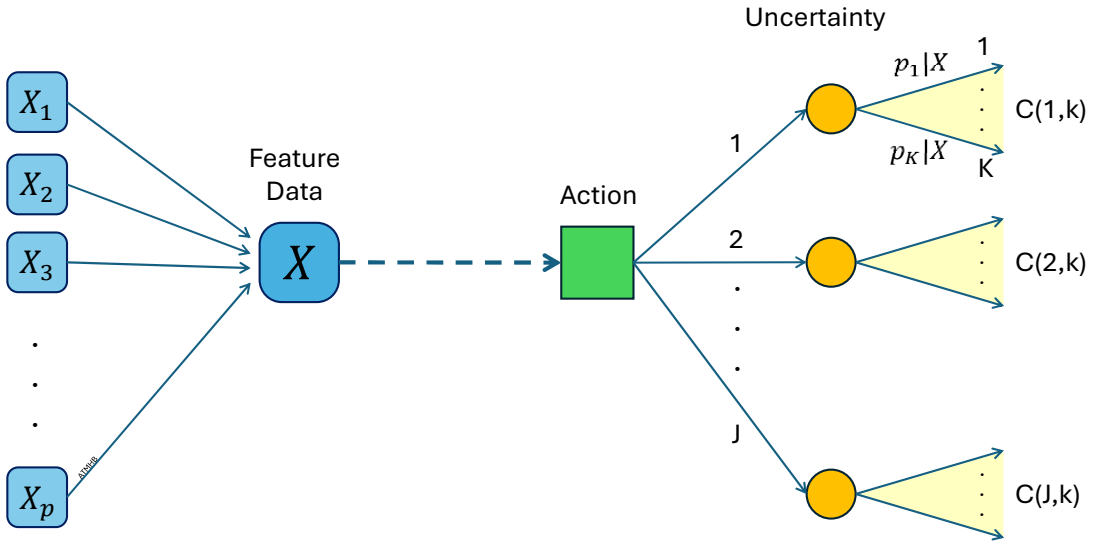

Figure 1.1: A simple structure of a data-driven decision making problem with many features

into account. For instance, Sachs (2015) considers price and temperature data in addition to demand observations and presents a model where the ordering quantity linearly depends on these two features. Ban and Rudin (2019) consider a more general model where there might be many features that might potentially influence demand and combine tools from statistical learning to address the prediction and optimization problem while addressing feature selection (model reduction) issues simultaneously. They show that significant gains can be made in inventory cost minimization when feature data is used effectively. Many other recent papers (for instance Van der Laan,Teunter, Romeijnders, and Kilic, 2020, Oroojloyjadid, Snyder and Takac, 2020, Neghhab, Khayyati and Karaesmen, 2023, Liu, Letchford, and Svetunkov, 2022, Notz and Pibernik, 2022 , Mandl et al., 2022 and Mandl and Minner, 2023) present methodology that supports the potential of data-based inventory optimization with feature information. Erkip (2023) presents some of the opportunities and new challenges presented by data availability on inventory problems.

Concepts and tools from statistical/machine learning are key factors when possibly big data (i.e., many features) is used for the prediction of random quantities that then serve as inputs to an operational problem. Bertsimas and Kallus (2020) refer to the framework of generating optimal operational decisions based on data including a large number of features as *prescriptive analytics* in contrast to *predictive analytics*, which is the main focus of machine learning methods. They propose a systematic approach for combining learning and stochastic optimization and present several examples.

Chapter 3 transitions to the domain of single-stage single-period lot sizing problems, a fundamental challenge in production and inventory management. In this setting, businesses must determine the optimal production or order quantities while dealing with uncertainties in both demand and production yields. Numerous features influence the realization of demand and yield, yet observational data is often limited. To address these challenges, in this chapter, we present four approaches designed to minimize the costs associated with overage and underage quantities. These methods yield actionable rules that learn from the observational data for identifying optimal lot sizes given a set of yield and demand features, striking a balance between efficiency and reliability.

Chapter 4 builds upon the previous chapter and addresses a more intricate scenario: the multi-period lot sizing problem with random yields. In this context, limitations in inventory and production capacities, combined with variability in yields, require distributing lot sizes across multiple periods. At each period, random yields influence both production output and subsequent inventory levels, creating a dynamic system where decisions made in earlier periods directly impact later periods. This complexity is further heightened by the uncertainty in yields at each period and the availability of limited observational data. To tackle these challenges, in this chapter, we introduce a novel reinforcement learning approach that determines optimal lot sizes at each period without requiring explicit knowledge of the underlying yield distributions. By combining flexibility and robustness, this method offers a practical and effective solution for real-world applications.

Together, these three chapters provide a cohesive framework for tackling diverse decision-making and optimization problems under uncertainty. By blending theory and practice, the methods developed in this thesis aim to empower decision-makers in various fields, from manufacturing to healthcare and beyond.

Chapter 2

A CLASS OF DATA-DRIVEN DECISION MAKING PROBLEM WITH ASYMMETRIC COSTS

2.1 Introduction

Decision making under uncertainty is a critical aspect of many real-world applications, ranging from medical diagnoses and financial investments to supply chain management and policy making. In such cases, decision makers often lack complete information or face unpredictable variables that can significantly impact the outcomes of their choices. The challenge lies in formulating strategies that can effectively manage these uncertainties while optimizing the desired outcomes. Traditional decision-making models often rely on probabilistic reasoning or robust optimization to navigate uncertainty, but these methods can fall short when dealing with complex or high-stakes decisions where the costs associated with wrong decisions are asymmetric.

In recent years, machine learning classification techniques have become essential for decision-making under uncertainty. While both fields aim to make optimal choices with incomplete information, they do so in distinct ways. Classification models categorize data into predefined classes based on observed features, assisting decision-makers in identifying patterns and predicting outcomes with varying confidence levels. When a model estimates the probability of an instance belonging to specific classes, it conveys its uncertainty regarding the correct label. The decision rule that maps these probabilities to final classifications (e.g., through thresholding) resembles decision-making under uncertainty (Duda et al., 2001).

In medical applications, for instance, classification models predict the presence or absence of disease from patient data, thereby guiding treatment choices. A key

issue in applying classification models to decision-making is that these models typically maximize accuracy or minimize classification errors, disregarding the costs associated with different error types. In many decision-making problems, the costs associated with different types of errors are not equal. In this example, a false negative (failing to identify a disease) might lead to severe consequences, while a false positive (incorrectly diagnosing a disease) could result in unnecessary treatment. Traditional classification models that aim to minimize overall error may not adequately address these cost disparities. The impact of misclassification errors can be significant, particularly when false positives and false negatives carry different costs. As Viaene and Dedene (2004) discuss, a challenge in traditional classification for decision-making contexts is prioritizing cost minimization over mere error reduction.

This is where cost-sensitive learning becomes essential. The introduced concept of cost-sensitive learning by Green and Swets (1966), Pearl (1988), Turney (1995), and Elkan (2001) has emerged as a prominent field that seeks to tackle the limitations of traditional classification by assigning different costs to different types of errors. Cost-sensitive learning, on the other hand, directly incorporates the varying costs of errors into the model training process. This approach allows for the development of decision-making models that are not only accurate but also aligned with the specific cost structures of the problem at hand, leading to more optimal and practical decisions. By integrating cost-sensitive learning into decision-making frameworks, we can better manage the inherent uncertainties and complexities of real-world problems, ultimately leading to more informed and effective decisions. However, most existing research in this domain have focused solely on binary class problems with imbalanced data sets. Additionally, current models often face limitations in areas such as scalability, real-time application, interpretability, and generalization.

This study advances the field by integrating the strengths of traditional classification techniques and cost-sensitive learning, while also addressing their limitations in managing asymmetric costs. To bridge this gap, we propose four novel approaches: Mixed-Integer Programming Model for Optimal Decision Tree (MIPODT), a Ran-

dom Forest of MIP Models (MIPRF), the Optimal Hyperplane Decision Tree (OHDT), and Separate Estimation and Optimization (SEO). These methods operate under the assumption that optimal decisions can be derived from decision rules extracted from data sets with limited observations, including corresponding features and labels (outcomes). We evaluate the performance of the proposed methods using publicly available real-world data sets, focusing on minimizing the expected costs associated with different types of errors (misclassifications). Our proposed method offers a novel, flexible, and effective approach to decision making under uncertainty, particularly in contexts where the economic implications of errors are not uniform. Through rigorous experimentation and analysis, we demonstrate the superior performance of our method in optimizing decision outcomes in scenarios characterized by asymmetric cost structures.

We can summarize the contributions of this thesis as follows:

- Introduces four novel optimization-based approaches for decision-making under uncertainty with asymmetric error costs.
- Effectively addresses both binary and multi-class problems, independent of data balance.
- Directly accounts for the real costs associated with different error types, rather than relying on tuning classifier thresholds.
- Provides decision-makers with efficient and actionable rules for future unseen events, improving decision-making in high-stakes environments.

This chapter is organized as follows: a literature review is presented in Section 2.2. The problem and the models including the data-driven case, are introduced in Section 2.3. The main key performance indicators (KPIs) are introduced in Section 2.4. Section 2.5 presents the numerical results of proposed methods. In addition, we assess and compare the performance of the proposed approaches. Finally, conclusions are presented in Section 2.6.

2.2 Literature Review

We present optimal solutions for decision-making problems under uncertainty, where decisions are influenced by various features. Our main objective is to introduce novel decision rules that facilitate decision-making problems while accounting for the concept of asymmetric costs associated with error types. In the decision making under uncertainty, there is a well-established literature going back to the pioneering work by Dantzig (1955) and Charnes and Cooper (1959), who laid the groundwork for stochastic programming and optimization under probabilistic constraints, respectively. Despite requiring different models and solution techniques, both fields share the fundamental assumption that the probability distributions of the random variables are precisely known. This assumption persists despite early warnings, such as Scarf (1958a)'s observation that future demand might unpredictably deviate from past distributions. Over the decades, most research has continued to rely on accurate knowledge of underlying probabilities, even though significant computational challenges arise, such as the need for multivariate integration to evaluate chance constraints and the large-scale nature of stochastic programming problems. Traditional methods often rely on assumptions about the probability distribution of uncertain parameters, which may not hold in real-world scenarios. For a comprehensive overview of solution techniques, refer to Birge and Louveaux (1997) and Kall and Mayer (2005).

In recent years, this area has expanded to include robust and stochastic optimization. Techniques such as robust optimization (Aharon Ben-Tal and Nemirovski, 2009) and scenario-based optimization (Rockafellar and Uryasev, 2000) have been developed to find optimal decisions that account for the variability in parameters. These methods often involve the creation of scenarios or distributions that represent the uncertain environment and optimize decisions across these scenarios. The foundational concepts of expected utility theory (Neumann and Morgenstern, 1944) and stochastic optimization (Aharon Ben-Tal and Nemirovski, 2009) have provided structured approaches to evaluating potential outcomes and making choices that maxi-

mize expected benefits or minimize risks. These methodologies have been crucial in environments characterized by varying degrees of uncertainty, where the decision-maker must account for the unknowns to make optimal decisions. A limitation of robust optimization is its tendency to be overly conservative, often resulting in sub-optimal solutions under typical scenarios. In contrast, stochastic optimization, while more flexible, demands extensive computational resources and is highly sensitive to the quality of the underlying probabilistic models.

Advancements in supervised machine learning have enhanced decision-making tools. While traditional methods like logistic regression (Cox, 1958) and decision trees (Breiman et al., 1984a) are widely used, advanced techniques such as support vector machines (Cortes and Vapnik, 1995) and random forests (Breiman, 2001) improve accuracy and generalization.

The Classification and Regression Trees (CART) framework introduced by Breiman et al. (1984b) uses a top-down strategy to partition data by maximizing homogeneity at each split, with terminal nodes representing outcomes (Loh, 2011; Romano et al., 2014). Despite CART's interpretability, it faces limitations in computational efficiency and optimal tree construction. To address these, Bertsimas and Dunn (2017) proposed Optimal Classification Trees (OCT), leveraging mixed-integer optimization to globally construct trees, enhancing predictive performance on complex data sets while maintaining interpretability. Various heuristic approaches for constructing optimal univariate decision trees have been explored, including linear optimization (Bennett, 1992), continuous optimization (Bennett and Blue, 1996), dynamic programming (Cox, 1989), genetic algorithms (Son, 1998), and stochastic gradient descent (Norouzi et al., 2015); however, none have produced certifiably optimal trees within practical time frames.

While most classification algorithms optimize expected outcomes, they typically assume symmetric error costs. However, in many real-world scenarios, such as medical diagnosis or fraud detection, the costs associated with different errors are often asymmetric. This necessitates more refined approaches, such as cost-sensitive learning, which explicitly addresses these asymmetric costs. The following literature

review examines key advancements in this area, emphasizing how they improve classification methods to better align with decision-making objectives under uncertainty and varying error costs.

Cost-sensitive learning originated from decision-making theories under uncertainty, with early insights from Green and Swets (1966) on weighting error types and Pearl (1988) on incorporating costs into Bayesian networks. These foundational works lacked practical machine learning algorithms for complex, multi-class scenarios. The application in machine learning began with Turney (1995), who evaluated cost-sensitive decision tree algorithms for binary classification. Elkan (2001) advanced the field by adjusting decision thresholds based on cost matrices, influencing methods like Zadrozny and Elkan (2001), though adaptation to multi-class settings and deep learning models remained unaddressed. To handle multi-class cost-sensitive problems, Naoki Abe and Langford (2004) introduced an iterative reweighting algorithm, and Zadrozny and Langford (2005) adapted existing algorithms by reweighting training examples. These approaches increased accessibility but were limited in handling imbalanced data or deep learning contexts.

Addressing class imbalance, Liu and Zhou (2006a) transformed multi-class problems into binary ones, and Liu and Zhou (2006b) integrated cost-sensitive learning into neural networks for imbalanced data. In deep learning, Keze Wang and Lin (2017) proposed the CEAL framework to minimize labeling costs, while Salman H. Khan and Togneri (2018) introduced a class-balanced loss function and feature-level cost-sensitivity to improve performance on imbalanced data sets.

To summarize, traditional models often rely on probabilistic risk assessment without adequately considering varying misclassification costs, leading to suboptimal real-world decisions. Many classification studies aim to maximize overall accuracy, neglecting the critical impact of misclassification costs, especially when some errors are significantly more costly than others. While cost-sensitive learning addresses some of these limitations, challenges like scalability, real-time application, interpretability, and generalization remain. We contribute by introducing four novel optimization-based approaches effective for both binary and multi-class problems,

regardless of data balance. Unlike methods that adjust classifier thresholds, our approaches directly account for the actual costs of error types, providing decision-makers with efficient and actionable rules for future unseen events.

2.3 The Problem and the Models

In this study, we focus on a class of decision-making problems characterized by an action (decision) prior to the observation of an uncertain event leading to finite discrete outcomes. The available actions are both discrete and finite, and the level of uncertainty remains unaltered by the choice of action. The objective of this study is to develop efficient rules for decision making in scenarios where unseen events are influenced by various features, and incorrect decisions incur asymmetric costs.

Let us consider the following setup, there is an uncertain event whose outcome k takes one of K possible values ($k = 1, 2, ..., K$). There are J discrete actions available. We assume that the actions do not affect the outcome of the uncertain event but each action generates a possibly different cost for each outcome. The estimated cost for each action j, considering all possible outcomes, is denoted as $\hat{c}(j)$. A combination of action j ($j = 1, 2, ..., J$) and outcome k generates a cost of c_{jk}. We assume that N past observations are available, and for each observation i ($i = 1, 2, ..., N$) a corresponding outcome k is recorded. A binary parameter w_{ik} denotes, whether observation i has an outcome of k or not. ($w_{ik} = 1$ if observation i is assigned to outcome k, and $w_{ik} = 0$ otherwise). Let $\mathcal{J} \equiv \{1, 2, ..., J\}$, finding the optimal action that minimizes the total cost for the data set in this setup corresponds to the solution of:

$$j^* = \operatorname{argmin} \mathbb{E}\left[\hat{c}(j)\right] \equiv \underset{j \in \mathcal{J}}{\operatorname{argmin}} \sum_i \sum_k c_{jk} w_{ik} \tag{2.1}$$

The most challenging version of the problem arises when, in addition to the outcomes and corresponding assigned decisions, we have access to a set of features that influence both the decision j and the outcome k. In this setup, consider a training data set (X, Y) consisting of n observations (X_i, y_i). Each observation

consists of $X_i \in \mathbb{R}^p$ represents a vector of p features and y_i which represents possible outcome k. Despite the high dimensionality of the feature space, the number of observations remains relatively small. For each observation i, a decision j must be made, which depends on its corresponding feature set X_i and is denoted as $j|X_i$. Similarly, the outcome k depends on the feature set X_i and is denoted as $k|X_i$. In this framework, the objective is to derive an optimal decision rule $j^*|X$ from the training set that minimizes the total cost across all observations, as expressed in the following formulation. This decision rule is then applied to unseen instances (test set), where decisions must be made based on the available feature set X' before the outcomes are realized.

$$j^*(\mathbf{X}) = \underset{j \in \mathcal{J}}{\operatorname{argmin}} \, \mathbb{E}\left[\hat{c}(j) \mid \mathbf{X}\right] \tag{2.2}$$

The above problem is challenging because one needs to find a mapping from the multi-dimensional feature vector $\mathbf{X}$ to the action space $\{1, 2, ..., J\}$ using some limited number of observations. Machine learning proposes some classes of mappings such as recursive binary trees or logistic regressions along with optimization methods to obtain a near-optimal mapping within a given class for prediction problems. We extend this framework to propose solutions to the case of finding rules that yield near optimal actions.

To clarify this framework, consider an example involving N observations of breast cancer patients. Each observation represents the comprehensive information documented for a patient, consisting of p features derived from various sources such as MRI scans, tests, and other diagnostic tools. Additionally, we have access to a binary outcome k, where $k = 0$ indicates the absence of cancer, and $k = 1$ indicates the presence of cancer. As a doctor, the decision to be made involves selecting the appropriate treatment type. In this context, $j = 0$ corresponds to late treatment (or no treatment), while $j = 1$ corresponds to early treatment. The objective is to assign the correct decision j^* to each outcome, ensuring that the decision minimizes the associated costs of incorrect treatment choices.

Consider the following scenarios:

- For w_{00}, when $Y = 0$ (i.e., the patient has no cancer) and $Z = 0$ (i.e., we start late treatment, or no treatment), the decision is correct, and the cost associated with this decision is negligible.

- For w_{11}, when $Y = 1$ (i.e., the patient has cancer) and $Z = 1$ (i.e., we start early treatment), the decision is correct, and the cost associated with this decision is negligible.

- For w_{01}, when $Y = 1$ (i.e., the patient has cancer) but $Z = 0$ (i.e., we delay or do not start treatment), the decision is incorrect. This may negatively impact the patient's health, making treatment harder and more complicated. Therefore, the cost c_{01} associated with this decision is very high.

- For w_{10}, when $Y = 0$ (i.e., the patient does not have cancer) but $Z = 1$ (i.e., we start treatment), the decision is also incorrect. However, in this case, the cost is much lower compared to the previous scenario ($w_{10} << w_{01}$) because the error results in unnecessary treatment rather than a life-threatening delay.

The primary objective of this study is to develop a solution for the Problem (2.2), which provides a decision rule for unseen events (patients) when only the features are available in advance, prior to knowing the outcomes. To achieve this, we propose four novel approaches for extracting such a decision rule: Mixed-Integer Programming Model for Optimal Decision Tree (MIPODT), a Random Forest of MIP Models (MIPRF), the Optimal Hyperplane Decision Tree (OHDT), and Separate Estimation and Optimization (SEO).

2.3.1 MIP Model for Optimal Decision Tree (MIPODT)

In recent years, the parallels between classification tasks and decision making under uncertainty have become increasingly apparent. Various classification algorithms (for example: the proposed MIP model by Bertsimas and Dunn (2017)) have been introduced for different applications, focusing on robustness, accuracy, and minimizing misclassification under balanced costs, with decision tree algorithms being a

key example. However, this model may not be suitable when asymmetric costs are involved for associated error types. To address this, we propose a novel MIP model for optimal decision tree (MIPODT) that incorporates asymmetric costs, aiming for more economically favorable outcomes in real-world decision-making scenarios.

Note: There are some differences in terminology between classification and decision-making problems. In classification, a model is trained on a sample and then used to predict outcomes for in/out-of-sample data, which may result in misclassification. To bridge the gap between classification and decision making, in this section and the following ones, prediction will refer to assigning a decision to an event or observation, while misclassification will refer to incorrect decisions, with different types of misclassification corresponding to various error types.

The primary objective of MIPODT is to recursively partition the p-dimensional space $[0,1]^p$ into hierarchical, disjoint regions, forming a classification tree, where p refers to the number of features. An example of a decision tree is shown in Figure 2.3.1. The resulting tree consists of two types of nodes: **Branch nodes**, which apply splits based on parameters a and b to direct points to the left or right branches depending on whether $a^T x_i$ is less than b, and **Leaf nodes**, which are assigned a class that predicts the outcome for all data points within that node, typically choosing the class that appears most frequently.

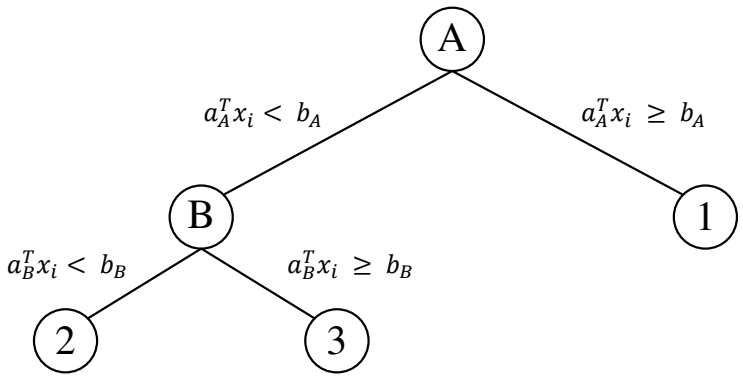

Figure 2.1: An example of a decision tree with two partition nodes and three leaf nodes

During the construction of the decision tree, several discrete decisions need to be made at each step. At each new node in the decision-making process, a choice must be made between branching or stopping. If the decision is made to stop branching at a node, a label must be selected for the new leaf node, identifying it within the structure. Conversely, when opting to branch, a variable must be chosen for this branching, further defining the path within the tree. Furthermore, during the classification of training points based on the evolving tree, a decision must be made regarding which leaf node a point should be assigned to. This assignment must ensure adherence to the tree's structure, maintaining the integrity and logic of the entire system.

By formulating this problem using MIP, we can encompass all of these discrete decisions within a unified problem, rather than treating them as isolated decision events in recursive, top-down approaches. This modeling approach enables us to consider the comprehensive impact of decisions made at the tree's upper levels, rather than solely making locally optimal choices. Moreover, it eliminates the necessity for pruning and impurity measures.

To accommodate the asymmetric cost structure within the decision-making framework, we present a detailed formulation of the MIP model for decision tree. This formulation, due to its complexity and length, is included in Appendix A. The MIP model captures the asymmetrical cost by incorporating distinct penalty terms for different types of misclassifications, thus ensuring that the objective function aligns closely with the cost-sensitive nature of the problem. For a comprehensive understanding of the model's constraints, variables, and objective function, we direct the reader to Appendix A.

2.3.2 A Random Forest of MIP Models (MIPRF)

Random Forest (RF) classification is often preferred over decision tree because RF reduces the risk of overfitting by averaging the predictions of multiple trees, which decreases variance without significantly increasing bias. This ensemble approach results in more accurate, stable, and generalizable models, especially in complex,

noisy, or imbalanced data sets. Additionally, RFs offer better handling of outliers, provide more reliable feature importance estimates, and can effectively manage high-dimensional data while maintaining scalability.

Similar to traditional decision tree classification, the primary objective of RF is to maximize accuracy, often without considering the costs associated with different error types. To address this limitation, we introduce a Random Forest of MIP Models (MIPRF), which incorporates asymmetric cost considerations.

The proposed MIPRF consists of N replicates of the MIPODT. Each tree in this ensemble is provided with a randomly selected subset of features. Additionally, during the construction of each tree, a stochastic subset of the training data is sampled with replacement, a technique known as bootstrapping. As a result, some samples may appear multiple times within a single tree, while others may not be included at all. Once all trees are constructed, their predictions are aggregated. For classification tasks, the final prediction is typically determined by majority voting, where the class receiving the most votes across all trees is selected. A simple structure of the proposed MIPRF is shown in Figure 2.3.2.

2.3.3 Optimal Hyperplane Decision Tree (OHDT)

As explained, in each branch node of the MIPODT, and consequently in the MIPRF, only one attribute or feature is assigned. This approach, however, limits the model's ability to capture the relationships among all features and outcomes, particularly given the defined depth of the tree. To address this limitation, we propose the Optimal Hyperplane Decision Tree (OHDT), a decision tree with a depth of one, where all features are assigned to the first branch node. Using the feature-based data X_{ij} from the training set, the model determines a separation rule by fitting a hyperplane. We then solve a MIP to compute the optimal hyperplane parameters a_j^* and β_0, with the objective of minimizing the cost associated with different error types, as follows:

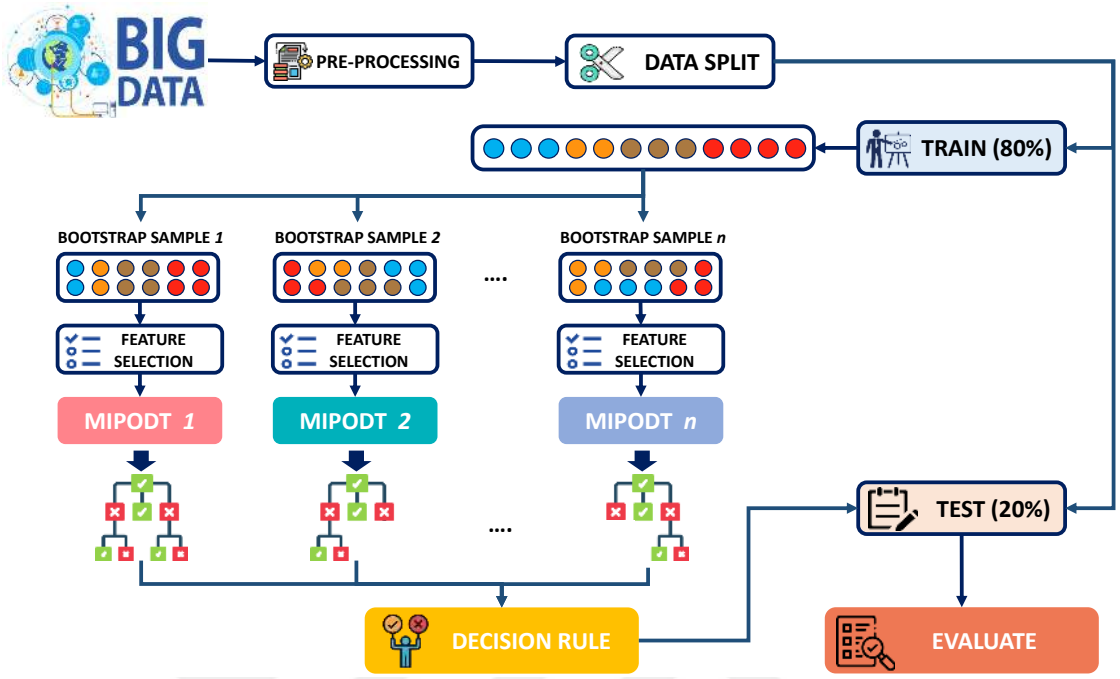

Figure 2.2: Structure of optimal random forest (MIPRF)

$$\min \sum_{k \in K} \sum_{r \in K} mc_{kr} O_{kr}$$

Here, O_{kr} serves to indicate the number of data points belonging to class k that are predicted as class r, and mc_{kr} indicates the cost matrix associated with error types. Due to complexity and length of MIP formulation, the model is included in Appendix B.

To elucidate this concept further, consider the concrete example of seeking linear rules defined by a threshold denoted as β_0 and a set of coefficients $(a_1, ..., a_p)$ where there are only two decisions available (labeled A and B), as shown in Figure 2.3.3. We then have:

$$Z^*|X = \begin{cases} A & \text{if } \sum_{j=1}^{p} a_j x_{ij} < \beta_0 \\ B & \text{otherwise} \end{cases}$$

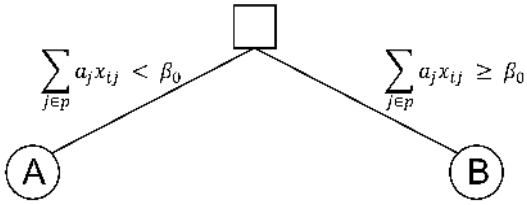

Figure 2.3: Simple decision tree with two-class of decisions

The proposed MIP model in Appendix B provides the optimal hyperplane coefficients, a_j^* and β_0^*, which define the decision rule for classifying new data points. Applying this decision rule to new instances enables optimal decision-making based on this model's outputs.

2.3.4 Separate Estimation and Optimization (SEO)

In this section, we present a separate estimation and optimization (SEO) method for decision-making problems involving k decisions corresponding to the possible outcomes. In decision-making problems, particularly those involving uncertainty, probability trees offer a structured approach to model possible outcomes based on a series of decisions and random events. The SEO translates the problem into a probability tree framework, as illustrated in Figure 2.3.4, which emphasizing the incorporation of asymmetric costs in the decision-making process. At the decision node (square node), there are k alternatives, denoted as Z_1 to Z_k. These alternatives lead to chance nodes (circular nodes), where outcomes Y_1 to Y_k occur with associated conditional probabilities $\hat{p}_1|X$ to $\hat{p}_k|X$ (which depend on feature vector X), each incurring specific costs.

In the literature, several authors have employed machine learning techniques to estimate probabilities based on features. Carey and Hall (1995) used logistic regression to model disease probabilities in genetic counseling. Domingos and Pazzani (1997) applied Naive Bayes for class probability estimation under feature independence. Breiman et al. (1984a) and Breiman (2001) introduced decision trees and Random Forests for robust probability estimation by averaging across multiple trees.

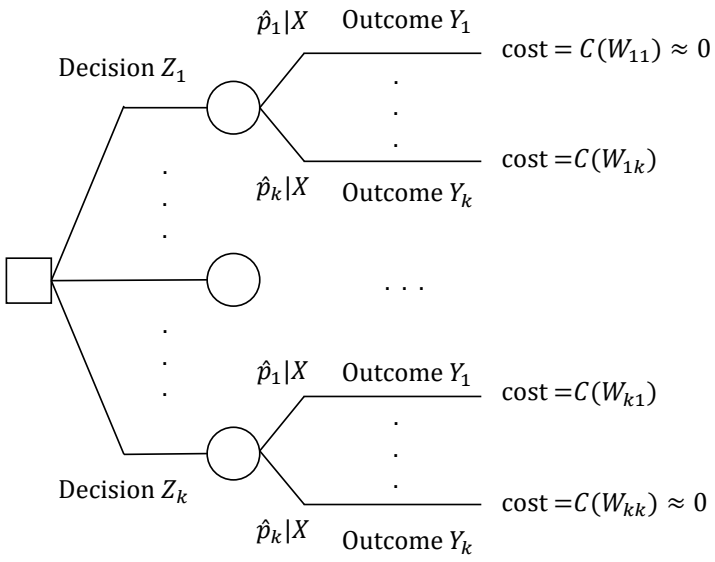

Figure 2.4: Probability tree of decision-making problem

Friedman (2001a) proposed Gradient Boosting Machines (GBM) for sequentially refining probability estimates. LeCun et al. (1989) used neural networks for complex probability estimation in digit recognition, and Platt (1999) extended SVMs to generate probabilistic outputs. However, in this study, we choose to use the Multinomial Logistic Regression model, as it is the simplest form of probability estimator that provides a linear decision rule, making it well-suited for our decision-making problem to derive a decision rule from the available features. This rule is formulated to minimize the expected cost associated with making incorrect decisions, considering the asymmetric costs linked to different types of errors. The goal is to facilitate an informed decision-making process by utilizing the available features effectively to predict the most appropriate action Z given the outcome Y. We present SEO in two cases: multi class classification and binary classification problems:

Multi-class problems using multinomial logistic regression

In the ***first*** step, we implement multinomial logistic regression which extends the binary logistic regression framework. This model estimates the probability of each

class Y_k given the feature set X. The probability for each outcome (class) is modeled as

$$\hat{p}_k(X) = \frac{e^{\beta_{k0}+\sum_{j=1}^{p}\beta_{kj}X_{ij}}}{\sum_{m=0}^{K} e^{\beta_{m0}+\sum_{j=1}^{p}\beta_{mj}X_{ij}}} \tag{2.3}$$

Here, $\hat{p}_k(X)$ represents the probability of outcome Y_k given the features X, and β_{kj} are the coefficients estimated through the regression process.

The ***second*** step of this method involves an optimization approach. We utilize the estimated conditional probabilities from Equation (2.3) to populate our probability tree. For each chance node in the tree, we calculate the expected cost, which is the product of the costs associated with error types and their respective probabilities. Since the goal is to minimize the costs associated with error types, for each observation's features, we assign the optimal decision Z_i^* that incurs the minimum expected cost, as follows:

$$C(W_{ij})^* = \min\left\{ i : \sum_{j=1}^{K} \hat{p}_j(X)\cdot C(W_{1j}), \ldots, \sum_{j=1}^{K} \hat{p}_j(X)\cdot C(W_{kj}) \right\} \rightarrow Z_i^* \tag{2.4}$$

Binary classification problems using binary logistic regression

Given that the logistic function provides its most straightforward decision rule in the case of binary outcomes, we present this rule for a binary classification problem. For binary decision-making problems, specifically designed for scenarios where the decision involves two distinct actions, denoted as Z_0 and Z_1, corresponding to two different outcomes, Y_0 and Y_1. In the binary logistic model, the function (2.3) converts to the following format:

$$p(X) = \frac{e^{\beta_0+\sum_{j=1}^{p}\beta_j X_{ij}}}{1+e^{\beta_0+\sum_{j=1}^{p}\beta_j X_{ij}}} \tag{2.5}$$

In this context, $p(X)$ represents the estimated probability of assigning decision Z based on given feature vector set X. The inverse of the logistic function, commonly referred to as the logit function, is defined as follows:

$$\ln\left(\frac{p(X)}{1-p(X)}\right) = \beta_0 + \sum_{j=1}^{p} \beta_j X_{ij} \tag{2.6}$$

In Equation (2.6), β_0, and β_j's are the unknowns which are estimated through logistic regression fitting on the training data set. To integrate asymmetric cost considerations, $p(X)$ is further refined using a probability tree approach.

Figure 2.3.4 presents a graphical translation of our problem, highlighting the consideration of asymmetric costs in decision making. At the decision node (square node), there are two alternatives, Z_0 and Z_1. These lead to chance nodes (circular nodes), where outcomes and their associated costs are determined based on certain probabilities. For simplicity, we assume that correctly assigning decisions to labels (outcomes) incurs a cost close to zero.

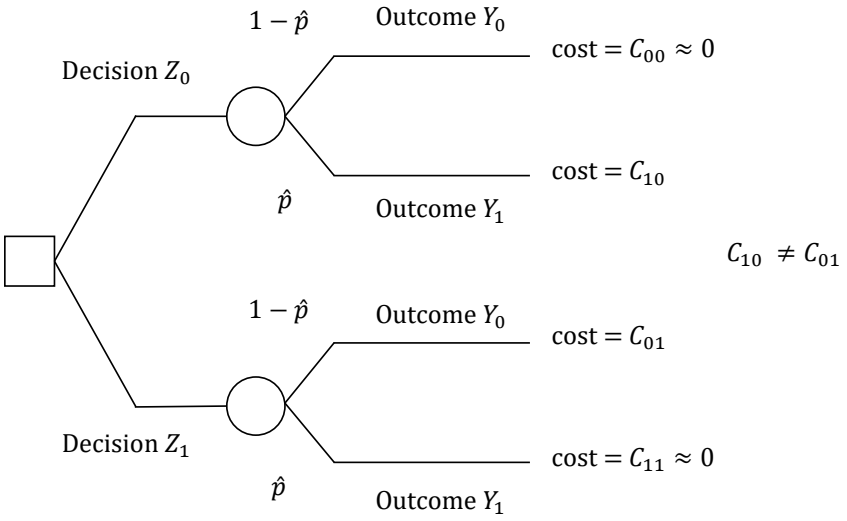

Figure 2.5: Asymmetric cost in decision making

We establish an equilibrium between the expected costs of choosing either branch in the probability tree, as delineated below:

$$(1-\hat{p}) \cdot 0 + \hat{p} \cdot C_{10} = (1-\hat{p}) \cdot C_{01} + \hat{p} \cdot 0 \tag{2.7}$$

Here, C_{10} represents the misclassification cost when decision Z_0 is assigned to outcome Y_1, and C_{01} represents the misclassification cost when decision Z_1 is assigned to outcome Y_0. Simplifying the above equation yields:

$$\hat{p} = \frac{C_{01}}{C_{01} + C_{10}} \tag{2.8}$$

When using the $\hat{p}$ value provided by Equation (2.8), there is no difference in choosing between Z_0 and Z_1, as both yield the same expected cost. By substituting this value into Equation (2.6), we arrive at the following decision rule:

$$\hat{Z}|\mathbf{X} = \begin{cases} \text{Decision } Z_0, & \text{if } \ln\left(\frac{\hat{p}}{1-\hat{p}}\right) \geq \beta_0 + \sum_{j=1}^{p} \beta_j X_{ij}, \\ \text{Decision } Z_1, & \text{otherwise.} \end{cases} \tag{2.9}$$

Based on Equation (2.9), when the given condition is satisfied, the decision Z_0 is assigned; otherwise, the decision Z_1 is assigned accordingly.

For both cases (multi class and binary), the function rule can be applied to unseen events (test data set) when only their features are known, prior to observing the outcomes.

2.3.5 Benchmarks

To evaluate our proposed methods, we require benchmarks from the literature to assess the effectiveness of our approach in minimizing the costs associated with different error types. We implement three benchmark models to compare the performance of our methods.

First, we use a standard random forest classifier, referred to as SRF (Standard Random Forest), imported from scikit-learn, which employs a symmetrical cost function.

Second, as mentioned in the literature, cost-sensitive learning assigns different costs to each type of misclassification. Although recent updates to scikit-learn allow for the application of customized cost functions in classification, they do not do so directly. To address this, we developed a package in PYPI called BijanClassifier, which facilitates fitting base classifiers like Decision Trees (DT), Random Forests

(RF), Gradient Boosting (GB), and Logistic Regression (LR) under a customized cost function. This package combines the TunedThresholdClassifierCV—a classifier that adjusts the decision threshold through cross-validation—and GridSearch methods with cross-validation. The full explanation is available in the library documentation, Bibak (2024). For model tuning, we utilize this package, which optimizes a binary metric, potentially constrained by another metric. We specified the base classification model and a customized cost function, while other parameters remained at their default settings. We applied this approach to decision tree and random forest models, referring to them as HDT (Heuristic Decision Tree) and HRF (Heuristic Random Forest), respectively.

2.4 Key Performance Indicators and Performance Measurement

In this section, we introduce three key performance indicators (KPIs) for our proposed models, with a focus on applying asymmetric costs to decision-making and classification problems.

2.4.1 Cost associated with error types

In this chapter, we implement four different models for decision-making problems presented in Section 3.3, taking into account the asymmetric costs associated with error types. We train these models on a training set to derive an efficient decision rule that minimizes the costs incurred from incorrect decisions, see Equation (2.2). Once the rule is established, it is applied to an out-of-sample test set. As previously mentioned, these costs are calculated by multiplying the error types by the cost matrix provided in Appendix C. The primary KPI of this study is the comparison of the models based on the costs incurred in the test set.

2.4.2 Accuracy

While the primary objective across all methods is to minimize the cost associated with different types of errors, analyzing the accuracy of our proposed models serves

as an additional metric for evaluating their performance. We anticipate that minimizing these costs may necessitate a trade-off in model accuracy, implying that a reduction in cost could result in lower accuracy. In this context, accuracy is defined as the percentage of correct prediction decisions over all examined events in the out-of-sample set.

2.4.3 Computational Run Time

Three of our four proposed methods are formulated as mixed-integer programming models with non-linear constraints. Due to the large number of features and observations, the resulting high number of variables makes computational run time a critical criterion for evaluating the effectiveness of these methods. The computations are performed on an Asus TUF Gaming A15 laptop, equipped with an AMD Ryzen 7 6800H CPU, 48GB RAM, a 2TB SSD, and an RTX 3060 GPU. For each method, across different data sets and varying ratios of asymmetric costs, we measure the run time in seconds, providing an additional KPI for model evaluation.

2.5 Numerical Results

In this section, we assess four proposed decision-making methods across all provided data sets under different asymmetry ratios ($D = 0, 1, 2, 3$). We present details on the data sets in Appendix C. We also present data cleaning and processing issues and some feature elimination considerations in the same section. In addition, concept of asymmetry ratios is presented in Appendix C.2.

We then compare the performance and KPI's of our models in terms of cost reduction, accuracy, and run time with the SRF method from scikit-learn (Pedregosa et al., 2011) without using a customized cost function and the HDT and HRF methods from the BijanClassifier package (Bibak, 2024) using a customized cost function. To assess and compare performances of the implemented methods, we use a common approach where the data set is randomly split into separate training and test (out-of-sample) sets with an 80%-20% ratio. All methods are trained on the same training set and performance is evaluated on the test set. A summary of the

procedure of the proposed methods in this study is shown in Figure 2.5.

Given that our decision-making tools account for asymmetric costs, the primary comparison metric is based on the cost associated with error types in the out-of-sample set. This approach ensures that any misclassification is evaluated with unbalanced cost considerations. In addition to the asymmetric cost metric, we also evaluate the models' accuracy and run time as supplementary criteria to assess the overall performance of our methods.

We use the Gurobi solver (Gurobi Optimization, LLC, 2024) for constrained optimization and perform all of the computations in Python.

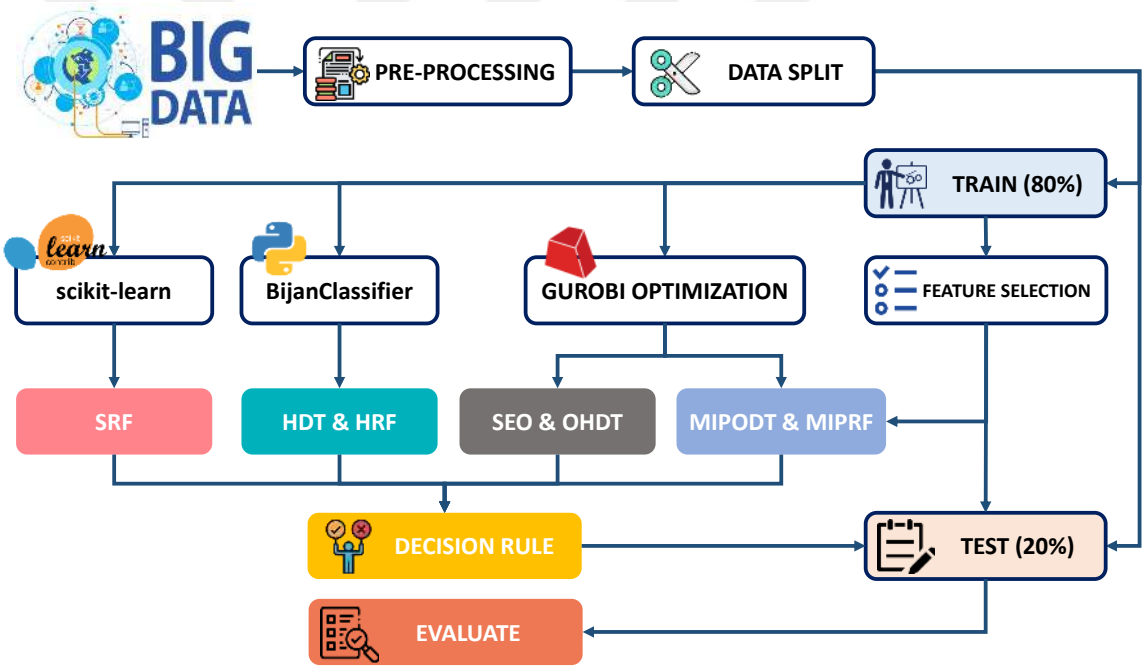

Figure 2.6: Procedure of running proposed strategies

2.5.1 Evaluating the cost associated with error types

The data presented in Table 2.1 provides an analysis of the costs associated with misclassification errors across all evaluated methods, including the four proposed

models and three benchmark models. These models are tested across four distinct data sets, with a focus on scenarios characterized by varying ratio of asymmetry, denoted as $D = 0, 1, 2, 3$. It is observed that at $D = 0$, where the cost function is symmetric, uniform performance is expected across all models. For each data set, the model with the minimum cost is highlighted in **bold** font. The results clearly indicate that as the ratio of asymmetry (D) increases, there is a corresponding rise in the cost of misclassification for all methods. This effect is particularly pronounced at $D = 2$ and $D = 3$, where the ratio between the maximum and minimum costs is significantly high.

As demonstrated in Table 2.1, our proposed models significantly outperform the SRF benchmark, which assumes balanced costs, in terms of cost reduction across various scenarios, especially when $D > 0$. This advantage is particularly notable in conditions with higher ratio of asymmetry (D). For example, in the SECOM data set, when $D = 2$, the OHDT model outperforms SRF, achieving a cost reduction of up to 96%. However, it should be noted that a direct comparison with SRF may not be entirely fair, as SRF assumes balanced costs, whereas our approach considers unbalanced costs. Nevertheless, in most real-world applications, traditional methods such as random forests or other standard classifiers are often employed for decision-making problems.

The other two benchmarks, HDT and HRF, only outperform our methods in one instance, specifically in the SECOM data set under $D = 1$, where their costs are very close to those of our methods. In other scenarios, the costs associated with the benchmark models are substantially higher than those of our proposed methods or, in extreme cases, equivalent to our results.

The F1-score is a widely used metric in classification tasks that balances precision and recall, providing an overall measure of model performance. However, it does not inherently account for the cost associated with error types, particularly in scenarios with asymmetric costs. In our analysis, we observed a general trend where, as the ratio of asymmetry increased from $D = 0$ to $D = 3$, the cost reduction achieved by our proposed approaches—MIPODT, MIPRF, OHDT, and SEO—compared to

Benchmark methods like HDT, HRF, and SRF became more pronounced. This demonstrates the ability of our methods to better minimize costs associated with high-stakes misclassifications under increasing decision complexity. Notably, this improvement in cost reduction is accompanied by a decrease in F1-scores, highlighting the trade-off between traditional classification performance metrics and economic cost optimization.

Table 2.1: Cost comparison

Data Set	n	p	K	D	Cost (in test set)							F1
					MIPODT	MIPRF	OHDT	SEO	HDT	HRF	SRF	
SECOM	1567	590	2	0	53	53	55	53	53	**49**	53	0.000
				1	471	477	477	483	588	**429**	477	0.000
				2	1,788	1,788	**1,764**	1,788	22,902	1,788	47,700	0.262
				3	1,788	1,788	**1,764**	1,788	1,500,402	1,788	3,180,000	0.262
Breast Cancer	569	30	2	0	6	6	**5**	**5**	8	**5**	**5**	0.966
				1	12	10	**6**	16	36	26	32	0.940
				2	**420**	1,020	520	1,200	2,176	682	3,020	0.268
				3	1,080	840	**520**	1,420	1,420	1,420	60,020	0.000
Heart Disease	303	13	5	0	29	31	30	**26**	36	30	30	0.355
				1	254	295	285	**244**	337	280	290	0.337
				2	3,039	2,997	2,158	**1,162**	8,160	7,884	7,904	0.255
				3	43,871	31,901	**12,076**	14,678	57,036	77,814	77,924	0.031
Haberman	306	3	2	0	**17**	19	19	18	**17**	18	18	0.830
				1	82	**78**	120	88	88	102	180	0.000
				2	1,760	1,760	1,780	**880**	**880**	1,020	18,000	0.000
				3	200,760	200,760	200,780	**880**	**880**	**880**	3,600,000	0.000

Figure 2.5.1 provides a visual representation of the percentage of cost reduction achieved by our proposed methods compared to SRF (which uses the traditional random forest for classification, shown in the right figure) and HDT and HRF (which

employ cost-sensitive learning for classification, shown in the left figure). As illustrated in the right figure, under $D = 0$, there is minimal improvement in cost reduction since both methods operate with balanced costs. However, when $D > 0$, cost reductions can reach up to 95% and in some cases up to 100%. In contrast, the left figure demonstrates that across almost all degrees of asymmetry, there is a noticeable improvement in cost reduction. Nevertheless, it is evident that in some cases, compared to the right figure, the cost improvement is lower.

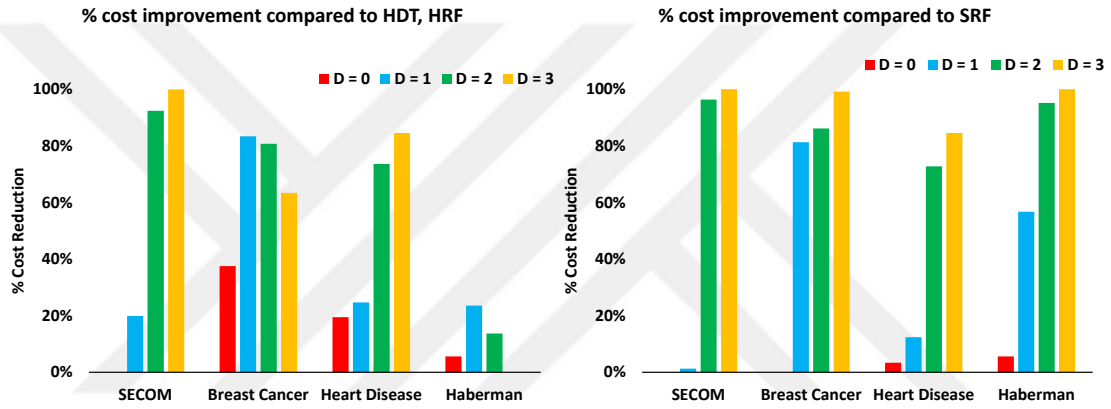

Figure 2.7: % cost improvement of proposed methods compared to the benchmarks

2.5.2 Evaluating the accuracy performance

As previously stated, the primary objective of this study is to develop decision-making approaches tailored to scenarios where incorrect decisions (classifications) incur asymmetric costs, with the ultimate aim of minimizing these costs. While accuracy remains a relevant metric, it is not the central focus of our proposed methods.

Table 2.2 presents the accuracy levels of our models on the test set. A notable observation is that the asymmetry ratio (D) does not impact SRF, which operates under balanced costs. Since SRF prioritizes accuracy based on predefined, balanced

costs, it remains unaffected by variations in the cost function.

As shown in Table 2.2, for most of our proposed methods, an increase in the degree of asymmetry (D) correlates with a decrease in accuracy. Conversely, as illustrated in Table 2.1, cost reduction improves as asymmetry increases. This trend suggests a trade-off: focusing on the most accurate models may lead to diminished cost reduction, and vice versa. For example, in the case of heart disease data, the OHDT method achieves accuracies of 50.8%, 49.2%, 44.3%, and 44.26% at D values of 0, 1, 2 and 3, respectively. However, the cost reduction relative to benchmark methods is approximately 0%, 2%, 73%, and 85%. As depicted in Figure 2.5.2, across various degrees of asymmetry in all data sets, an inverse correlation between accuracy and percentage cost reduction is evident. This indicates that prioritizing maximal cost reduction over benchmarks may lead to a decrease in the accuracy of the proposed models. Notably, methods like MIPODT and MIPRF outperform other proposed methods in terms of accuracy. Overall, it is expected that in some scenarios, SRF may outperform our proposed methods, as our methods sacrifice accuracy to minimize costs.

2.5.3 Evaluating the Computational Run Time

Table 2.3 illustrates the computational run time of the implemented methods across various data sets. As anticipated, the MIPODT, MMIPRF, and OHDT models, which involve mixed-integer optimization with non-linear constraints, exhibit significantly longer run times compared to other methods. As discussed in Section C.1, feature selection was applied to MIPODT and MIPRF, utilizing a subset of features to reduce computational time without adversely affecting model accuracy. Notably, MIPRF, due to its multiple replicates of MIPODT, requires substantially more run time than MIPODT. The SEO method, rooted in logistic regression (LR), benefits from the inherent efficiency of LR, resulting in considerably lower run times compared to our other proposed methods. While OHDT only considers a single level in its decision tree but includes all features in the optimization process, its run time remains lower than that of MIPODT and MIPRF.

Table 2.2: Accuracy comparison

Data Set	n	p	K	D	Accuracy %						
					MIPODT	MIPRF	OHDT	SEO	HDT	HRF	SRF
SECOM	1567	590	2	0	84.9	84.9	84.3	84.9	84.9	**86.0**	84.9
				1	**84.9**	**84.9**	84.3	84.6	77.5	83.7	**84.9**
				2	15.1	15.1	16.2	15.1	73.8	15.1	**84.9**
				3	15.1	15.1	16.2	15.1	73.8	15.1	**84.9**
Breast Cancer	569	30	2	0	94.7	94.7	95.6	95.6	93.0	95.2	**96.5**
				1	93.9	95.6	**97.4**	93.0	91.2	95.6	96.5
				2	81.6	55.3	77.2	47.4	91.2	72.8	**96.5**
				3	53.0	63.15	77.15	37.71	37.71	37.71	**96.5**
Heart Disease	303	13	5	0	52.5	49.0	50.8	**57.3**	41.0	50.8	50.8
				1	**54.1**	49.0	49.2	**54.1**	42.6	50.8	50.8
				2	39.0	11.0	44.3	34.4	42.6	49.2	**50.8**
				3	6.55	8.19	44.26	6.55	42.62	49.2	**50.8**
Haberman	306	3	2	0	**72.6**	69.3	69.4	71.0	**72.6**	70.9	70.9
				1	40.3	43.5	61.3	29.0	29.0	56.5	**70.9**
				2	37.1	37.1	35.5	29.0	29.0	56.5	**70.9**
				3	37.09	37.09	35.48	29.0	29.0	29.0	**70.9**

The HDT, HRF, and SRF methods are designed with heuristic approaches for classification, contributing to their swift run times. Although our proposed methods, particularly those involving optimization, have higher run times compared to benchmark models, they offer the advantage of significantly reducing the costs associated with error types—a critical trade-off. It is also important to note that the primary objective is to derive an efficient decision rule; hence, the run time is not a primary concern. While extracting such a decision rule may require considerable computational effort, its potential to reduce costs in the long term, particularly in applications such as healthcare (e.g., cancer diagnosis), is highly valuable.

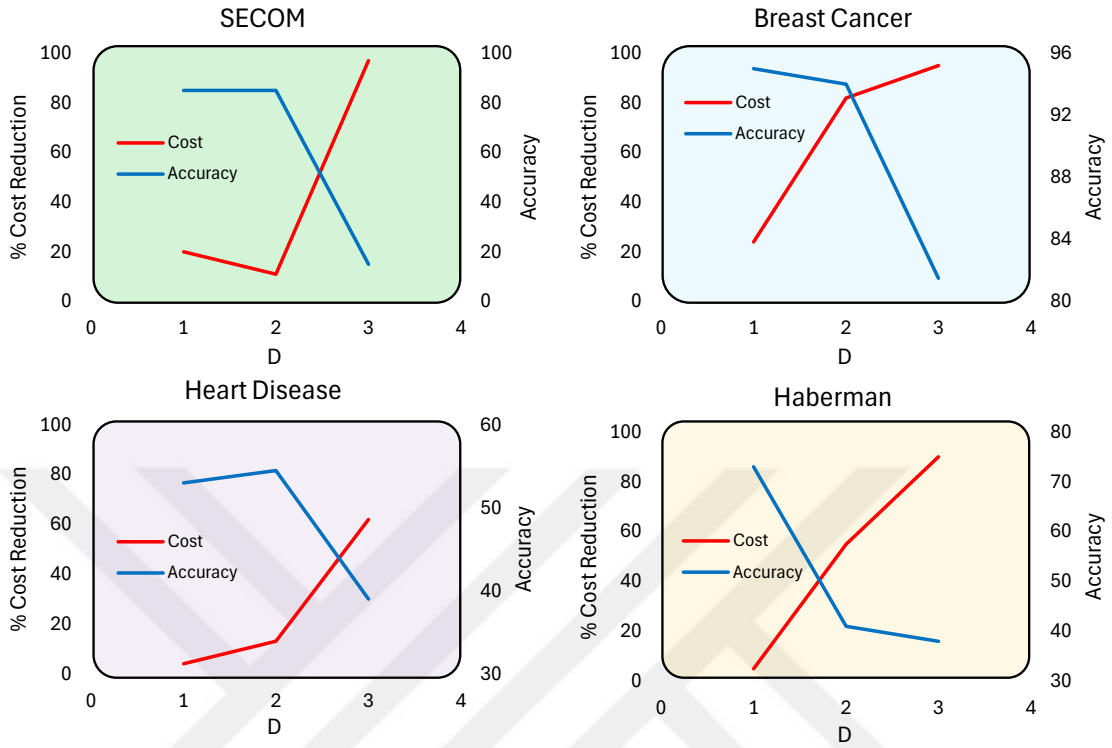

Figure 2.8: Cost vs. Accuracy of MIPODT Model

2.5.4 Evaluating the Sample Size on Cost

In this section, we evaluate the impact of training sample size on the cost associated with wrong decisions in the test set across four data sets: SECOM, Breast Cancer, Heart Disease, and Haberman, as depicted in Figure 2.5.4. To conduct this analysis, each data set is randomly split into training and test sets using an 80-20% ratio. The size of the training set is systematically varied, while the test set size is held constant. The MIPODT model is chosen and then trained on each of these varying training sets to extract decision rules, which are subsequently applied to the test set to record the associated costs. In this experiment, the ration of cost is fixed to $D = 2$.

The results across all data sets exhibited a consistent pattern: as the size of the training set increased, the cost associated with wrong decisions in the test set

Table 2.3: Computational run time comparison

Data Set	n	p	K	D	Computational Run Time (seconds)						
					MIPODT	MIPRF	OHDT	SEO	HDT	HRF	SRF
SECOM	1567	590	2	0	174.08	1859.79	3540.16	0.14	1.16	2.75	0.24
				1	114.24	2004.06	3235.68	0.18	1.09	2.50	0.20
				2	110.43	1661.58	1000.83	0.18	1.10	2.54	0.19
				3	110.43	1661.58	1000.83	0.18	1.17	2.71	0.19
Breast Cancer	569	30	2	0	226.65	7016.27	366.16	0.14	1.24	1.88	0.25
				1	130.78	5016.68	689.10	0.10	1.05	2.43	0.13
				2	354.28	3371.74	68.32	0.13	1.05	2.21	0.13
				3	420.93	3962.09	8.17	0.37	1.19	2.22	0.13
Heart Disease	303	13	5	0	1132.13	31542.14	37.22	0.06	1.12	39.88	0.59
				1	426.85	20458.76	25.47	0.12	1.19	41.16	0.09
				2	562.34	13811.07	21.99	0.06	1.11	33.58	0.17
				3	121.23	4087.19	16.34	0.73	2.99	17.18	0.17
Haberman	306	3	2	0	75.75	2383.33	12.37	0.10	1.09	1.96	0.08
				1	197.84	3706.14	14.72	0.11	1.06	1.81	0.08
				2	35.35	1326.18	2.18	0.08	1.16	2.01	0.09
				3	69.17	936.37	1.63	0.08	1.19	1.92	0.09

decreased exponentially. This trend underscores the importance of a sufficiently large training sample in improving the robustness and efficiency of the decision-making model. A larger training set provides more comprehensive coverage of the underlying data distribution, enabling the model to better generalize and reduce the likelihood of costly errors when applied to new, unseen data (test set). This finding highlights the critical balance between training sample size and model performance, reinforcing the need for careful consideration of training data adequacy in decision-making problems where minimizing error costs is paramount.

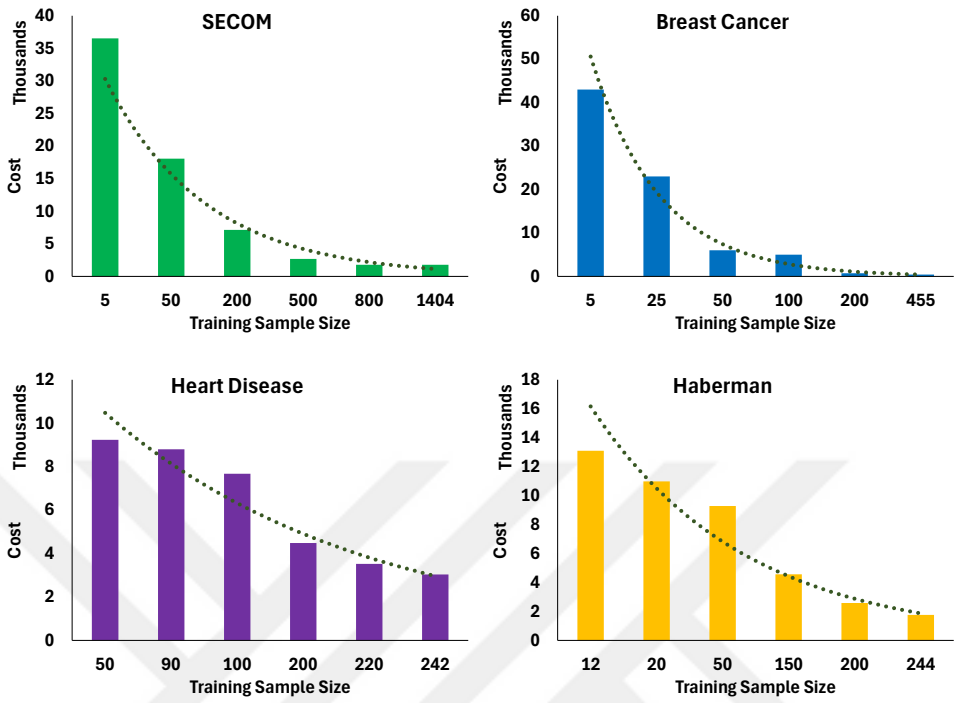

Figure 2.9: Impact of training sample size on the cost associated with wrong decisions in the test set

2.6 Conclusion

This chapter presents innovative methods that bridge machine learning and optimization to address decision-making challenges in uncertain environments with asymmetric error costs. Our proposed approaches demonstrate substantial improvements over traditional benchmarks, showcasing their robustness in both binary and multi-class classification problems. By reducing misclassification costs by up to 95%, our methods offer a valuable solution for practitioners who need to make proactive decisions with limited data in high-stakes contexts. These results highlight the potential of our techniques to enhance decision quality across various applications, particularly when the economic impact of incorrect decisions is critical. Future research may further refine these methods by exploring their scalability and adaptability across broader domains.

Chapter 3

A DATA-DRIVEN SINGLE-PERIOD LOT SIZING PROBLEM WITH RANDOM YIELD

3.1 Introduction

In several industries, a significant planning challenge arises due to yield issues, where the actual production or replenishment quantity received differs from the planned quantity. Additionally, the yield of the production or replenishment process is typically uncertain, adding another dimension to the challenge. In the fields of operations research and management, a considerable body of literature has investigated production/inventory planning under random yield, with many fundamental contributions. A comprehensive survey by Yano and Lee (1995a) classifies the literature and presents the main findings up until the mid-nineties.

In this chapter, we are motivated by a case from the semiconductor industry where wafer manufacturing is known to be plagued by yield issues (Bitran and Gilbert (1994) for example). Despite huge improvements in design and technology, data and information from a recent major European research project, Productive 4.0, shows that yield uncertainty is still a major issue in semiconductor manufacturing. Based on recent data from a major manufacturer, Diefenbach (2023) reports cases where yield rates may be low for certain products, especially in the early phases of production. In particular, planning still needs to take into account yield uncertainty even as product variety increases and lead times shorten. While yield uncertainty and the resulting loss are still significant issues, yield prediction methods have become promising due to advances in machine learning. It now appears that even for new products, design and environment-based features provide valuable information to improve yield rate predictions. Motivated by the above, we inves-

tigate a data-based estimation and optimization approach for a planning problem with yield uncertainty where additional features that can help estimate the yield rate are available.

Data-based optimization under uncertainty aims to obtain simple but effective decision rules using a past sample of observations that are related to the uncertain factors that affect planning. This requires combining the estimation related to extract predictions from the data and the optimization under uncertainty that solves the planning problem. We develop approaches that address a lot sizing problem with yield and demand uncertainty. We test the performance of the approaches using synthetic data as well as real data from a publicly available data set for a semiconductor yield. This data set comprises 590 features for each observation of the yield. The prediction component of the problem is therefore not trivial and requires pre-processing and model reduction (see, for example (Lee et al., 2019)). This constitutes a good example of some of the challenges from a real-sized data set. Lee et al. (2019) show that for this public data set accurate predictions can be made when the feature information is used. Our focus in this study is not on predictions but on prescriptions: how to find the optimal lot size decision when feature data is available and on assessing the prescriptive (cost) performance of different methods.

We can summarize the contributions of this chapter as follows:

- We contribute to the newly developing prescriptive analytics literature in operations by studying the data-driven version of a single-period, single-stage, single-quantity problem for the random (multiplicative) yield with a large number of features and limited data.

- We develop and implement joint estimation and optimization methods that are adapted to the random yield case and measure their performance for different training sample sizes and number of features.

- The results demonstrate that, when the best method is used, one can achieve cost reductions of 90% for (modified) real data and even higher for synthetic data when compared to the method that disregards features.

This chapter is organized as follows: a literature review is presented in Section 3.2. The problem and the models including the data-driven case, are introduced in Section 3.3. Section 3.4 presents the solution approaches within the data-driven framework. The main key performance indicators (KPIs) are introduced in Section 3.5. In Section 3.6, the synthetic and real data sets are explained in detail and all the required pre-processing and feature selection techniques are described. In addition, we assess and compare the performance of the proposed approaches. Finally, conclusions are presented in Section 3.7.

3.2 Literature Review

We present a data-based approach for a lot sizing problem where the lot size to be released is determined taking into account the yield uncertainty in manufacturing. In particular, our main interest is in the case where there is data available on external features that influence yield and demand. In the absence of such feature data, there is a well-established literature for inventory problems under random yield and random demand going back to the pioneering work of Karlin (1958), who investigated optimal ordering policies for a single-stage single-period inventory model where the quantity received is different from the quantity ordered. This led to the development of an important sub-field of inventory theory. Yano and Lee (1995a) and Grosfeld-Nir (2004) present comprehensive reviews of this sub-field.

Our particular assumptions on random yield are based on the multiplicative (also called proportional) yield model, where the quantity received is a random fraction of the ordered lot size. For this model, Shih (1980) investigates the optimality condition and presents solutions for some specific cases. Gerchak and Parlar (1990) investigate further properties of the optimal solution, and Henig and Gerchak (1990) extend the results to a more general setting and to multi-period situations. Okyay, Karaesmen, and Ozekici (2014) investigate three random yield models, including the multiplicative yield case, under the assumption that the random demand and supply are not necessarily independent and establish conditions for the existence of structured solutions.

Since the early foundations, inventory theory has addressed issues of optimization under limited data on uncertain demand. For instance, Scarf (1958b) assumes that the demand distribution is not completely known, but some of its parameters may be known and proposes a robust optimization approach. Scarf (1959) also presents a case where Bayesian updates are performed. More recent research directly focuses on data-based inventory optimization, exploiting the empirical distribution of the available data. (Levi, Roundy, and Shmoys, 2007) establish the convergence of the Sample Average Approximation approach in an inventory setting. Liyanage and Shanthikumar (2005) propose a method that integrates estimation from limited sample data and optimization of operational decisions and shows that integrating estimation and optimization leads to improved solutions. Chen and Xie (2021) propose a robust optimization based approach for an inventory problem with random yield where the joint distributions are estimated from limited data.

Recent years have seen an increase in data-based inventory optimization models where additional feature data that provides information about demand is taken into account. For instance, Sachs (2015) considers price and temperature data in addition to demand observations and presents a model where the ordering quantity linearly depends on these two features. Ban and Rudin (2019) consider a more general model where there might be many features that might potentially influence demand and combine tools from statistical learning to address the prediction and optimization problem while addressing feature selection (model reduction) issues simultaneously. They show that significant gains can be made in inventory cost minimization when feature data is used effectively. Many other recent papers (for instance Van der Laan,Teunter, Romeijnders, and Kilic, 2020, Oroojloyjadid, Snyder and Takac, 2020, Neghhab, Khayyati and Karaesmen, 2023, Liu, Letchford, and Svetunkov, 2022, Notz and Pibernik, 2022, Mandl et al., 2022 and Mandl and Minner, 2023) present methodology that supports the potential of data-based inventory optimization with feature information. Erkip (2023) presents some of the opportunities and new challenges presented by data availability on inventory problems.

Concepts and tools from statistical/machine learning are key factors when pos-

sibly big data (i.e., many features) is used for the prediction of random quantities that then serve as inputs to an operational problem. Bertsimas and Kallus (2020) refer to the framework of generating optimal operational decisions based on data including a large number of features as *prescriptive analytics* in contrast to *predictive analytics*, which is the main focus of machine learning methods. They propose a systematic approach for combining learning and stochastic optimization and present several examples.

In this chapter, our motivation comes from a lot sizing problem with random yields that is encountered in semiconductor manufacturing. It is known that the random semiconductor yield depends on many measurable features, including the design parameters and production parameters specific to the product and the lot. There are a few papers that focus on semiconductor yield prediction using some specific feature information. For instance, Susto et al. (2015) use monitoring and physical metrology data, and Al-Kharaz et al. (2019) use process alert data. Unfortunately, full-feature data on product and process characteristics is proprietary and is rarely openly available. On the other hand, a fairly complete data set (McCann and Johnston, 2008b) is publicly available in an open repository. Lee et al. (2019) investigate machine learning-based methods for classification using this data set. Unlike the above papers focusing on yield prediction, our focus is on the operational problem of lot sizing under yield uncertainty when features are available. We develop estimation and optimization methods using the publicly available data (with some modifications) in McCann and Johnston (2008b) to ensure that our methodology can cope with the complications and size of this representative data. We also assess the performance of our methods on this data to characterize the potential benefits of using the feature data. But we perform further tests and assessments in a synthetic data set.

3.3 The Problem and the Models

We address a realistic but simplified version of the following semi-conductor planning problem. The manufacturer has a contract to deliver d *good-quality* items by a due

date. Our motivating example comes from the production process of wafers, which are used in semiconductor manufacturing. Each wafer consists of a large number of microchips. The manufacturer typically produces its microchips in lots, where each lot is a batch of wafers. Individual lot sizes may differ depending on more involved production planning constraints. We therefore focus on the cumulative production quantity decision to be made in order to meet the contract requirements.

In order to relate to the rich existing literature, we assume a more general situation where the contracted quantity is a random variable D. The manufacturing process is imperfect, and whenever Q items (i.e., wafers) are released to the manufacturing stage, only an uncertain proportion passes the quality threshold and is classified as *good* at the end. Let us denote by Y the uncertain yield rate (the proportion of good items). We assume that whenever Q items are released, QY is good for delivery. This is the well-known multiplicative (or proportional) model of yield uncertainty (Henig and Gerchak (1990)). For our case, we assume that Y takes values between 0 and 1. A simplified version of the planning problem takes into account the following cost structure: each unit short of the contract target D incurs a unit underage cost of b, and each unit above the target generates an overage cost of h. The objective is to choose the lot size Q that minimizes the expected cost.

Our focus in this chapter is on the case where there are features that can potentially provide information on the random yield or demand. Moreover, the relationship between the features and yield/demand has to be learned from the data. This motivates a data-driven optimization problem with a potentially large number of features.

We review the known results from the well-studied case with no feature information and full distributional information in Appendix D. Some of these results are used in the case with feature information

3.3.1 The Case with Full Distributional Information

Let us consider the situation with on random demand D and yield Y where additional features that provide information can be available. We assume that the fea-

tures are available before determining the lot size Q. More precisely, let us assume that $\mathbf{X}$ is a vector consisting of r features that can potentially provide information on D and $\mathbf{U}$ is a vector consisting of s features that can provide information on Y. If full probabilistic information on the conditional random variables $D|\mathbf{X}$ and $Y|\mathbf{U}$ is available, we can exploit the known results from Appendix D and obtain the following optimality condition:

$$\min z(Q) = \mathbb{E}\left[\left(b(D|\mathbf{X}=\mathbf{x} \; - \; QY|\mathbf{U}=\mathbf{u})^+ + h(QY|\mathbf{U}=\mathbf{u} \; - \; D|\mathbf{X}=\mathbf{x})^+\right)\right] \tag{3.1}$$

Remark: We note that there are some papers that consider infinite-horizon inventory problems under Markov-modulated random demand (as in (Song and Zipkin, 1993a)) or random supply (as in (Özekici and Parlar, 1999)) where the modulating environment process is an observable feature. The above model is in the same spirit but considers a single-period, single-stage, single quantity problem with a new challenge: a large number of features and limited data rather than complete probabilistic information.

We observe that the optimality condition (D.2) (from Appendix D) holds for the conditional random variables $D|\mathbf{X}=\mathbf{x}$ and $Y|\mathbf{U}=\mathbf{u}$ and the optimal feature-dependent lot size can be expressed as:

$$\frac{\mathbb{E}[(Y|\mathbf{U})\mathbb{1}_{\{(D|\mathbf{X})\leq Q^*(Y|\mathbf{U})\}}]}{\mathbb{E}[(Y|\mathbf{U})]} = \frac{b}{h+b} \tag{3.2}$$

The optimal lot size now depends on the observed features $\mathbf{X} = (x_1, x_2, .., x_r)$ and $\mathbf{U} = (u_1, u_2, ..., u_s)$. While it is helpful to have a theoretical expression that characterizes the optimal lot size, it is not plausible to have accurate estimations of the conditional distribution with a large number of features and limited data to work with. We note that based on Equation (3.2), the relationship between the optimal lot size and the expected yield is nonlinear.

Remark: There are other data-driven examples where the optimality condition is characterized by similar conditional probability distribution expressions (for

instance the newsvendor problem with features). The useful feature data for prescriptive analytics are those that are available prior to the decision (as we also assume). Additional unobservable or partially observable features might require transformations but that is out of our scope.

3.3.2 Value of Feature Information with Complete Distributional Information

Before moving on to addressing the case with a large number of features and limited data, let us investigate the complete distributional information case to gain some insight into what can be gained by feature information. Infinite-horizon models in the same spirit were studied by Song and Zipkin (1993b) and Özekici and Parlar (1999). We assume that the yield depends on a random feature vector $\mathbf{U} = (U_1, U_2, ..., U_s)$, where each feature can take the value of 0 or 1. We assume that the conditional distributions of the random variables $Y|\mathbf{U}$ and the distribution of random demand D are completely known. To simplify the notation, let us denote by $\mathbf{u}_0 = (u_{1,0}, u_{2,0}, ..., u_{s,0})$ a fixed vector of features and by $Y_0 \equiv Y|\mathbf{U} = \mathbf{u}_0$. Let us further assume that $P(\mathbf{U} = \mathbf{u}_0) = \alpha_0$ $(0 \leq \alpha_0 \leq 1)$. Note that we would observe $\mathbf{U} = \mathbf{u}_0$ α_0 fraction of the time and $\mathbf{U} \neq \mathbf{u}_0$ the rest of the time.

If we consider the optimal lot sizing problem (3.1), clearly, a decision maker who observes the feature vector $\mathbf{u}_0$ before the decision would determine the quantity as the solution that depends on the observed $\mathbf{u}_0$ as in (3.2). On the other hand, if the decision maker is not able to observe the features or does not take them into account, he would consider the unconditional random variable Y and would choose a single Q by solving (D.2) (from Appendix D) regardless of the features. Except for special cases, Y_0 and Y are different. Therefore, the corresponding lot sizes Q_0^* and Q^* are also different. Let us denote the expected cost of the model that does not use features as $z(Q^*)$ (from 3.1). We can also see that the expected cost for the feature-based model is: $z(Q_0^*)$ when $\mathbf{U} = \mathbf{u}_0$. The conditional value of feature information is the difference in the two expected costs. This value depends on the difference of the expected costs when different lot sizes are taken. The next proposition quantifies the difference in the expected costs of two systems using respective lot sizes of $\hat{Q}$

and Q^*.

Proposition 1. *Consider Q_u^* as the optimal lot size as a function of the observed features $\mathbf{u}_0$, and $\hat{Q}$ as an alternative choice different from Q_u^*. The difference in expected costs is given by:*

$$z(\hat{Q}) - z(Q_u^*) = (b+h)\mathbb{E}[\ |\hat{Q}-(D/Y)|\ ]\mathbb{E}[Y\mathbb{1}_{\{\hat{Q}Y \leq D \leq Q_u^* Y\}}]$$

The proof of this proposition can be found in Appendix E.

Proposition 1 characterizes the difference in the expected cost when choosing the suboptimal lot size $\hat{Q}$ instead of the feature optimal lot size Q_u^* when the particular feature information is available (i.e., α_0 fraction of the time). It is observed that this difference depends on two quantities: first, on the expected absolute deviation between $\hat{Q}$ and D/Y and second, on the expected yield in the interval when demand falls between $\hat{Q}Y$ and Q_u^*Y. For the second term, the denser the joint probability distribution within this interval, the greater the discrepancy becomes. We conclude that as $\hat{Q}$ diverges from Q_u^*, the corresponding expected cost $z(\hat{Q})$ also diverges from the optimal cost $z(Q_u^*)$. Moreover, since $z(Q)$ is a convex function of Q (Okyay et al., 2014), the cost deviations can be significant as $|\hat{Q} - Q_u^*|$ grows. Coming back to the value of feature information, this implies that the feature based expected cost, which uses different lot sizes for each value of observed feature (Q_0^*, Q_1^*) can have significantly lower expected costs than the expected cost when the features are not taken into account (i.e. when Q^* is used). In fact, for any possible vector of features. Proposition 1 confirms that there might be significant value in using the feature information to determine the lot size if full distributional information is available.

3.3.3 The Data Driven Case

Using a sample of observations (i.e., a training data set), we would like to solve the optimization problem (3.1) where $\mathbf{X}$ is a vector consisting of r features that can potentially provide information on random demand D and $\mathbf{U}$ is a vector consisting of s features that can provide information on the random yield rate Y. Let us assume

that we have observed a sample of n demand and yield realizations (d_i and y_i) along with the corresponding values of the features $\mathbf{X}_i$ and $\mathbf{U}_i$ for each observation i. Each observation in the sample is then of the following form:

$$\mathbf{S}_i = ((d_i, y_i), (x_{i,1}, x_{i,2}, ..., x_{i,r}), (u_{i,1}, u_{i,2}, ..., u_{i,s}))$$

Where $x_{i,j}$ and $u_{i,k}$ denote the realization for demand features j and yield feature k for observation i corresponding to demand and yield respectively. To provide an example of the potential challenges of data-based optimization, the representative semiconductor data set has more than 500 features for a relatively small number of yield observations. A part of this data set is presented in Table 3.1. The second column (Pass/Fail) reports the yield outcome for a given production. For each individual yield outcome, the next 590 columns report the corresponding features. Most of the feature values are unique and continuous, and there are only 1567 observations, a small number with respect to the number of features. This makes the direct use of the formulation (3.1) impossible.

Remark: There is no information on the SECOM data set whether all features are observable before the lot size decision but we assume that they are.

3.4 Data-Driven Solutions

This section presents methods that combine estimation and optimization in different ways to address the issue of a large number of features and a limited sample size. With a large number of features and a limited sample size, there are too few observations to directly fit a probability distribution for any given combination of features. Machine learning methods that estimate the relationship between the features and the outcome from training data are therefore helpful. There is some recent evidence that integrated machine learning and optimization approaches as in Ban and Rudin (2019), Bertsimas and Kallus (2020), Mandl and Minner (2023), Neghab et al. (2022) have strong prescriptive performance. At the same time, it is not clear whether a given method would always outperform the others for operational problems. We therefore use the data to train multiple methods and compare their

Table 3.1: A part of SECOM data: 1567 observations (i) for yield outcome (y_i) with 590 associated features ($u_{i,k}$) for each observation, https://archive.ics.uci.edu/ml/datasets/SECOM

Observation i	**Yield** (y_i)	$u_{i,1}$	$u_{i,2}$	$u_{i,3}$	$u_{i,4}$	**...**	$u_{i,589}$	$u_{i,590}$
1	1	3030.93	2564	2187.73	1411.12	...	*NaN*	*NaN*
2	1	3095.78	2465.14	2230.42	1463.66	...	0.006	208.20
3	0	2932.61	2559.94	2186.41	1698.01	...	0.0148	82.86
4	1	2988.72	2479.90	2199.03	909.79	...	0.0044	73.84
5	1	3032.24	2502.87	2233.36	1326.52	...	0.0044	73.84
6	1	2946.25	2432.84	2233.36	1326.52	...	0.0052	44.01
7	1	3030.27	2430.12	2230.42	1463.66	...	0.0052	44.01
8	1	3058.88	2690.15	2248.90	1004.46	...	0.0063	95.03
9	1	2967.68	2600.47	2248.90	1004.46	...	0.0045	111.65
⋮	⋮	⋮	⋮	⋮	⋮	...	⋮	⋮
1565	1	2978.81	2379.78	2206.30	1110.50	...	0.003	43.52
1566	1	2894.92	2532.01	2177.03	1183.73	...	0.008	93.49
1567	1	2944.92	2450.76	2195.44	2914.18	...	0.005	137.78

performance on an out-of-sample test set. The methods include Sample Average Approximation as in Kleywegt et al. (2001) that ignores the features, a method that exploits the theoretical solution in (3.2) while estimating the conditional probability distributions using ML-based methods and a class of methods that use a mathematical programming formulation for a specified class of feature-dependent decision rules (Ban and Rudin, 2019).

3.4.1 Joint Conditional Distribution Estimation (CDE) and Optimization

Typical ML approaches focus on point estimators for the predicted quantities. On the other hand, the optimization formulation in (3.1) and its solution (3.2) depend on

the distribution of demand and yield conditional on the features. If one can estimate the conditional distributions $D|\mathbf{X}$ and $Y|\mathbf{U}$ by the corresponding estimators $\hat{D}|\mathbf{X}$ and $\hat{Y}|\mathbf{U}$, then the optimization problem in (3.1) can be formulated and solved by using the condition (3.2) or equivalently by SAA (conditional on the features).

There are many plausible approaches for extracting feature-dependent distributional estimators for some commonly used non-linear ML-predictors. For instance, Ban and Rudin (2019) use kernel-based estimators where the relative weight of the neighbor within the kernel gives the probability mass for that neighbor.

We use a Random Forest (RF) approach as suggested in Bertsimas and Kallus (2020) for a prescriptive case. The reason we think that an RF-approach might perform well is that it is an ensemble method that generates many approximately independent estimators for a given set of features (Breiman, 2001). This generates high variance and low bias estimators which can then be averaged to reduce variance. We anticipate that using all independent estimators simultaneously rather than just a single point estimator to solve the stochastic optimization problem should lead to a better approximate solution to (3.2).

To this end, in order to obtain an estimator for the yield rate $Y|\mathbf{U}$, we generate a random forest of K randomized decision trees based on the training data. Any tree k generates an estimator for a new observation: $\hat{y}_{new,k}|\mathbf{U}_{new}$. This enables us to obtain K different approximately independent point estimators that can be weighted with a corresponding probability mass of $1/K$. We then repeat the same procedure to estimate the distribution of conditional demand $D|\mathbf{X}$. The estimators are depicted in Figure 3.4.1.

Proposition 2. *For an observation i from the test set let $(\hat{y}_{i,k}, \hat{d}_{i,k})$ be the estimators for yield and demand from tree k $(k = 1, 2, ..., K)$. Then, the optimal lot size is given by:*

$$Q_i^* = \min\left\{ Q : \frac{\sum_{k=1}^{K} \hat{y}_{i,k} \mathbb{1}_{\{\hat{d}_{i,k} \leq Q\hat{y}_{i,k}\}}}{\sum_{k=1}^{K} \hat{y}_{i,k}} \geq \frac{b}{b+h} \right\}. \tag{3.3}$$

Proof of Proposition 2: Assuming that the Random Forest generates K approximately independent point estimators for $\hat{Y}_i$ and $\hat{D}_i$, we can take the sample of point

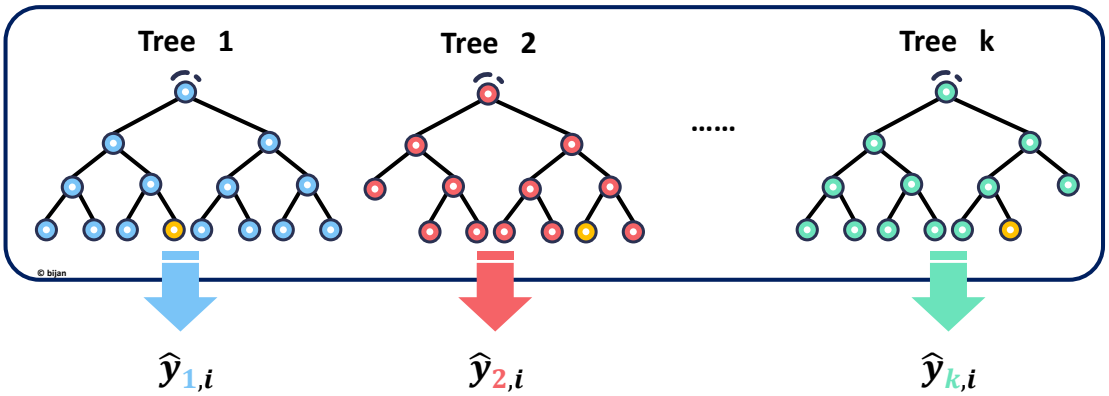

Figure 3.1: Random Forest Structure: for each k, $P(\hat{Y}_i = \hat{y}_{k,i}) = 1/k$

estimators to be the input sample to an SAA approach. The optimal solution on this sample is given by the above version of (3.12). □

We refer to the above CDE estimation/optimization approach through random forests *CDE-RF*. We note that the above approach could be applied similarly to other composite estimators, such as kernel-based ones as in Ban and Rudin (2019) with some modifications.

3.4.2 Empirical Risk Minimization with Features using Linear Rules

Next, we focus on some possible approaches to exploit available features in a plausible manner within the Empirical Risk Minimization (ERM) framework which aims to find feature-dependent decision rules using an optimization formulation for the training data.

Linear Rules for Demand and Yield

One attractive property of the SAA approach in subsection 3.4.3 is that the resulting optimization problem is also a linear program (as outlined in Appendix F). To maintain the LP formulation in (F.1)-(F.5), following a similar approach for a newsvendor problem with demand features by Ban and Rudin (2019), let us consider

lot size (ordering) rules that are feature dependent and linear as a function of the features. In particular,

$$Q_{LR,i} = \beta_0 + \beta_1 x_{i,1} + ... + \beta_r x_{i,r} + \beta_{r+1} u_{i,1} + ... + \beta_{r+s} u_{i,s} \tag{3.4}$$

As already noted in Ban and Rudin (2019), the linear lot sizing rule can be extended by taking non-linear transformations of the features. In Appendix G, we present a linear programming (LP) formulation that finds the optimal linear rule for the lot size as a function of the features. The solution of this LP uses a sample of training data (n observations) and yields the optimal values of the decision variables as: $\beta_0^*, \beta_1^*, ..., \beta_{r+s}^*$. The decision rule for a new out-of-sample observation is simply:

$$Q_{LR}(new) = \beta_0^* + \sum_{j=1}^{r} \beta_j^* x_{new,j} + \sum_{k=1}^{s} \beta_{r+k}^* u_{new,k}$$

where the coefficients β_0, β_1,..,β_{r+s} are learned from the training data through optimization.

We immediately note an important drawback with the above linear decision rule. The optimal lot size and the expected yield rate have a strongly non-linear relationship as observed in the optimality conditions (D.2) (from Appendix D) and (3.1). Indeed, the typical relationship between the optimal lot size and the expected yield rate is as depicted in Figure I.1. This is in contrast with the case for demand uncertainty where a linear model in terms of the features for demand $D = b_0 + b_1 x_1 + b_2 x_2 + .. + b_r x_r + \epsilon$ leads to an optimal lot size that is also linear in terms of the features when ϵ is normally distributed (Ban and Rudin, 2019). Based on this non-linearity that distinguishes the random yield case, in the rest of this subsection, we propose approaches that attempt to capture the non-linearity between the expected (conditional) yield and the optimal lot size. As a benchmark, for a partial version of the linear estimation and optimization formulation, we use a linear lot sizing rule for demand (if demand is random):

$$Q_i = \beta_0 + \sum_{j=1}^{r} \beta_j x_{i,j}$$

but ignore the yield features, thereby modeling uncertainty in the yield as feature-independent like in the SAA approach. The resulting LP formulation using the

training sample data as inputs is given in Appendix G, and we refer to the method as *ERM1* and the corresponding lot sizing rule for an out-of-sample observation is:

$$Q^*_{ERM1}(new) = \beta^*_0 + \sum_{j=1}^{r} \beta^*_j x_{new,j}.$$

Non-linear Yield Prediction Methods Combined with Linear Lot Sizing Rules

Recent advances in statistical learning have enabled powerful non-linear prediction methods. Our preliminary analysis of the SECOM yield data (McCann and Johnston, 2008b) confirms earlier reported results in Lee et al. (2019) that best performing yield classification/prediction methods for the data set are non-linear ones. For instance, for classifying the SECOM data for yield instance predictions, we find that random forest and gradient boosting classifiers are extremely effective. Encouraged by this finding, we propose to combine linear lot sizing rules with non-linear point estimators for yield.

To pursue the above reasoning, we combine a linear rule that finds an intermediate quantity T_i using the demand features assuming perfect yield:

$$T_i = \beta_0 + \sum_{j=1}^{r} \beta_j x_{i,j}$$

and combine it with a point estimator for the yield $\hat{y}_i$ to obtain a lot sizing rule:

$$Q_i = \frac{T_i}{\hat{y}_i}.$$

To execute the above, we learn the coefficients $(\beta^*_0, \beta^*_1, ..., \beta^*_k)$ using an LP similar to the one expressed in (G.2) to (G.5) (from Appendix G) using the training data. The LP model that minimizes the empirical risk on the training data is as follows:

$$\min \hat{z}(\beta_0, \beta_1, \ldots, \beta_{r+s}) = \frac{1}{n}\left(\sum_{i=1}^{n}(bo_i + hu_i)\right) \tag{3.5}$$

$$\text{s.t.} \quad u_i \geq d_i - Q_i y_i \qquad \forall i = 1, 2, \ldots, n \tag{3.6}$$

$$o_i \geq Q_i y_i - d_i y_i \qquad \forall i = 1, 2, \ldots, n \tag{3.7}$$

$$T_i = \beta_0 + \sum_{j=1}^{r} \beta_j x_{i,j} \qquad \forall i = 1, 2, \ldots, n \tag{3.8}$$

$$Q_i = \frac{T_i}{y_i} \qquad \forall i = 1, 2, \ldots, n \tag{3.9}$$

$$\beta_j \text{ unrestricted} \qquad \forall j = 1, 2, \ldots, n \tag{3.10}$$

$$u_i, o_i, T_i, Q_i \geq 0 \qquad \forall i = 1, 2, \ldots, n$$

Note that T_i in (3.8) can be interpreted as the optimal lot size for the case of a perfect yield. We then perform the adjustment $Q_i = T_i/y_i$ in (3.9). The solution of the above LP gives the optimal values of the coefficients: $\beta_0^*, \beta_1^*, ..., \beta_{r+s}^*$. The rule to compute the lot size for an out-of-sample observation is then:

$$Q_{NL}^*(new) = \left(\beta_0^* + \sum_{j=1}^{r} \beta_j^* x_{new,j}\right) / \hat{y}(new) \tag{3.11}$$

where $\hat{y}(new)$ is a point prediction for the yield rate of the new observation.

For the above approach, the accuracy of the point prediction for the yield rate is critical. We implement and test three classes of non-linear prediction methods to compute $\hat{y}(new)$. The first method uses a random forest predictor (using the library from scikit-learn (Pedregosa et al., 2011)) where a large number of decision trees are fit to the training data and the final point prediction can be obtained by averaging the prediction from all of the trees. We refer to this implementation of the rule in (3.11) as NLRF. The second method uses a gradient boosting regressor (using the library from scikit-learn (Pedregosa et al., 2011)) where a sequence of decision trees is fit to the training data, with each tree correcting the errors of the previous ones, resulting in an ensemble model. The final point prediction is obtained by aggregating the predictions from all the trees in the sequence. We refer to this implementation of the rule in (3.11) as NLGB. In the third method, we implement

a Multi-layered (i.e. Deep) Neural Network (using the Keras library (Chollet et al., 2015)) consisting of an input layer, three hidden layers, and an output layer. We refer to the implementation of the rule in (3.11) based on the point prediction from the neural net as: NLNN.

3.4.3 Benchmark: The Sample Average Approximation Solution

The methods in subsections 3.4.1 and 3.4.2 exploit the dependence of yield using the feature information. As an alternative, one can ignore the features and use only a sample of n demand and yield realizations $(d_i,\ y_i)$ as the training data. In this case, a plausible approach for optimization is to use Sample Average Approximation (SAA) (see (Kleywegt et al., 2001)) and solve the following problem:

$$\min_{Q} \tilde{z}(Q) = \frac{\sum_{i=1}^{n} \left(b(d_i - Qy_i)^+ + h(Qy_i - d_i)^+\right)}{n} \tag{3.12}$$

This is still a convex optimization problem, and one can find a sample-based solution using a modified version of the optimality condition (D.2) (from Appendix D) (Okyay et al., 2014):

$$Q^* = \min\left\{Q : \frac{\sum_{i=1}^{n} y_i \mathbb{1}_{\{d_i \leq Qy_i\}}}{\sum_{i=1}^{n} y_i} \geq \frac{b}{b+h}\right\} \tag{3.13}$$

Alternatively, the sample optimization problem can be expressed as a linear program (LP). We present the LP formulation in Appendix C. We use the SAA as a benchmark to assess the value of using feature information in our numerical experiments.

3.4.4 Discussion

The prescriptive analytics literature in operations applications is rapidly growing. The literature and the theory on Sample Average Approximation methods is well established (Kleywegt et al., 2001). In the standard SAA setting there are no features and independence assumptions are required. Most single-stage optimization problems under uncertainty where the decisions affect uncertainties are naturally modeled as stochastic programs (Bertsimas and Kallus, 2020). Sometimes a particular decision rule can be embedded in the stochastic program directly as in Ban and

Rudin (2019) or Liu et al. (2022). Our ERM method follows that approach. However, that has limitations for the random yield problem which we attempt to correct in the non-linear modifications NLRF, NLGB and NLNN. The CDE-RF method is different because it exploits the idea that the solution to the sample optimization problem with features can be characterized if distributional estimators are available. There is some evidence that RF-based methods perform really well for predictions (Hastie et al., 2009) and some more limited evidence in prescriptions. There is also additional evidence that methods that combine theoretical optimality conditions with conditional distributions perform well (Ban and Rudin, 2019). CDE-RF may be considered as a combination of the two ideas.

In the next section, we assess the performance of the methods thoroughly on multiple data sets since it is difficult to anticipate in advance which method would perform the best. A significant issue is that a large number of features may lead to overfitting for the proposed methods. Some methods (such as random forests) are immune to overfitting by design but for others we propose using feature elimination based on predictive performance in the pre-processing stage. This will be further investigated in the numerical study.

3.5 Key Performance Indicators and Performance Measurement

In this section, we introduce three key performance indicators (KPIs) for our proposed data-driven models, emphasizing the importance of utilizing features and information to address optimal lot sizing problems effectively.

3.5.1 Value of Feature Information

In this study, we implement and compare six different models to determine the optimal lot sizes by integrating optimization and machine learning tools. The main approach involves extracting a lot sizing decision rule from the training set and then evaluating its cost performance by applying this decision rule to an out-of-sample set. The primary KPI of this study is the comparison of the models based on the average cost in the test set and measuring the percentage of cost reduction (saving)

compared to the best-performing model.

3.5.2 Prescription Error (PE)

Yield and demand prediction naturally introduce errors in the optimization stage. In order to measure the effect of such errors in planning, we assume that there is access to perfect information about the yield and solve the planning problem under perfect yield information. We use the Prescription Error (PE) metric (similarly as in Mandl and Minner (2023)) which represents the percentage increase in cost without perfect information relative to the cost that can be achieved under perfect yield information. Since the CDE-RF model appears to outperform the other models in terms of cost performance, we evaluate this KPI for this model. We assume that in case of perfect yield information (labeled as PI), for all observations i, we have access to the true value of the yield y_i rather than its estimator $\hat{y}_i$. The optimality condition (3.3) then becomes:

$$Q_i^{PI*} = \min\left\{Q : \frac{\sum_{k=1}^{K} \mathbb{1}_{\{\hat{d}_{i,k} \leq Q y_i\}}}{K} \geq \frac{b}{b+h}\right\}. \tag{3.14}$$

Let us denote by $\hat{C}$ and C^{PI} as the realized average cost respectively without and with perfect yield information. PE is then defined as the percentage cost increase due to using imperfect information:

$$PE \equiv \left(\frac{\hat{C} - C^{PI}}{C^{PI}}\right) 100. \tag{3.15}$$

3.5.3 Value of Integrating Estimation and Optimization

Machine learning based prediction tools can readily be used to provide point predictions for the conditional means of demand $\hat{d}(\mathbf{x})$ and yield $\hat{y}(\mathbf{u})$ provided that training, model reduction, and feature selection issues can be resolved from data. It is therefore relatively easy to reach the following approximate solution using predictive ML methods:

$$Q^*(\mathbf{x}, \mathbf{u}) = \hat{d}(\mathbf{x})/\hat{y}(\mathbf{u}). \tag{3.16}$$

This would be a naive solution to the problem unless the predictions are extremely accurate. In most realistic cases, the conditional point estimators $\hat{d}(\mathbf{x})$ and $\hat{y}(\mathbf{u})$ have considerable uncertainty, and this motivates using the full characterization in (3.2) based on the conditional distributions of the estimators.

To demonstrate the value of integrating estimation and optimization, we first estimate the lot sizes using (3.16) and calculate the corresponding average costs, denoted as C^{EST}. We then find lot sizes using (3.2), referred to as CDE-RF, and calculate the corresponding costs on the test set, denoted as $\hat{C}$. To assess the value of integrating estimation and optimization (VIEO), we use the following performance measure as proposed by Mandl and Minner (2023):

$$VIEO \equiv \left(\frac{C^{EST} - \hat{C}}{\hat{C}} \right) 100. \tag{3.17}$$

3.6 Numerical Results

In this section, we assess the performance of data-driven solutions to the lot sizing problem presented in Sections 3.3 and 3.4. Multiple methods are compared on the modified SECOM data and the synthetic data sets.

We assess performance and KPI's of the proposed methods on two different types of data sets: some synthetic data sets and a modified version of the SECOM data set, which is publicly available (McCann and Johnston, 2008b) (and partially presented in Table 3.1). We present details on the synthetic data sets and the modified SECOM data in Appendix H. We also present data cleaning and processing issues and some feature elimination considerations in the same section.

To assess and compare performances of the implemented methods, we use a common approach where the data set is randomly split into separate training and test sets with an 80%-20% ratio. All methods are trained on the same training set and performance is evaluated on the test set. In the training set, there are 1404 (SECOM) and 1600 (synthetic) randomly selected observations while in the test set there are 351 (SECOM) and 400 (synthetic) observations. A summary of the procedure of the proposed methods in this study is shown in Figure 3.6. The SAA

benchmark does not use any feature information. ERM, on the other hand, uses the demand features but not the yield features. The other methods use the information in different ways. CDE-RF uses conditional distribution estimators obtained from random forests for both demand and yield.

We use the Gurobi solver (Gurobi Optimization, LLC, 2024) for constrained optimization and perform all of the computations in Python.

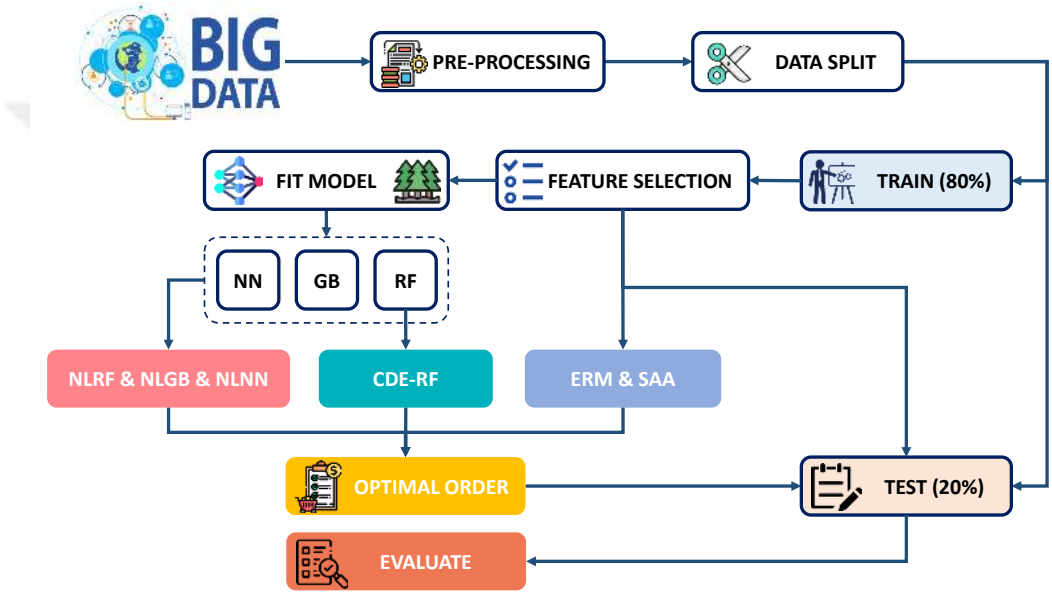

Figure 3.2: Procedure of running proposed strategies

3.6.1 Detailed Results on Cost Performance of Methods

In this subsection, we focus on an instance of the problem where the financial parameters are fixed to $h = 10$ and $b = 15$. For this particular case of costs, we present some detailed results on performance and on the decisions taken by the different methods proposed.

To compute the optimal lot size for SAA, we find the optimal quantity in the training sample by computing (3.13). For the CDE-RF method, we train the random forest using the training data. This leads to a number of individual estimators for each observation in the test set. We then use Proposition 2 to compute the optimal

lot size. For ERM1, NLRF, NLGB, and NLNN, we solve the corresponding linear programs using the training data set as input.

Below, we describe some of the lot sizing decision rules based on the training data.

Linear Lot Size Decision Rules Based on Training Data

First, SAA gives a lot size that does not depend on the features. In the synthetic data set, the mean demand is approximately 172 units, with a standard deviation of 67 units, and the mean yield is 0.71, accompanied by a standard deviation of 0.14. In the SECOM data set, the mean demand is approximately 40 units, with a standard deviation of 16 units, and the mean yield stands at 0.82, with a standard deviation of 0.28. The lot sizes for the synthetic data set and the SECOM data set are found as:

$$Q^*_{SAA-SYN} = 289.86 \text{ and } Q^*_{SAA-SECOM} = 52.08$$

Second, we implement ERM solving (G.6) to (G.9) (from Appendix G). In the synthetic data set, there are five relevant demand features; within this data set, the average values of features are approximately 5, 5, 5.1, 5.1, and 4.9 units, respectively. The optimal lot size can be calculated based on equation (3.18) as follows:

$$Q^*_{i-SYN} = -0.83 + 13.83x_{i,1} - 2.80x_{i,2} + 18.16x_{i,3} + 7.30x_{i,4} + 1.41x_{i,5} \tag{3.18}$$

In the SECOM data set, there are four demand features, and the average values of features are 8.1, 16.4, 3, and 11.5 units, respectively. The optimal lot size can be calculated based on equation (3.19) as follows:

$$Q^*_{i-SECOM} = -5.19 + 9.32x_{i,1} - 0.77x_{i,2} - 3.44x_{i,3} + 0.07x_{i,4} \tag{3.19}$$

Third, for NLRF, NLGB, and NLNN, we utilize random forest, gradient boosting, and neural net estimators for obtaining a point prediction $\hat{y}_i$ for the yield rate, which is then inserted into (3.11). In the synthetic data set, there are five relevant demand features, the optimal lot size can be calculated based on equations (3.20) and (3.22)

if the yield estimated by RF, GB, and ANN respectively as follows:

$$Q^*_{i,\text{NLRF-SYN}} = (4.79 + 15.42x_{i,1} - 2.47x_{i,2} + 26.68x_{i,3} + 7.15x_{i,4} + 0.39x_{i,5})/\hat{y}_{i,RF} \quad (3.20)$$

$$Q^*_{i,\text{NLGB-SYN}} = (5.63 + 15.59x_{i,1} - 2.61x_{i,2} + 26.08x_{i,3} + 7.48x_{i,4} + 0.17x_{i,5})/\hat{y}_{i,GB} \quad (3.21)$$

$$Q^*_{i,\text{NLNN-SYN}} = (4.66 + 15.18x_{i,1} - 2.55x_{i,2} + 26.63x_{i,3} + 6.98x_{i,4} + 0.12x_{i,5})/\hat{y}_{i,NN} \quad (3.22)$$

In the SECOM data set, there are four demand features, the optimal lot size can be calculated based on equations (3.23) and (3.25) as follows:

$$Q^*_{i,\text{NLRF-SECOM}} = (46.04 + 3.32x_{i,1} - 0.83x_{i,2} - 3.41x_{i,3} + 0.06x_{i,4})/\hat{y}_{i,RF} \quad (3.23)$$

$$Q^*_{i,\text{NLGB-SECOM}} = (23.78 + 5.78x_{i,1} - 0.79x_{i,2} - 3.21x_{i,3} + 0.05x_{i,4})/\hat{y}_{i,GB} \quad (3.24)$$

$$Q^*_{i,\text{NLNN-SECOM}} = (30.77 + 4.94x_{i,1} - 0.87x_{i,2} - 3.53x_{i,3} + 0.12x_{i,4})/\hat{y}_{i,NN} \quad (3.25)$$

Please note that, unlike the previous rules, (3.20) - (3.25) use a separate yield predictor $\hat{y}_i$ whose rule is obtained from the training data and is applied to the test data.

Cost Performance

We take the decision rules from the previous subsection and implement them on the test part of each data set data and compute cost performance for $h = 10$ and $b = 15$. We report the dispersion of cost on the test set in the box plots format (including min, max, Quantile 1, Quantile 3 and average cost). Figure 3.3 reports the results for the synthetic and the SECOM data sets.

For this instance of costs, the results shown in Figure 3.3 are consistent. Feature-based methods compare significantly better than SAA which does not use features in terms of average cost. Among feature-based methods, CDE has the best average performance by a margin, whereas the other three methods have similar performance. For the SECOM data, NLRF, NLGB, and NLNN seem to perform somewhat better than ERM.

In terms of dispersion, we note that CDE-RF has excellent performance on the synthetic data (Fig. 3.3 - left side). It also has very good dispersion performance

for most of the data in the SECOM data set (Fig. 3.3 - right side). We can observe, however, that it has a few extreme instances for the SECOM data.

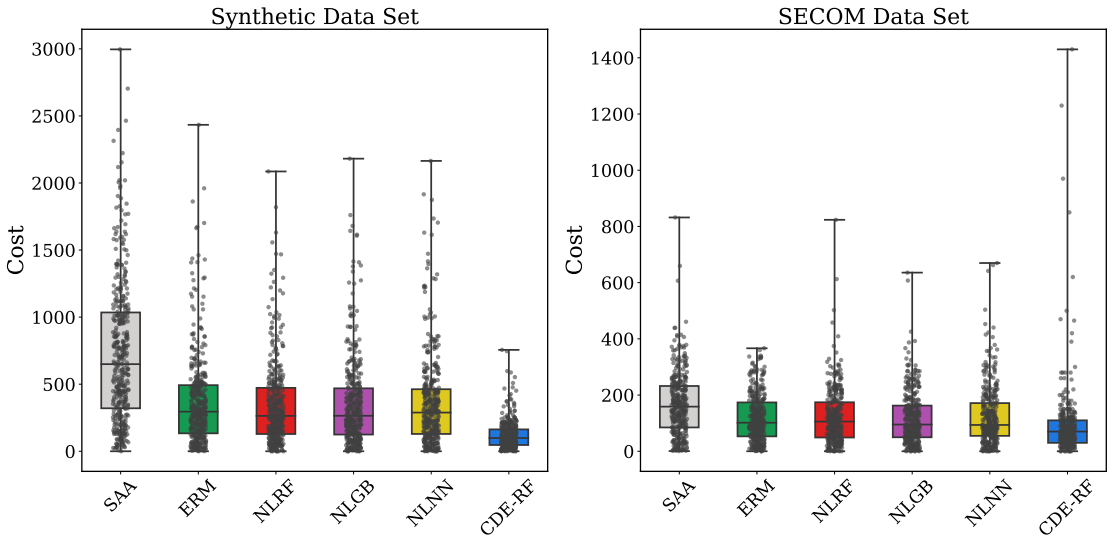

Figure 3.3: Dispersion of costs in the test sample for the synthetic and SECOM data sets ($h = 10$, $b = 15$)

Let's define the percentage loss (PL) of any method with respect to CDE-RF as:

$$PL \equiv \left(\frac{C(\text{method}) - C(CDE - RF)}{C(CDE - RF)} \right) 100. \tag{3.26}$$

Table 3.2 presents a summary of the results for generated costs for each method in terms of PL and we also report the required run time. We observe that for the synthetic data set the costs are decreased by a factor of five when features are used using the CDE-RF method. For the SECOM data, the corresponding decrease is relatively smaller but still very significant. In both data sets, however, the CDE-RF method generates the lowest cost. We note that it needs a higher run time to find the optimal lot size and corresponding cost with respect to the other methods. We note that the CDE-RF method has a longer running time than the other methods

since multiple trees are generated. However, the overall running times are fairly small in the data sets used.

Table 3.2: Performance Comparisons on the Test Set (PL with respect to CDE-RF)

Method	Synthetic Data Set			SECOM Data Set		
	Average Cost	PL (%)	Run Time (h:m:s)	Average Cost	PL (%)	Run Time (h:m:s)
SAA	735	+507.44%	0:00:00.25	167	+83.52%	0:00:02.70
ERM	370	+205.79%	0:00:29.66	137	+50.55%	0:00:33.95
NLRF	346	+185.95%	0:00:01.74	123	+35.16%	0:00:03.50
NLGB	353	+191.73%	0:00:00.56	117	+29.61%	0:00:03.50
NLNN	366	+202.48%	0:00:15.35	125	+37.36%	0:00:14.70
CDE-RF	121	-	0:02:35.61	91	-	0:00:45.90

To validate the effectiveness of the proposed models and their influence on overall costs, we provide additional results on ten other synthetic data sets. These data sets are characterized by distinct linear distributions in demand and yield, encompassing five features each and 3000 observations. As before, we fit the models on the training data and report results on the test data. The findings, as depicted in Figure 3.4, reinforce previous observations. The approaches, which consider features such as CDE-RF, NLRF, NLGB, NLNN, and ERM, consistently outperform the SAA model which does not take these features into account. This highlights the critical role of feature inclusion in cost reduction. As was already noted, among all methods, CDE-RF leads to the most significant cost reductions, while ERM, NLRF, NLGB, and NLNN exhibit comparable cost performance in the synthetic data sets.

A Comparison of Lot Sizing Decisions

Here, we present a brief summary of the lot sizing decision prescribed by each method for the observations in the test set (for $h = 10$ and $b = 15$). The results are presented in Figures 3.5 and 3.6. For these figures, the test data is sorted in increasing order of realized demand. The black curve corresponds to the realized demand after sorting. We note that the cost in (3.1) depends on the deviation of

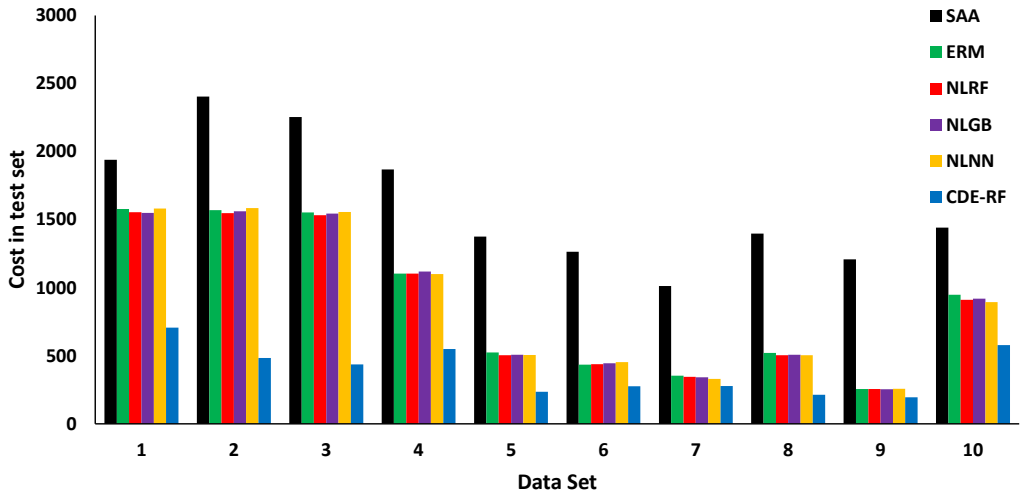

Figure 3.4: Average costs in the test set for additional 10 synthetic data sets ($h = 10$, $b = 15$)

the number of good items Q_iY_i and the corresponding demand D_i. In the figures, the dots around the black curve correspond to the number of good items Q_iY_i for the corresponding observation. The red (blue) dots signify observations where the number of good items Q_iY_i is higher (lower) than D_i. Any rule that generates lower deviations around the demand results in lower costs.

We observe in Figure 3.5 that for the synthetic data set, the lot sizes obtained from ERM and NLRF lead to similar deviations whereas CDE-RF leads to visibly smaller deviations around the demand which explains its better cost performance. The situation is similar for the SECOM data as depicted in Figure 3.6. Once again ERM and NLRF lead to similar levels of deviations around the realized demand whereas CDE-RF leads to lower deviations, especially when $QY > D$. We should also note that the complexity of the SECOM data set leads to significant outliers in terms of $|Q_iy_i - d_i|$ for a few observations in the test set.

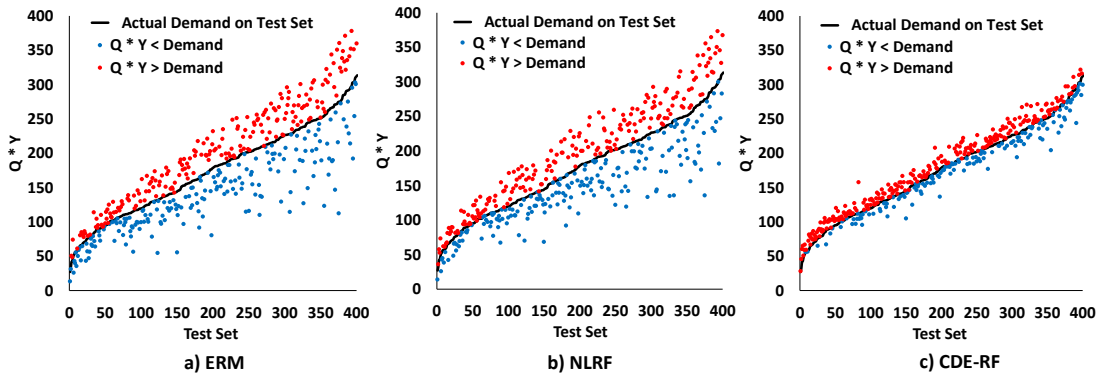

Figure 3.5: Optimal lot sizes for the observations in the test set for the synthetic data set ($h = 10$, $b = 15$)

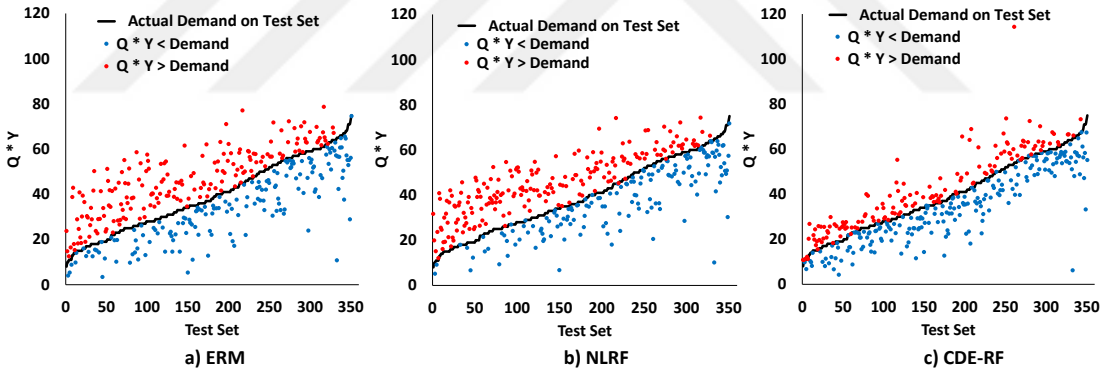

Figure 3.6: Optimal lot sizes for the observations in the test set for the SECOM data ($h = 10$, $b = 15$)

3.6.2 Assessment of Cost Performance for Different Demand Models and Financial Parameters

Our previous analyses focus on a specific case of the financial parameters where $h = 10$ and $b = 15$. It is clear from (D.1) and (D.2) (from Appendix D) that the

ordering rule and the corresponding expected cost depend on the ratio $b/(b+h)$ significantly. Moreover, in the motivating example for this study, the contracted demand quantity is known in advance. In this subsection, we report the results on cost performance where the demand model and the critical fraction $b/(b+h)$ vary.

Constant Demand and Random Yield

While our analysis covers the case with both random demand and yield, one motivating case for our model is one where the demand was contracted and known in advance. In this subsection, we present results for this case. We assume that the contract targets for the synthetic and SECOM data set are $d_{SYN} = 150$ and $d_{SECOM} = 40$ units, respectively. We compare the performance of the proposed methods using the synthetic yield data as well as the SECOM yield data when the critical ratio $b/(b+h)$ changes.

Results for the Synthetic Data Set

We use the synthetic yield data from and train all methods using the training data and report results based on the test data. Figure 3.7 depicts that the CDE-RF method performs much better than all other methods in terms of the average cost in this case. Its average lot size is somewhat higher than all other methods when the critical ratio is less than 0.7 and lower than other methods for ratios above 0.7.

Results for the SECOM Data Set

We now take the SECOM yield data and repeat the same procedure as in Section 3.6.2: we train all methods using the training data and report the cost performance on the test data.

As shown in Figure 3.8, in terms of cost performance, the CDE-RF performs significantly better than other methods when the critical ratio is above 0.6. For all methods, when the critical ratio increases, the average lot size increases as expected. We remark that when the ratio is above 0.6, the CDE-RF method chooses lot sizes that are higher on average.

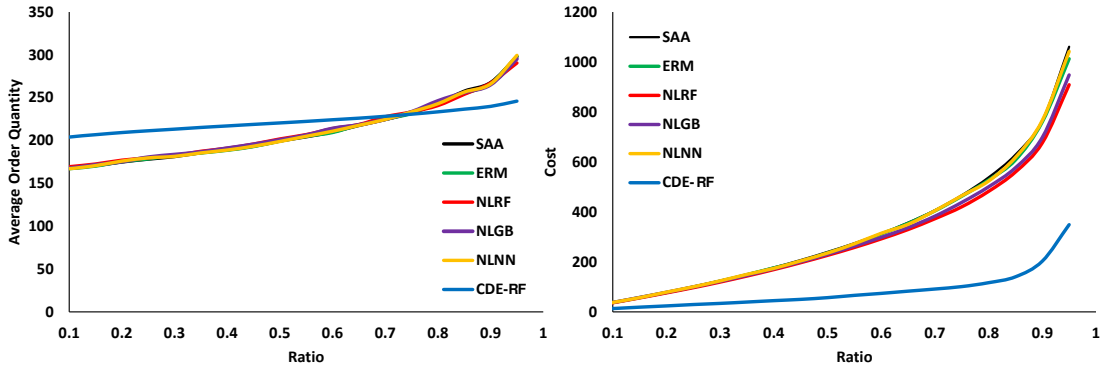

Figure 3.7: Expected lot sizes and expected cost (test set) for the synthetic data set under constant demand

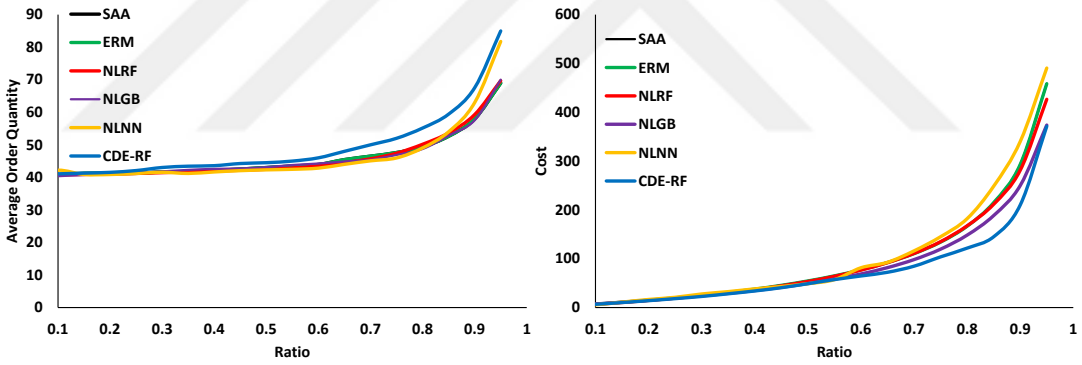

Figure 3.8: Average lot sizes and average cost (test set) for SECOM data set under constant demand

Random Demand and Random Yield

We now repeat the previous analysis in for the case of random demand and yield for the synthetic data set and the SECOM data set to as a function of the ratio: $b/(b+h)$.

Results for the Synthetic Data Set

As before, we train all methods on training data and present results for the test data. Figure 3.9 displays that CDE-RF performs the best in terms of cost by a wide margin. With respect to Figure 3.7, we observe this time that NLRF, NLGB, and NLNN perform significantly better than SAA. This may be a sign that both methods are capturing the demand feature data correctly. We remark that all methods perform significantly better than SAA, which does not use any feature information.

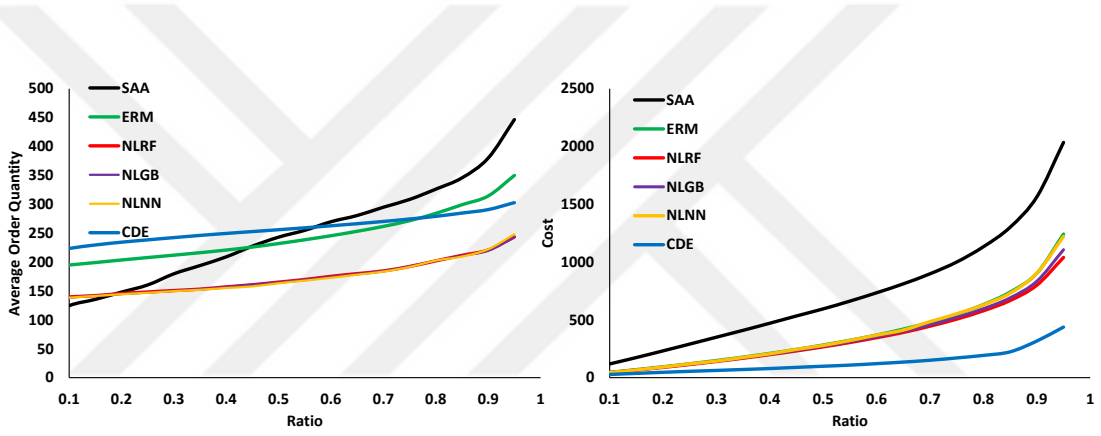

Figure 3.9: Average lot sizes and average cost (test set) for the synthetic data set under random demand

Results for the SECOM data set

This time we use the synthetic demand data combined with the SECOM yield data and report results on the test data set as before. The test set performance is depicted in Figure 3.10. In comparison to Figure 3.9, the methods that use feature information perform better than SAA but the relative improvement is smaller than before. In fact, SAA performs slightly better than ERM for high critical ratios. CDE-RF is still the best performer in terms of cost among all methods. The decrease

in the relative gain is understandable since the SECOM yield data is significantly more complicated than our synthetic yield data. Nevertheless, CDE-RF still presents significant gains with respect to SAA, except for extremely high critical ratios (where it still performs better)

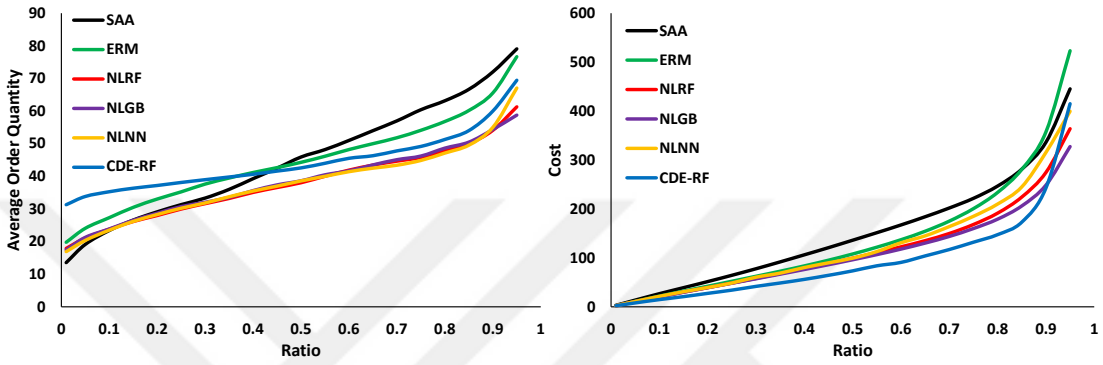

Figure 3.10: Average lot sizes and average cost (test set) for the SECOM data set under random demand

Summary: Performance Comparison

Let us consider the case of constant demand that was explored above to present a relative performance table for the methods. We report the performance with synthetic yield data and the SECOM yield data separately. The CDE-RF method consistently generated lower costs, therefore we take its cost performance as the benchmark and report the relative percentage cost increase of the other methods for four different values of critical ratios: $(b/(b+h) = 0.2, 0.4, 0.6, 0.8)$. The summary report can be found in Table 3.3.

In Table 3.3, we see that the gains in cost performance are very significant when demand and yield features are exploited in data-based methods. CDE-RF is the clear best performer and it brings potentially huge benefits over SAA even for a challenging yield data set like SECOM. For the synthetic data set, the gains are even

Table 3.3: PL (test set) in cost for synthetic (GEN) and SECOM data sets under different ratios with respect to CDE-RF (ratio=$b/(b+h)$)

Ratio →	**0.2**		**0.4**		**0.6**		**0.8**	
Method ↓	**GEN.**	**SECOM**	**GEN.**	**SECOM**	**GEN.**	**SECOM**	**GEN.**	**SECOM**
SAA	399%	87%	496%	90%	505%	84%	488%	67%
ERM	103%	53%	166%	49%	204%	51%	228%	58%
NLRF	91%	42%	152%	39%	185%	35%	200%	30%
NLGB	96%	42%	157%	36%	191%	30%	210%	21%
NLNN	99%	43%	161%	43%	203%	44%	227%	42%
CDE-RF	-	-	-	-	-	-	-	-

larger. This implies that there is additional value in feature-based methods when the relationship between the features and yield is stronger (and can be uncovered by linear or non-linear models). Among the other methods, non-linear modifications of linear rules (NLRF, NLGB, NLDL) have very similar performance and lead to savings with respect to the linear ERM rule. We also observe that all methods that consider features perform significantly better than the SAA that ignores features. The stronger performance of the CDE-RF approach on the synthetic data may be dependent on the generation of the synthetic data but nevertheless it also performs significantly better than the other approaches on the SECOM data.

3.6.3 Prescription Error (PE) Performance

We measure the PE values for different training sample sizes and critical ratios for both synthetic and SECOM data sets using the CDE-RF method. As shown in Figure 3.11, for both cases, by increasing the training sample size (left to right direction), the PE values decrease, while increasing the critical ratios (up to down direction), the PE values increase. The PE values range between 0% and 36% for the synthetic data set and between 2% and 122% for the SECOM data set. We observe that using a medium to large amount of training data very low PE's can be achieved if the critical ratio is not high.

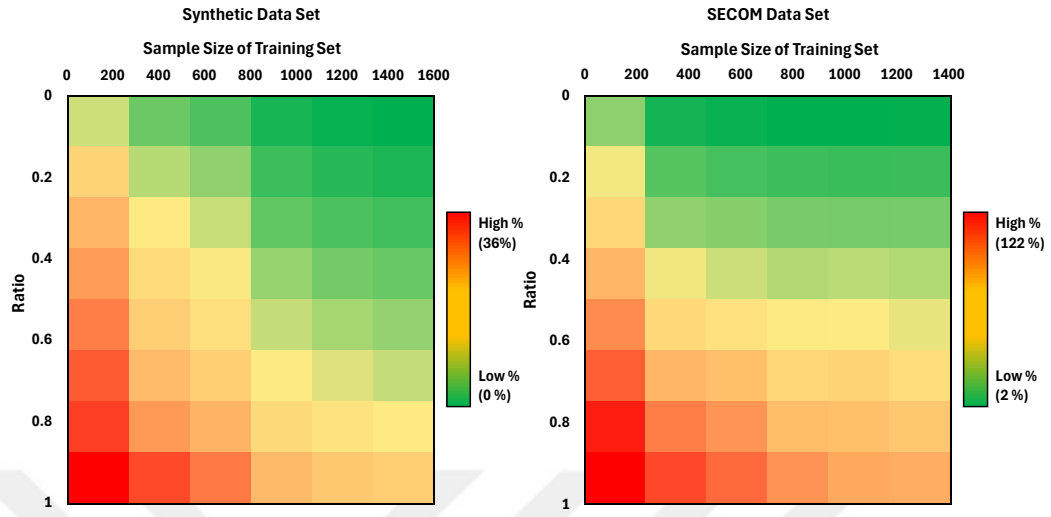

Figure 3.11: Impacts of Training Set Size and Critical Ratio on PE (%) in the Test Set

3.6.4 Value of Integrating Estimation and Optimization (VIEO)

In this subsection, we analyze the impacts of varying critical ratios on VIEO values using two data sets: SECOM and a synthetic data set. As illustrated in Figure 3.12, for both data sets VIEO is significant which confirms that there is value in combining estimation and optimization stages. Second, the VIEOs as a function of the critical ratio exhibit a similar pattern for both data sets. Initially, as the critical ratio increases from 0 to approximately 0.55, VIEO decreases, with a sharp decline observed for ratios up to 0.3. However, beyond 0.55, the VIEO values increase significantly.

The overall cost associated with estimated lot sizes by (3.16) is highly dependent on the quality of the fitted models for demand and yield. Generally, fitted models with higher R^2 values result in lower underage and overage costs. In this section, we measure the VIEO by fitting demand and yield models with R^2 values ranging from 70% to 80%.

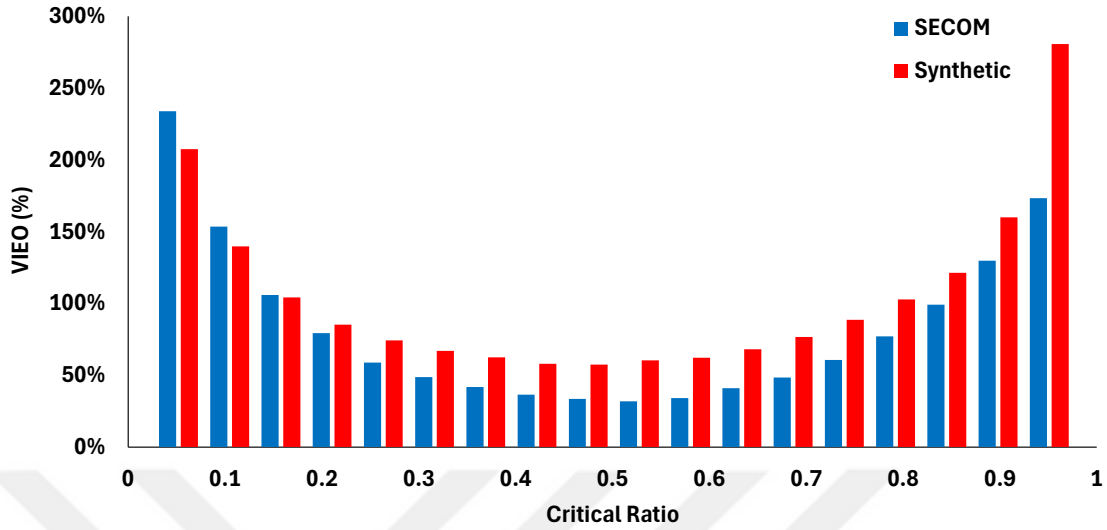

Figure 3.12: Impacts of Critical Ratio $b/(h+b)$ on VIEO (%) in the Test Set

3.6.5 Impacts of Training Sample Size and Number of Features on the Expected Cost

In this subsection, we analyze the impacts of changing the training set sample size and the number of features on the achieved average cost on the identical test set using the SECOM data. We train the model using different training set sizes. assuming a fixed critical ratio of $b/(h+b) = 0.6$.

We employ two models. **First**, the CDE-RF model is used to measure performance since it is the best performing one among the methods considered. **Second**, we utilize the linear ERM rule to assess the impact of features and potential overfitting. We vary the training sample size from 10 to 1404. To experiment with different number of features. We first sort the features in terms of their predictive performance by using a random forest regression we then use the first l features according to predictive importance and vary l from 1 to 590, running both the CDE-RF and linear models on the training set. The average cost achieved by each model for each scenario is then calculated.

As illustrated in Figure 3.13, the average cost in the test set decreases rapidly when the training sample size increases for small sample sizes (i.e. less than 50 observations). The relative decrease in the cost due to the increase in the sample size is smaller once sample sizes become larger (i.e. 500 above). The average cost range between 170 and 17,116 for the Linear method and between 90 and 206 for the CDE-RF method.

This is very promising since the SECOM data set is fairly large and requires pre-processing and further model reduction checks for feature selection and the final model determination. However, the impact of the number of features is more complex. In the right figure, the optimal lot sizes are derived from the conditional distribution extracted by the random forest. Because the random forest is an ensemble method which uses many small trees but each time selecting the most significant features, increasing the number of features/trees does not lead to significant overfitting in predictions most of the time (Hastie et al., 2009). We observe that this property carries over to the cost performance. This is evident in the green region (low cost) in the right image, even when using the full observation set and all features. In contrast, the left figure depicts that the cost performance of linear rules like ERM is not robust to the number of features. In this case, increasing the number of features initially decreases the cost until it reaches a peak when the number of features is between 20 and 45. Beyond this point, the cost increases rapidly due to overfitting and model complexity. Thus, in this case, the lowest cost occurs when using the full observation set but limiting the number of features to around 20 to 45.

3.6.6 Performance Comparison with a Theoretical Benchmark

In the previous subsections, we checked the performance of using feature-based optimization with a large number of features where the relationship between features and uncertain responses were inferred from data. As a final check, we explore the model in subsection 3.3.2 where there are some binary features and the relationship between the features and the random yield is known.

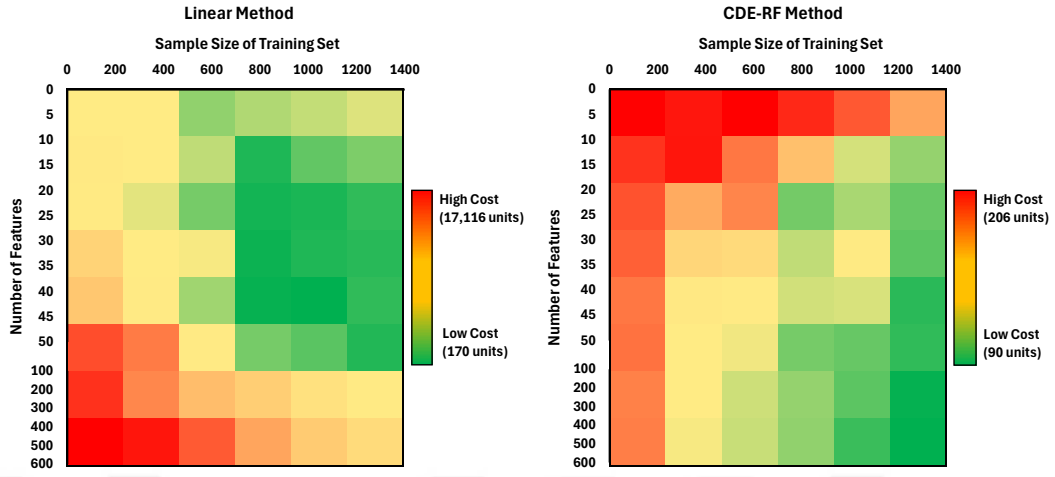

Figure 3.13: Impacts of Training Set Size and Number of Features on Cost Reduction on the Test Set

Let's assume that there are 10 binary features $u_i \in \{0, 1\}$, $(i = 1, 2, ..., 10)$ observable before the lot sizing decision is made. To test whether overfitting might be an issue for ML-based methods with a large number of features, let us further assume that yield Y only depends on the first two features u_1 and u_2. In particular, we assume that the $Y|\mathbf{U}$ has a uniform distribution specified as follows:

$$Y|\mathbf{U} = (u_1, u_2, ..., u_{10}) \sim \text{ Uniform } ((2u_1 + u_2)/4, (2u_1 + u_2 + 1)/4)$$

Assuming that demand d is a constant, we can compute $Q^*|\mathbf{U}$ and $E[c(Q^*|\mathbf{U})]$ explicitly and given a probability distribution for the features, we can also compute the expected optimal cost per period as a complicated but explicit expression. The details are given in Appendix J.

In order to compare the performance of data-driven methods proposed in this study against the theoretical benchmark, we generate a simulated data set assuming that $P(U_1 = u_1, U_2 = u_2) = 1/4$ for $u_1, u_2 \in \{0, 1\}$. The other binary features (3 to 10) are generated randomly. The theoretical benchmark uses the optimal feature-based lot size for each observation (from Appendix J) whereas the data-driven methods learns a feature-based ordering rule from the training data and

applies the rule for each observation in the test set.

We employ the CDE-RF method in a simpler way for this example. Since the number of features is low and their effect on yield is clear, the RF method is always able to identify the identical correct tree. We therefore take a single decision tree instead of multiple trees and consider all observations from the training data that fall under a specific leaf to estimate the conditional distribution.

Figure 3.14 summarizes the generated costs for SAA, CDE-RF, and the theoretical benchmark under three different critical ratio levels. In this analysis, the generated data set is divided into a training set with varying sample sizes and a fixed test set size of 1000 observations, which is then used to evaluate the cost performance of each approach. As shown in the figure, the costs generated by SAA are less sensitive to training sample size, as they do not rely on features to determine the lot sizing rule. In contrast, CDE-RF is more sensitive to sample size, as it learns from a limited training data set that includes irrelevant features. The theoretical benchmark does not depend on the training sample. We note, as before, that the data-driven method taking into account the features (CDE-RF) demonstrates significantly better performance compared to SAA, which serves as a reasonable upper bound for average cost. More importantly, the CDE-RF method performs competitively approaching the theoretical lower bound achieved by the model with full information even for small training sample sizes.

3.7 Conclusion

We present and compare data-driven methods to compute the optimal lot size of a manufacturing process subject to yield and demand uncertainty that depends on many observable features. We use tools from machine learning for prediction based on a large number of features that are potentially associated with the yield rate or demand. We then combine the predictions with the expected cost minimization framework to learn feature-dependent rules to determine the optimal lot size in a training data set. There are many plausible ways to do this and we experiment with state-of-the-art prediction methods (random forests, gradient boosting, deep

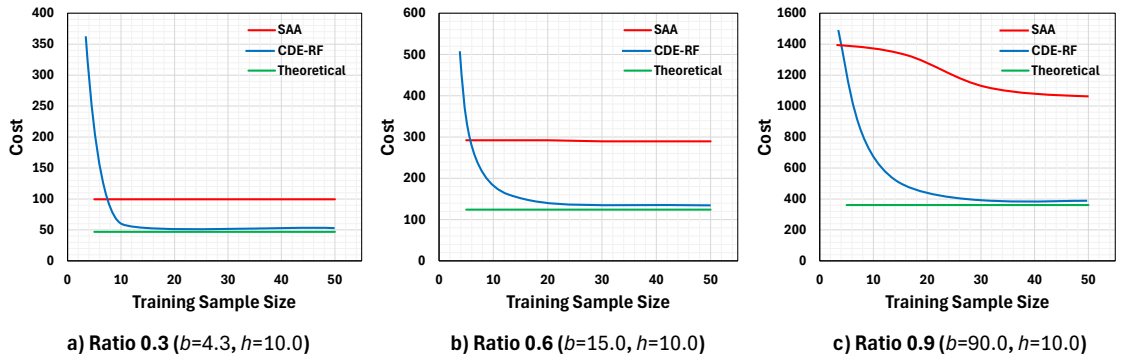

Figure 3.14: Cost comparison of CDE-RF and SAA versus Theoretical benchmark under different critical ratios

neural nets) and different optimization frameworks to compare and contrast the performance. For consistency, all methods are trained on the same training data and performance is measured and compared on the same test data.

Part of our performance analysis is based on a publicly available data set that presents many challenges including a large number of potential features. We demonstrate that the combined machine learning-optimization approach is able to handle such realistic sized data sets and leads to significant cost improvements over a base case that does not use the feature information.

The best performing method among the several that we tested uses a combination of theoretical knowledge on the optimization properties of the problem and conditional distribution estimations coming from a random forest algorithm. The method performs well with respect to multiple KPI's and seems to have good overfitting properties. This is a promising finding that underlines the value of theoretical knowledge on top of strong estimation algorithms for prescriptive analytics.

In a different prescriptive analytics model, (Bertsimas and Kallus, 2020) observe that RF-based optimization methods are the best performing for a shipment planning problem (among a number of methods implemented). It is also known that RF and gradient boosting have stronger predictive performance than neural nets with

tabular data of small to medium size. Since it is difficult to argue (with today's knowledge) that there is a single best method for prescriptive analytics with limited data, it is advisable to implement and compare multiple methods. We present some more empirical evidence that RF methods embedded into stochastic optimization are competitive but it would not be correct to make a general claim that RF methods would always work best.

There is a need to study a more diverse class of problems to develop common insights how ML-tools can be combined with optimization for operational decision making problems.

In the context of the random yield problem, we consider the single-period lot sizing model in this chapter. There are several other important versions of the problem: multiple lot sizing, multiple periods with stationary or non-stationary demand. Some static (open-loop) optimization versions of the problem may be addressed using similar approaches to the ones in this study. However, dynamic optimization brings a different set of challenges. Finally, significant planning challenges exist for agricultural yield where there are rich data sets (weather and soil related data) that are available. This may be a promising area for data-driven optimization.

Chapter 4

A DATA-DRIVEN MULTI-PERIOD LOT SIZING PROBLEM WITH RANDOM YIELD

4.1 Introduction

We investigate a problem motivated by semi-conductor manufacturing where a requirement of d good products have to be delivered to a customer by a deadline. Due to scheduling requirements, multiple lots are planned. Each lot is subject to random yield. The objective is to choose the sizes of each lot such that the production requirement is met at minimum cost.

The multi-period lot sizing problem is a critical issue in production and inventory management, where the primary goal is to determine optimal lot size quantities across multiple time periods while minimizing total costs. In real-world production environments, uncertainties such as fluctuating yield rates—affected by external factors like weather conditions, machine performance, and raw material quality—add complexity to this problem. As a result, production managers face significant challenges in making optimal decisions, especially when future yield uncertainties make it difficult to predict how much of the lot size quantity will actually be usable.

In industries facing yield uncertainties, distributing lot size quantities across multiple periods offers key advantages, such as mitigating risks, enhancing flexibility, and optimizing inventory management. This approach allows for dynamic adjustments to lot sizes based on updated information about yields and demand, thereby reducing the likelihood of excess inventory or shortages while minimizing overage and underage costs. Moreover, it facilitates the management of capacity constraints, mitigates the effects of poor yields, and enables learning from earlier periods, ultimately enhancing cost efficiency and production outcomes.

Traditional models like static and dynamic optimization models, often assume deterministic yields where the distribution of yield is known in advanced, which simplifies the problem but fails to capture the reality of variable production outcomes. We compare and contrast some approaches that take into account the randomness in the yield in a data-driven framework where the yield rate is estimated from data. Within static optimization, we propose a scenario-based stochastic optimization approach that takes inputs from predictions coming from data. Next, we formulate a stochastic dynamic program that allows for dynamic data-driven decision making. Finally, we propose a reinforcement learning method that estimates the yield rate and find the optimal lot sizing policy within the same optimization problem.

The primary contribution of this chapter is to extract data-driven decision rules from available yield data at each period, aiming to minimize overage and underage costs plus holding cost associated intermediate periods and production cost while ensuring that total customer demand is fulfilled by the final period. Reinforcement learning is utilized to dynamically determine the optimal lot sizes across periods, adapting to fluctuations in the yield process. To assess the effectiveness of the proposed methods, we compare their performance within the SECOM data set already used in Chapter 3.

This chapter is organized as follows: a literature review is presented in Section 4.2. The problem and the models including the reinforcement learning model, are introduced in Section 4.3. The main key performance indicator (KPI) is introduced in Section 4.6. In Section 4.7, we assess and compare the performance of the proposed approaches. Finally, conclusions are presented in Section 4.8.

4.2 Literature Review

The multi-period lot sizing problem with random yield has been a longstanding topic of research within production and operations management due to its practical implications in manufacturing environments. In this context, manufacturers are often required to meet rigid demand precisely while managing uncertainties in production yields that arise from imperfections in production processes. Research on this

topic spans from early single-period models to complex multi-period systems with yield randomness and various inspection or rework options. This literature review synthesizes key developments in lot sizing models with random yield, emphasizing recent advancements in multi-period systems and highlighting the novel contributions of reinforcement learning (RL) applications for optimizing lot sizes under yield uncertainty.

The multi-period lot sizing problem, particularly in manufacturing, has been a central research focus for decades. Initial models, such as the classic Economic Order Quantity (EOQ) model developed by Harris (1913), assumed deterministic yield rates, which simplified the problem but did not account for real-world yield uncertainties. These uncertainties, arising from machine breakdowns, raw material quality, or environmental variability, are pervasive in manufacturing environments. To bridge this gap, the concept of lot sizing with random yield was initially explored by Bowman (1955) and Llewellyn (1959), who proposed fundamental models to handle uncertainties in production output. These models were motivated by real-world manufacturing scenarios where the output of each production lot is not deterministic but rather affected by yield randomness due to factors like material defects or process instability. Early research in this domain provided essential insights but was largely limited to single-stage production scenarios. Over the years, researchers extended these foundational models to accommodate more realistic multi-stage production systems. Yano and Lee (1995b) provided a comprehensive review of lot sizing models with random yields, discussing both single and multi-stage models. They emphasized that random yields create unique challenges in production management, particularly when inspection and rework are necessary to meet demand in its entirety. Such reviews underscored the importance of finding a balance between large production lots, which could result in overproduction, and smaller lots, which may require more frequent setups.

Early deterministic lot sizing models, including the dynamic lot sizing model of Wagner and Whitin (1958), provided foundational approaches for optimizing production decisions when both demand and yield rates were known in advance. How-

ever, the limitations of deterministic models in capturing real-world uncertainties led to the development of stochastic lot sizing models. Stochastic lot sizing models emerged to address the limitations inherent in deterministic approaches. Yano and Lee (1995b) provide a comprehensive review of lot sizing models that incorporate random yields, emphasizing that uncertain yield rates introduce a need to balance the costs of underage and overage. These models aim to optimize lot sizes under uncertain production outcomes, ensuring demand fulfillment while minimizing excess production. Bookbinder and Tan (1988) explored probabilistic models in multi-stage production systems, showing how yield variability could be managed through planned lot sizing decisions that account for probabilistic outcomes.

Advances in optimization techniques, particularly Linear Programming (LP) and Dynamic Programming (DP), played a significant role in early stochastic models for multi-stage systems. Graves (1986) extended dynamic programming to multi-stage inventory systems, using recursive methods to optimize lot sizing at each production stage while managing overall system costs. Although these traditional techniques provided a structured way to handle multi-stage uncertainties, they are typically limited by their reliance on known probability distributions for yield outcomes. Grosfeld-Nir and Gerchak (2004) advanced multi-stage lot sizing by incorporating random yields, inspection, and rework strategies. They provided a framework for managing yield variability and highlighted the role of bottleneck stages in optimizing production costs under uncertainty.

The field of static optimization for multi-stage lot sizing has seen considerable development over recent decades, with each study contributing unique methodologies to manage demand uncertainty and production constraints. The foundational comparative analysis by Gupta and Keung (1992) evaluates various static lot sizing models across multi-stage production systems within a simulation framework, providing early insights into model performance across different scenarios. Following this, the hybrid optimization approach presented by Ramezanian and Saidi-Mehrabad (2013) combines simulated annealing with MIP-based heuristics to address the stochastic lot sizing and scheduling problem in a capacitated environment,

showcasing an adaptable solution for dynamic production demands. Later works have further refined these approaches to include complex demand patterns and supply chain intricacies. For instance, Hu et al. (2018) propose a multi-stage stochastic programming framework that accounts for demand uncertainty, introducing a robust optimization strategy for lot sizing and scheduling that has applications in various industrial settings. Similarly, Quezada et al. (2020) contribute a multi-echelon model that addresses the complexities of returns and lost sales, integrating stochastic integer programming to capture the dynamic needs of modern production systems. More recent studies build upon these frameworks, incorporating real-time updates and workforce considerations. Wan and Zhan (2021) extend multi-stage lot sizing models by including demand information updates in a flexible flow shop, demonstrating a practical, adaptable application in manufacturing environments. Lastly, Seyfi et al. (2023) utilize a scenario-based stochastic programming approach to jointly manage lot sizing and workforce scheduling, presenting a comprehensive solution for production planning in industries with fluctuating demand.

While traditional lot sizing models rely on assumed probability distributions for yields, recent advancements in data-driven methodologies, especially machine learning and reinforcement learning (RL), offer promising solutions for optimizing lot sizing decisions in complex, uncertain environments. Unlike traditional models, RL-based methods enable direct interaction with uncertain environments, allowing adaptive decision-making without predefined yield distribution assumptions. Reinforcement learning, introduced as a robust tool by Sutton and Barto (2018), has demonstrated significant promise in dynamically adjusting production strategies through iterative learning, making it particularly suitable for uncertain manufacturing environments.

Combining RL with deep learning, Mnih et al. (2015) introduced Deep Q-learning, which has since been applied to inventory and production control problems, showing enhanced performance over classical approaches. In the domain of multi-stage lot sizing, RL approaches can continuously adapt to varying yields by learning from historical and real-time production data. This adaptability is crucial for environ-

ments where yield distributions are either unknown or highly variable due to external factors like machine wear or raw material inconsistency.

Further, recent studies have explored hybrid approaches combining RL with Multi-Stage Stochastic Programming (MS), yielding computational advantages. Stranieri et al. (2024) proposed a hybrid DRL-MS model, wherein deep reinforcement learning is used to handle long-term production decisions, while MS programming manages short-term logistics under uncertain demand and capacity. This combination leverages RL's capacity for high-level decision-making and MS's efficiency in short-term optimization, providing a viable alternative to traditional LP methods in multi-stage lot sizing with random yields.

Contributions of This Chapter

Unlike previous studies that rely on known probability distributions or static optimization methods, our approach leverages dynamic programming and reinforcement learning to dynamically adjust lot sizes at each production period, utilizing either predicted yield rates or real-time data without requiring yield distribution assumptions. This enables the model to balance overage and underage costs more effectively, adapting to yield uncertainties while optimizing production decisions.

By implementing data-driven decision-making, this research contributes a novel approach to handling stochastic yield in multi-period lot sizing, reducing the need for distributional assumptions and enhancing adaptability to real-world production environments. In doing so, this chapter provides a modern perspective on optimizing lot sizing under uncertainty, building on both classical and contemporary methods to offer a resilient production solution.

4.3 Problem Setting and Data-Driven Estimation of Yield

We study a multi-period lot sizing problem in a factory setting, where the objective is to meet a contractual demand of d items by the end of a specified delivery period. The production process is divided into $N \in \{1, 2, \ldots, n\}$ distinct periods, with each period n assigned a lot size Q_n. As in the previous chapter, we assume multiplicative

production yield. At each period, the production yield is stochastic and modeled by a random variable Y_n. The aim is to determine the optimal lot size Q_n^* for each period to minimize the total costs, including underage and overage costs at the final period, holding costs for intermediate periods, and production costs.

Let I_N denote the inventory level at the end of the final period N, which is influenced by the yield values across all preceding periods. Let, b and h_N represent the per-unit underage and overage costs respectively. The mismatch cost between the number of good items that are produced by the deadline (I_N) and the contracted quantity d is given by:

$$\mathbb{E}\left[b\left(D - I_N\right)^+ + h_N\left(I_N - D\right)^+\right] \tag{4.1}$$

In addition, there are production and inventory carrying costs that incur before the deadline. Let c denote the per-unit production cost, and h_n is the per-unit holding cost at intermediate period n. A central challenge in this problem is the uncertainty in production yields, influenced by various external factors. In each period n, yield rates are random, impacting both the production output of that period and the inventory available for the subsequent period. This sequential dependence creates a multitude of scenarios, ultimately affecting the final inventory level at the last period. Given the data sets available for yield rates at each period, the primary objective is to derive a rule for determining optimal lot sizes.

$$\begin{aligned}\min_{Q_1,Q_2,\ldots,Q_N} z(Q_1,Q_2,\ldots,Q_N) = \mathbb{E}\Bigg[&b\left(D - I_N|(Q_1,Q_2,\ldots,Q_N)\right)^+ \\ &+ h_N\left(I_N|(Q_1,Q_2,\ldots,Q_N) - D\right)^+ \\ &+ c\sum_{n=1}^{N} Q_n + h_n\sum_{n=1}^{N-1} I_n|(Q_1,Q_2,\ldots,Q_N)\Bigg]\end{aligned} \tag{4.2}$$

The multi-period lot sizing problem has been extensively studied, though relatively few scholars have addressed stochastic yield variability at each period. Among

existing studies, researchers frequently employ static and dynamic optimization approaches to find solutions to Equation 4.2. We introduce these approaches as benchmarks for validating our model and then propose a novel reinforcement learning method that efficiently derives optimal lot sizing rules at each period, conditioned on the inventory levels from preceding periods.

Our main investigation in this thesis is on how to drive decisions using data on uncertain quantities (yield rates for this chapter). We assume for this chapter that similar yield data is available as in Section 3.3.3 of Chapter 3. We assume that the data contains past observations of yield rate y_i and also $\mathbf{U_i}$, a vector consisting of s features that can provide information on the random yield rate Y. Let us assume that we have observed a sample of n yield realizations (y_i) along with the corresponding values of the features $\mathbf{U}_i$ for each observation i. Each observation in the sample is then of the following form:

$$\mathbf{S}_i = (y_i, u_{i,1}, u_{i,2}, ..., u_{i,s}).$$

An example can be found in the SECOM data from Table 3.1 and its modified version for yield rates in Table H.1. In the multi-period problem, we need to estimate yield rates from the initial data for all periods. Depending on the assumptions of information availability, there might be different cases:

- If no feature data is available, we estimate the yield rate for all periods from the available yield data y_i. This assumes that the yield rates are stationary. We then need to estimate the distribution of $\hat{Y}$ from the data.

- If the features are available and are corresponding to the same product, we can then assume that the features $\mathbf{U}_i$ are identical for all periods of production. This assumes that the yield rates are stationary conditional on the given features. We then need to estimate the distribution of $\hat{Y}|\mathbf{U}_i$ from the data.

- If the features are available and are corresponding to the same product as well as the time of production conditions (that are known in advance), we can then

assume that the yield rate for period j depends on $\mathbf{U}_{ij}$. This assumes that the yield rates for period j are stationary conditional on the given features. We then need to estimate the distribution of $\hat{Y}_j|\mathbf{U}_{ij}$ from the data.

In this chapter, we mostly focus on the first case where we assume that past yield data is available but not the corresponding. We discuss later how to handle the other two cases.

In the next sections, we consider the optimization of the lot sizes under random yield rates that are estimated from data. We consider two main optimization approaches can be employed to determine Q_n^*: (i) ***Static Optimization***, which sets the order quantities at time zero (without recourse) and assigns an optimal lot size Q_n^* to each period, regardless of the outcomes of yield in previous periods; and (ii) ***Dynamic Optimization*** or ***Reinforcement Learning***, which computes the optimal lot size Q_n^* as a function of the yield y_{n-1} and inventory I_{n-1} from the previous period, denoted as $Q_n^*|(y_{n-1}, I_{n-1})$. These two approaches provide different formats for determining the optimal lot size at each period. The overall objective of the static problem can be expressed mathematically as follows. The objective function for the dynamic problem, which has a more complex structure, will be detailed in this section at a later period.

Next, we present two approaches for determining the order quantities Q_n ($n = 1, 2, .., N$) at time zero to minimize the expected costs taking into account the random yield. The first approach is a deterministic approximation based on point estimators for yield rates. The second approach estimates the yield rate distribution and proposes a scenario based formulation.

4.4 Static Optimization

In this section, we explore two distinct approaches for addressing the multi-period lot sizing problem under random yield uncertainty. The first approach, presented in Subsection 4.4.1, adopts a straightforward predict-then-optimize methodology that uses point predictors to estimate yield rates and subsequently determines optimal

lot sizes through a linear programming model. This method emphasizes simplicity by relying on point estimates derived from machine learning models. The second approach, detailed in Subsection 4.4.2, extends this framework by incorporating distributional estimators of yield rates, enabling a more nuanced stochastic programming model that accounts for uncertainty across multiple scenarios. Together, these subsections illustrate the progression from simple predictive methods to more sophisticated optimization techniques that leverage probabilistic yield distributions.

4.4.1 A Simple Predict-then-Optimize Approach using Point Predictors

The challenge of the optimization problem comes from the fact that yield rates are random and have to be estimated from data which may contain a large number of features. A simple idea is to completely separate the estimation and optimization periods. We have already seen in Chapter 3 that machine learning methods can be used to estimate the yield rates. The typical default output of a machine learning method is a point estimator for the mean of the yield rate. Let us assume that we use one of the tested methods (random forests, gradient boosting, neural nets) from Chapter 3 to obtain point estimators $\hat{y}_n$ for each period n of production.

$$\min \quad bu_N + h_N o_N + h_n \sum_{n \in N} I_n + c \sum_{n=1}^{N} Q_n \tag{4.3}$$

subject to:

$$I_n = \sum_{j=1}^{n} Q_j \hat{y}_j, \qquad \forall n \in \{1, 2, \ldots, N\} \tag{4.4}$$

$$C_{\min} \leq Q_n \leq C_{\max}, \qquad \forall n \in \{1, 2, \ldots, N\} \tag{4.5}$$

$$u_N \geq d - I_N, \tag{4.6}$$

$$o_N \geq I_N - d, \tag{4.7}$$

$$Q_n \geq 0, \qquad \forall n \in \{1, 2, \ldots, N\} \tag{4.8}$$

$$u_N, o_N \geq 0. \tag{4.9}$$

The objective function (4.3) minimizes the total cost across all periods $n \in$

$1, 2, ..., N$. Constraint (4.4) ensures that the total inventory in each period I_n is a function of the lot sizes Q_n and the random yields $\hat{y}_j$. Capacity constraints are enforced by (4.5), limiting the lot size Q_n in each period n to remain within the bounds of maximum and minimum capacity. Constraints (4.6) and (4.7) define the underage u_N and overage o_N inventories, ensuring that the demand d is met by adjusting these values. Finally, non-negativity constraints (4.8) and (4.9) ensure that both production quantities and the underage/overage values are non-negative. We note that the above model is a linear program albeit with a large number of decision variables due to the large number of scenarios.

4.4.2 Predict-then-Optimize Approach with Distributional Estimators

We have seen in Chapter 3 that distributional estimators of yield improve optimization performance. Let us assume that we can estimate the distribution of the yield by a random forest as in Chapter 3. Assume that we have M trees and that the yield point estimator from tree m is $\hat{y}_m$ (see Figure 3.4.1). We then estimate the yield probability distribution by a discrete distribution with probability mass $1/M$ for each outcome $\hat{y}_m$ $(m = 1, 2, .., M)$. In addition, if each period is estimated using different features, we would have a different distributional estimators for each period.

The evolution of the inventory level as a function of the chosen order quantities Q_n can be represented as in Figure 4.1

To simplify the model, let us assume that there are M possible yield rate point estimators for each period. The yield rates, denoted $\hat{Y}_{nm}$, are random variables that take values with corresponding probability p_{nm} representing that the items produced in period n have yield rate $\hat{y}_{nm}$. As illustrated in Figure 4.1, each period has M possible yield outcomes across the N periods, resulting in a total of $K = M^N$ distinct scenarios based on all possible yield combinations. To describe the scenarios let us assume that k a particular scenario which is a distinct combination of yield outcomes in all periods. For instance, scenario 1 may correspond to the smallest yield outcomes at all periods $(y_{11}, y_{21}, ..., y_{n1})$. We note that:

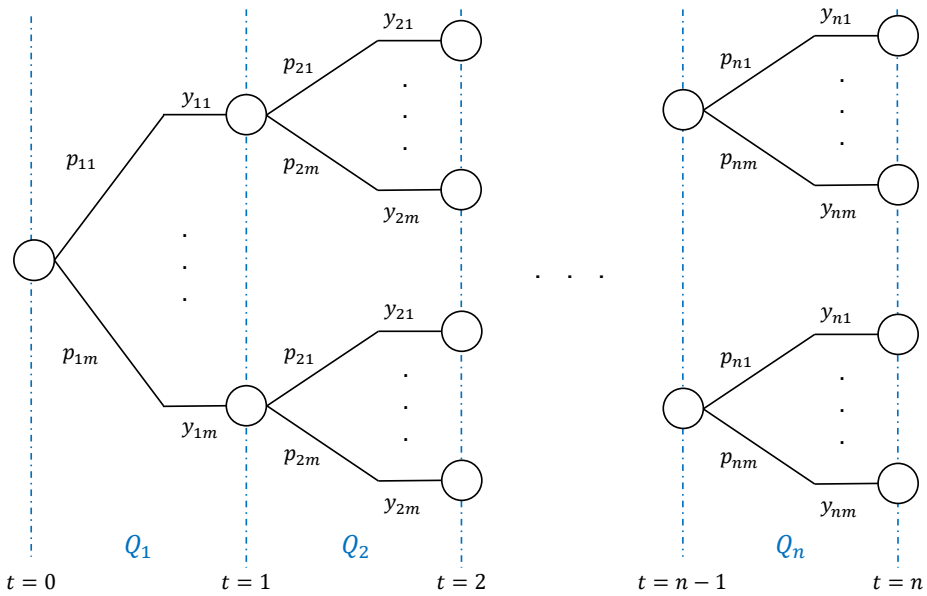

Figure 4.1: General framework of the problem: Multi-period lot sizing with probabilistic random yields

$$P_1 = \prod_{n=1}^{N} p_{n,1}$$

To generalize let us define $l_n(k)$, the index m of the outcome corresponding to scenario k in period n. Assuming independence of yield rates, the probability P_k of each scenario k is computed as the product of the individual yield probabilities:

$$P_k = \prod_{n=1}^{N} p_{nl_n(k)}$$

In this model, the delivery of the total demand d occurs at the end of the final period, with production from each period stored in inventory. The initial inventory at the beginning of the first period is assumed to be negligible. Due to yield uncertainty, only a fraction $y_{nl_n(k)}$ of the lot size Q_n is accepted as good items at each period, here $l_n(k)$ refers to the yield outcome in period n which falls in scenario k. The total inventory in scenario k, denoted by I_k, is calculated as the cumulative sum of the accepted quantities from all periods:

$$I_k = \sum_{n=1}^{N} Q_n y_{nl_n(k)}$$

At the end of the last period, the factory incurs overage costs h_N and underage costs b, depending on the available inventory. In this model, we assume a holding cost h_n is incurred for each unit of inventory held at each intermediate period 1 to $N-1$. Additionally, each unit of lot size incurs a cost c, and the lot size for each period is limited by a maximum and minimum production capacity constraints, denoted by $C_{\max}, C_{\min}$. The goal at each period is to determine the optimal lot size to meet the total demand while minimizing overall associated costs. The mathematical model is formulated as follows:

We can now formulate a stochastic program that uses the estimated yield distributions in a scenario-based model which chooses the lot sizes Q_n at time zero to minimize expected costs at delivery time.

$$\begin{aligned} \text{Minimize} \quad & b\sum_{k=1}^{K} P_k u_k + h_N \sum_{k=1}^{K} P_k o_k + h_n \sum_{k=1}^{K}\sum_{n=1}^{N-1} P_k Q_n y_{nl_n(k)} \\ & + c\sum_{n=1}^{N} Q_n \end{aligned} \tag{4.10}$$

Subject to:

$$I_k = \sum_{n=1}^{N} Q_n y_{nl_n(k)}, \qquad \forall k \in \{1, 2, \ldots, K\} \tag{4.11}$$

$$C_{\min} \leq Q_n \leq C_{\max}, \qquad \forall n \in \{1, 2, \ldots, N\} \tag{4.12}$$

$$u_k \geq d - I_k, \qquad \forall k \in \{1, 2, \ldots, K\} \tag{4.13}$$

$$o_k \geq I_k - d, \qquad \forall k \in \{1, 2, \ldots, K\} \tag{4.14}$$

$$Q_n \geq 0, \qquad \forall n \in \{1, 2, \ldots, N\} \tag{4.15}$$

$$u_k, o_k \geq 0, \qquad \forall k \in \{1, 2, \ldots, K\} \tag{4.16}$$

The objective function (4.10) minimizes the total cost across all scenarios $k \in 1, 2, ..., K$. Constraint (4.11) ensures that the total inventory in each scenario I_k is

a function of the lot sizes Q_n and the random yields $y_{nl_n(k)}$. Capacity constraints are enforced by (4.12), limiting the lot size Q_n in each period n to remain within the bounds of maximum and minimum capacity. Constraints (4.13) and (4.14) define the underage u_k and overage o_k inventories, ensuring that the demand d is met by adjusting these values. Finally, non-negativity constraints (4.15) and (4.16) ensure that both production quantities and the underage/overage values are non-negative. We note that the above model is a linear program albeit with a large number of decision variables due to the large number of scenarios.

In the literature, solutions to Equation 4.2 have been developed under the primary assumption of a known distribution of random yield. To determine optimal lot sizes, two main approaches have been proposed: a static optimization-based lot sizing method (SOLS) and a dynamic optimization-based lot sizing method (DOLS). Both methods aim to identify optimal lot sizes at each period, with the objective of minimizing the total system cost due to underage and overage costs at the final period. As benchmarks, we employ these two methods to validate the effectiveness of our proposed approach in terms of cost reduction.

4.5 Dynamic Optimization

This section introduces two advanced methods for solving the multi-period lot sizing problem under yield uncertainty: the Dynamic Optimization-based Lot sizing (DOLS) method and the Reinforcement Learning-based Lot sizing (RLLS) method. Subsection 4.5.1 explores the DOLS method, which employs dynamic programming to optimize lot sizes by leveraging the principle of optimality and accounting for realized yield rates and current inventory levels. Subsection 4.5.2 presents the RLLS method, which integrates estimation and optimization through reinforcement learning, allowing the agent to directly learn optimal lot sizing rules from observed yield data without requiring explicit yield estimators. Together, these methods highlight distinct approaches to managing uncertainty and minimizing costs in multi-period lot sizing problems.

4.5.1 Dynamic Optimization-based Lot sizing (DOLS) Method

The DOLS method offers a dynamic approach that allows for taking into account the current inventory level using the realized yield rates to solve the multi-period lot sizing problem.

Dynamic programming is an optimization technique that solves complex problems by breaking them down into simpler subproblems. It is particularly effective for sequential decision-making processes where decisions at one period affect outcomes in subsequent periods. In our multi-period lot sizing problem, dynamic programming allows us to determine optimal ordering quantities at each period by considering possible future outcomes and associated costs.

To express the dynamics, we use I_n, the inventory at the end of period n as the state variable and Q_n as the corresponding action in period n. The state transitions are random and governed by:

$$I_n = I_{n-1} + Q_n Y_n \quad \text{for } n = 1, 2, ..., N.$$

where we assume that $I_0 = 0$. In this multi-period inventory optimization problem, the value function recurrence relation (Equation 4.17) which is a form of the Bellman equation or Bellman recursion is denoted by $V_n(I_{n-1})$, which captures the minimum optimal expected cost incurred from period n to the final period N, given an inventory level I_{n-1} at the start of period n. This recursive relation enables the determination of optimal inventory decisions at each period by accounting for the randomness in yield rates and the probabilities associated with each yield scenario.

To use the data-driven estimation framework, we first note that if yield rates are estimated by point estimators as in Section 4.4.1, then we have a deterministic dynamic program whose solution is given by the solution of the linear program: (4.3)-(4.9).

Let us assume that the yield rate distributions are estimated from data as in Section, 4.4.2. The optimality equation is then given by:

$$V_n(I_{n-1}) = \min_{C_{\min} \leq Q_n \leq C_{\max}} cQ_n + h_{n-1}I_{n-1} + \left(\sum_{m=1}^{M} p_{nm} V_{n+1}(I_{n-1} + Q_n \hat{y}_{nm}) \right), \text{ for } n = 1, 2, ..., N-1 \quad (4.17)$$

Where Q_n represents the lot size quantity at period n, constrained by the maximum and minimum allowable capacity. The next inventory level is given for Q_n and yield rate $\hat{y}_{nm}$ is given by: $I_n = I_{n-1} + Q_n \hat{y}_{nm}$. This recursive structure, solved backward from period N, helps identify the optimal lot sizes while minimizing the total expected costs throughout all periods.

For the final period N, the value function incorporates the underage and overage costs:

$$V_N(I_{N-1}) = \min_{C_{\min} \leq Q_N \leq C_{\max}} \left(\sum_{m=1}^{M} p_{Nm} \Big[b \max(0, d - (I_{N-1} + Q_N \hat{y}_{nm})) + h_N \max(0, (I_{N-1} + Q_N \hat{y}_{nm}) - d) \Big] + cQ_N \right) \quad (4.18)$$

The numerical solution of the above dynamic program finds the optimal lot sizing policy (i.e. a rule that yields an order quantity as a function of the inventory level for each period n). The solution also gives the expected optimal cost under the optimal policy.

The DOLS method leverages dynamic programming to determine optimal lot sizes and policies based on inventory levels at each period. While DP is not inherently computationally efficient, the method's efficiency depends on the discretization of the state space, which involves dividing the range of possible inventory levels and yields into manageable intervals. It offers reduced complexity compared to the SOLS method, better handles yield scenarios, and allows for greater flexibility in incorporating constraints, enhancing decision-making insights across periods.

State space discretization involves dividing continuous variables (e.g., inventory levels, yields, or lot size quantities) into discrete intervals. This reduces the infinite

possibilities of continuous variables into a manageable finite set for computational analysis. In this setup, lot size quantities are discretized into pre-defined ranges, while inventory levels are grouped into intervals using a function, which rounds values to the nearest multiple of inventory bin size. Similarly, yield distributions are approximated with discrete values and their probabilities. This systematic discretization ensures computational feasibility while maintaining solution accuracy.

To validate the optimal policy derived from the DOLS model using the yield rate estimators from the training data, we simulate the production process using test yield data. In each simulation, we start by applying the optimal lot size Q_1^* determined for period 1. The resulting inventory I_1 is calculated using the observed yield y_1 from the test data. For period 2, the policy determines the lot size Q_2^* based on the discretized inventory level I_1, matched to the closest pre-computed value in the policy table. Using the observed yield y_2 from the test data, the inventory I_2 is computed. This process continues for all subsequent periods, ultimately calculating the final inventory I_N. The associated costs, including underage, overage, production, and holding costs, are computed for each scenario. The simulation is repeated across multiple test scenarios, and the average cost is evaluated to assess the performance and robustness of the policy under realistic yield variability.

4.5.2 Reinforcement Learning-based Lot Sizing (RLLS) Method: Unknown Yield Distribution

In this section, we propose a Reinforcement Learning (RL) method that integrates estimation and optimization. Reinforcement learning (RL) enables an agent to learn optimal policies through interaction with the environment, making it well-suited for handling problems with uncertain and dynamic conditions. For this subsection, we assume that the only data available is the observations of the yield rates y_i. Unlike the previous methods, we do not attempt to first obtain a point estimator or a distributional estimator to later use in an optimization framework. The RL approach proposes to learn the optimal lot sizing rule directly from the data.

Our implementation of the RLLS method utilizes Q-learning, a model-free RL

algorithm, to learn the optimal lot size policy in a multi-period production process with uncertain yields. The agent learns to make lot size decisions at each period by maximizing cumulative rewards, which are defined as the negative of the costs incurred due to underage and overage inventories.

In the context of multi-period lot sizing under uncertainty, several key concepts define the framework. The **Agent** refers to the decision-maker responsible for learning the optimal ordering policy across periods. The **Environment** represents the production process, which is subject to probabilistic yields that influence the outcome of each period. The **State Space** comprises the discretized inventory levels at each period, $s_n = I_n$, providing the agent with information on current inventory status. The **Action Space** includes the discretized possible lot size quantities available to the agent at each period, determining the lot size decisions, $a_n = Q_n$. The **Reward Function** is the negative of the cost incurred, where the agent is encouraged to minimize total costs by optimizing its decisions, as follows:

$$R_n = \begin{cases} -[b(d - I_n)^+ + h_N(I_n - d)^+ + cQ_n], & \text{if } n = N, \\ -[cQ_n + h_n I_n], & \text{otherwise.} \end{cases}$$

The **Policy** is a rule mapping states to actions, which the agent learns over time to maximize cumulative rewards. To facilitate this learning, the agent uses a **Q-table**, which stores the expected utility (Q-values) of taking certain actions in given states, helping it make informed decisions to minimize overall costs. As indicated in the reward function, overage and underage costs are calculated in the final period. In the intermediate periods, we account for holding costs associated with items in inventory, alongside production costs incurred at each period. Q-value is updated based on following rule:

$$Q(s_n, a_n) \leftarrow Q(s_n, a_n) + \alpha \left[R_n + \gamma \max_{a'} Q(s_{n+1}, a') - Q(s_n, a_n) \right],$$

where α is the learning rate and γ is the discount factor. Then, for each state s,

the optimal action $a^*(s)$ is calculated as following:

$$a^*(s) = \arg\max_a Q(s, a).$$

The RLLS method involves the agent interacting with the environment over numerous episodes to learn the optimal policy. The main steps of the algorithm are as follows:

1. **Initialize Parameters**:
   - Define the action space A_n for each period n.
   - Discretize the inventory levels to form the state space S_n for each period.
   - Initialize the Q-table $Q(s, a)$ with optimistic values to encourage exploration.
   - Set learning rate α, discount factor γ, and exploration parameters for the epsilon-greedy policy.
2. **Episode Iteration**:
   - For each episode:
     (a) Initialize the current period $n = 0$ and inventory $I_0 = 0$.
     (b) While $n < N$:
       – **Action Selection**: Choose an action Q_n using the epsilon-greedy policy.
       – **Environment Interaction**: Apply Q_n and observe the yield Y_{nm} to get the next inventory I_{n+1}.
       – **Reward Computation**: Calculate the reward R_n, which is the negative of the cost incurred.
       – **Q-table Update**: Update the Q-value using the Bellman equation.
       – **State Transition**: Move to the next period $n = n + 1$.

3. **Policy Extraction**:

   - After training, extract the optimal policy by selecting the action with the highest Q-value for each state.

4. **Performance Evaluation**:

   - After training, the optimal policy is extracted and evaluated using test yield data. The total expected cost is calculated to assess the performance of the RLLS method. The flowchart of RLLS method is shown in Figure 4.2.

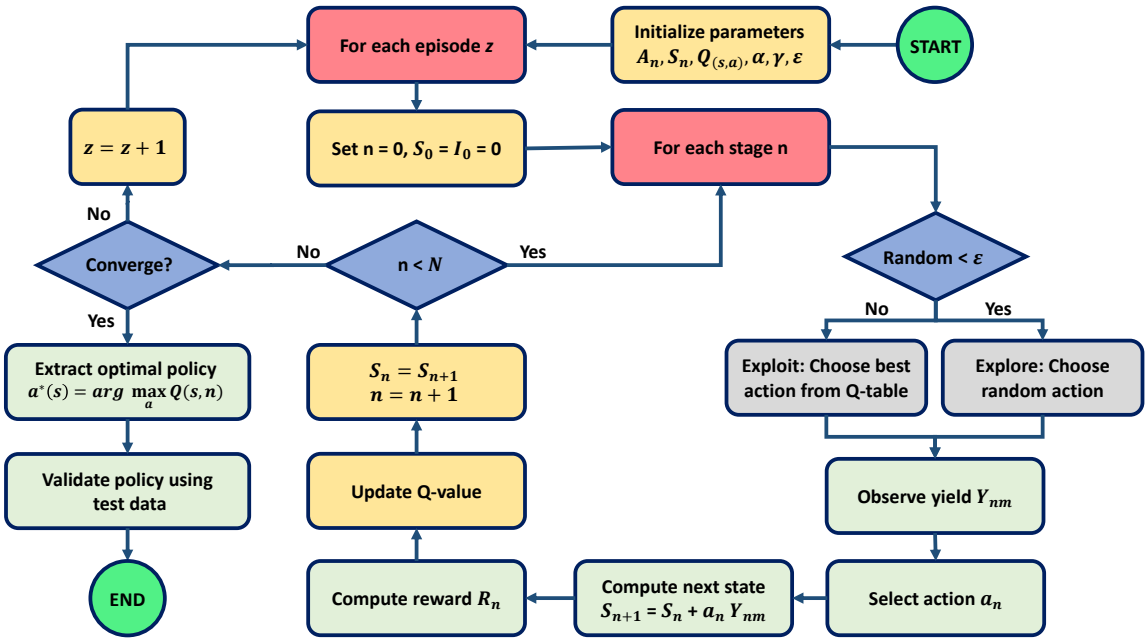

Figure 4.2: Flowchart of RLLS method

The RLLS method offers several distinct advantages in tackling uncertain production environments. One of its primary benefits is model-free learning, meaning it

does not require prior knowledge of yield distributions, allowing it to operate effectively in settings where such information is unavailable or hard to estimate. Another key strength is its adaptability, as the method can adjust to changing conditions in the environment over time, making it well-suited for dynamic, real-world applications. Moreover, the RLLS method exhibits scalability, handling high-dimensional problems that might challenge traditional optimization methods, proving particularly useful in complex production processes.

Compared to traditional methods, the RLLS method provides several significant advantages. First, it operates without a distribution assumption, freeing it from the need to rely on predefined yield distributions. Second, its ability to learn from data means it continually improves by interacting with the environment, enhancing decision-making as more data becomes available. Lastly, its versatility allows it to address various types of uncertainties and cost structures, providing broader applicability across different manufacturing settings.

To further enhance the RLLS method, several key implementation steps can be taken. Hyperparameter tuning is essential, as experimenting with learning rates, discount factors, and exploration strategies can optimize the model's performance. Scaling up the method to include more periods or moving towards continuous state and action spaces through function approximators can expand its utility in larger, more complex problems. Lastly, integration into production planning systems for real-time decision-making would offer practical application, allowing companies to implement RLLS in operational environments effectively.

The RLLS method represents a significant advancement in solving lot sizing problems under uncertainty, offering a flexible, adaptive, and scalable solution for complex production environments without the need for explicit distributional knowledge.

4.6 Performance Evaluation

To evaluate the performance of the proposed lot sizing methods—SOLS, DOLS, and RLLS—we apply them to two data sets: the SECOM data set and a generated

data set from the previous chapter. For each data set, we divide the whole data into multiple periods. In this evaluation, we assume there are two periods. Each period's data is split into training and test sets, following an 80-20% ratio.

For the SOLS and DOLS methods, we first estimate the yield distributions for each period using the training data. We consider several candidate probability distributions, such as the Beta, Normal, Gamma, and Exponential Weibull distributions. To determine the best-fitting distribution, we perform a Kolmogorov-Smirnov goodness-of-fit test, selecting the distribution with the highest p-value for each period. Once the best distribution is selected, we fit its parameters to the training data to create the stochastic model for yield rates. In Figure 4.3, the fitted distributions for the training sample of the generated data set are presented. For both periods, the Beta distribution is identified as the best fit, with parameters (3.7891, 1.7516, 0.1910, 0.7659) for the first period and (4.0028, 1.7299, 0.1768, 0.7743) for the second period, where the parameters represent α, β, location, and scale, respectively. For the SECOM data set, the fitted distributions are Beta with parameters (97.8593, 0.8010, -7.1866, 8.1866) and an Exponentiated Weibull with parameters (0.1723, 775726391.9676, -33273097.6893, 33273098.7044).

After fitting the distributions, we discretize them into yield scenarios. This discretization process generates a set of possible yield outcomes along with their associated probabilities. These scenarios are then used to model the uncertain production yields in each period, allowing the SOLS and DOLS methods to optimize lot sizes based on these probabilistic outcomes.

For the RLLS method, we directly use the yield values from the training data to allow the reinforcement learning agent to learn optimal policies through interaction with the environment. This method does not require a pre-estimation of yield distributions, making it suitable for cases where the yield distributions are unknown or difficult to estimate.

Once the optimal lot sizing policies are extracted from each method, we apply them to the test set to evaluate their performance. The primary evaluation criterion is the minimization of the overall system cost, which includes both underage and

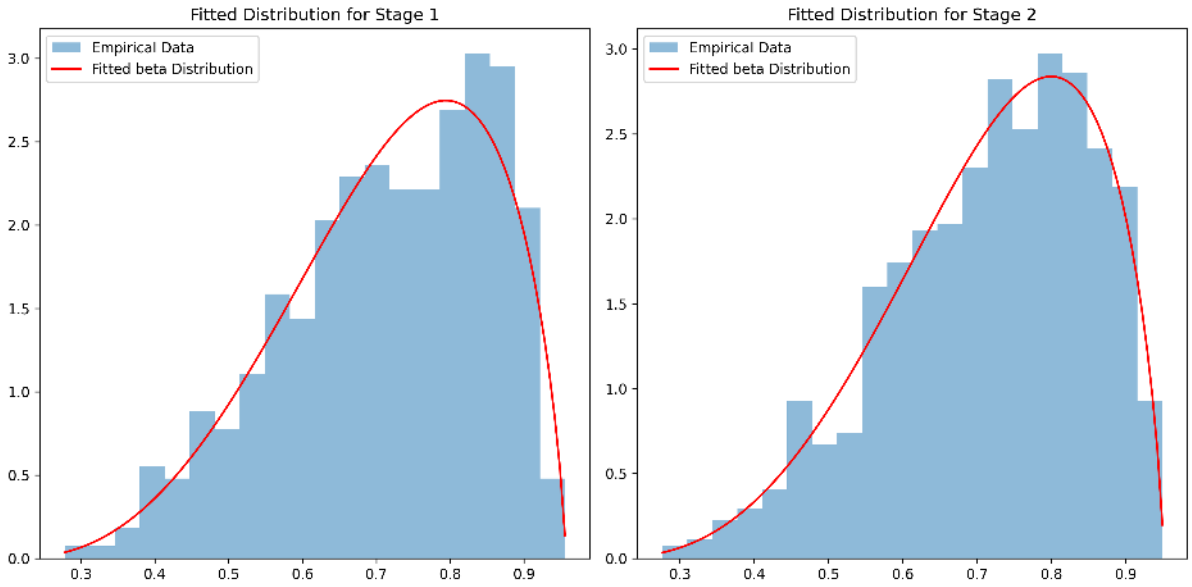

Figure 4.3: Fitted Distribution on Generated Data Set

overage costs at the end of the last period plus holding costs in intermediate periods and production cost. By comparing the results from the SOLS, DOLS, and RLLS methods, we assess the effectiveness of each approach in addressing the uncertainties in production yields.

4.7 Results and Discussion

We evaluate the performance of the SOLS, DOLS, and RLLS methods on two data sets: a generated data set and the SECOM data set. The objective is to minimize the total system cost associated with underage, overage, holding and per-unit production costs, where the total demand is $D = 600$. Two scenarios are analyzed:

Scenario 1: A two-period problem where the delivery of items occurs at the end of the second period. The lot size capacities for the two periods are 600 and 1200 units, respectively. The underage and overage costs are set to 5 and 2, respectively. The initial inventory is assumed to be zero, the holding cost at intermediate periods

is set to 0.8, and the per-unit production cost is set to 0.2.

Scenario 2: A three-period problem where the delivery of items occurs at the end of the third period. The lot size capacities for the three periods are 350, 400, and 700 units, respectively. The same underage and overage costs as in Scenario 1 are assumed. The initial inventory is set to zero, and both the holding cost at intermediate periods and the per-unit production cost are set to 0, representing the most basic problem setup. Detailed results for each data set and method are presented below.

4.7.1 Generated Data Set

Scenario 1: Two-period problem

Under the described settings for two-period problem, the models are provided with the generated data set to determine the optimal lot size quantities for each period and calculate the corresponding costs on the test set. Subsequently, we evaluate the performance of our proposed model in comparison to traditional models in terms of cost reduction. The results are presented as follows:

SOLS method

For the SOLS method (static optimization), the optimal lot sizes for the two periods are 100 and 743 units, respectively. The average cost on the test data for this method is 504.36. This method may result in a higher overall cost compared to the other methods due to its reliance on a static yield distribution model that may not fully capture variability in yields across periods.

DOLS method

For the DOLS method, the optimal lot size in the first period is 100 units. In the second period, the optimal lot sizes vary based on the yield rate and the inventory level (I_1) at the end of period 1. Figure 4.4 illustrates the potential yield rates in the first period and inventory levels at the end of the first period, along with the

corresponding optimal lot sizes in the second period. The results indicate that as the yield rate increases, the inventory level also increases, leading to a reduction in the optimal lot size for the subsequent period.

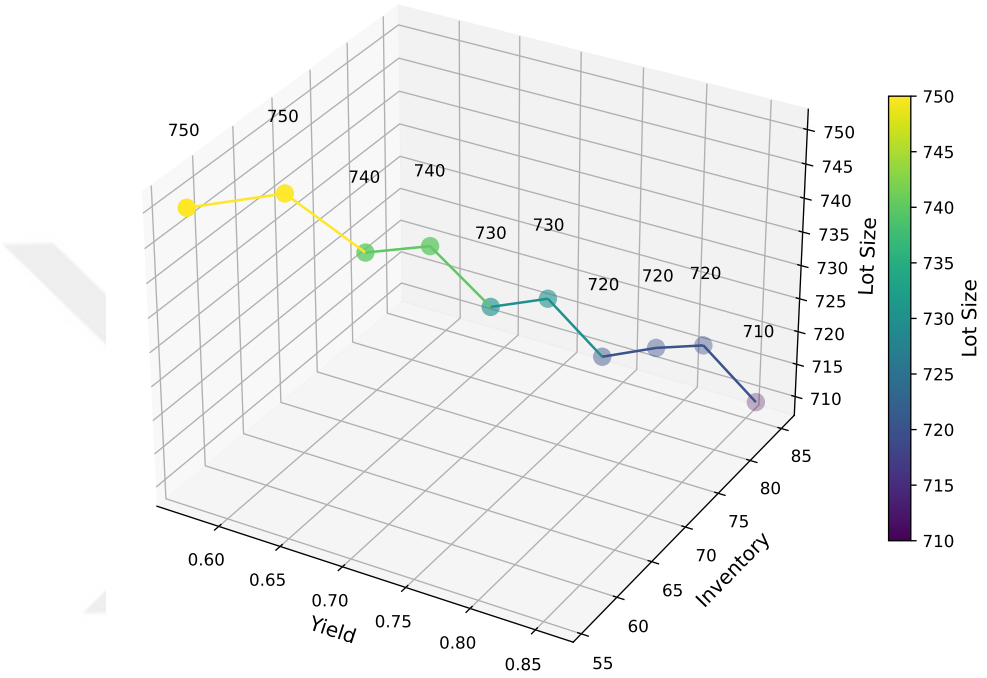

Figure 4.4: Two-period lot sizing rule for generated data set using the DOLS method

The average cost on the test data for the DOLS method is 267.48, indicating a substantial reduction in total cost compared to the SOLS method (about 47% cost reduction). Furthermore, as demonstrated by the lot sizing rule, an increase in the yield rate results in a corresponding decrease in the lot size.

RLLS method

The RLLS method yields the following optimal policies. The optimal lot size in period 1 is 150 units. As shown in Figure 4.5, the optimal lot sizes in period 2 vary with the yield rate and inventory levels from the previous period. Similar to the DOLS method, as the yield rate increases, the inventory level also increases, resulting

in a reduction in the optimal lot size for the subsequent period. The colorful line represents the optimal lot sizing rule for the generated data set obtained using the RLLS method. Compared to DOLS, the variation in the lot sizing rule is lower; however, for lower yield rates, this method tends to recommend higher lot size quantities compared to the DOLS method.

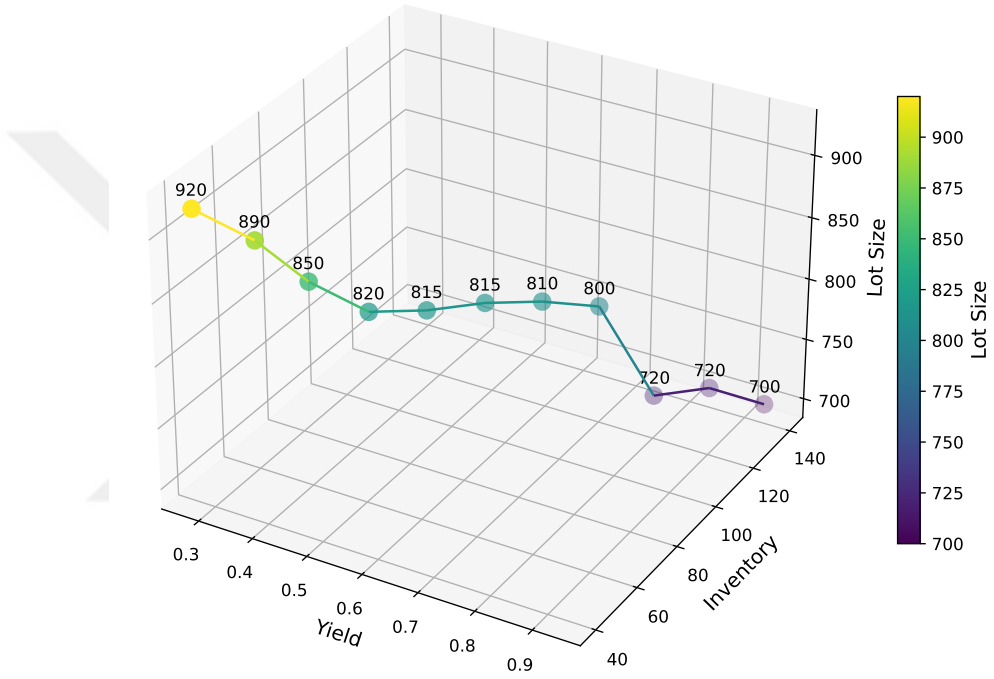

Figure 4.5: Two-period lot sizing rule for generated data set using the RLLS method

The average cost on the test data for the RLLS method is 279.40. While the RLLS method performed slightly worse than DOLS, it significantly outperformed SOLS, demonstrating that reinforcement learning is well-suited to addressing the problem of uncertain yield distributions. Figure 4.6 shows the smoothed total reward and Q-value across episodes. As depicted, the Q-values converge to a stable range after approximately 20,000 episodes, with no further improvement observed. In the left panel, the average award stabilizes at the same range.

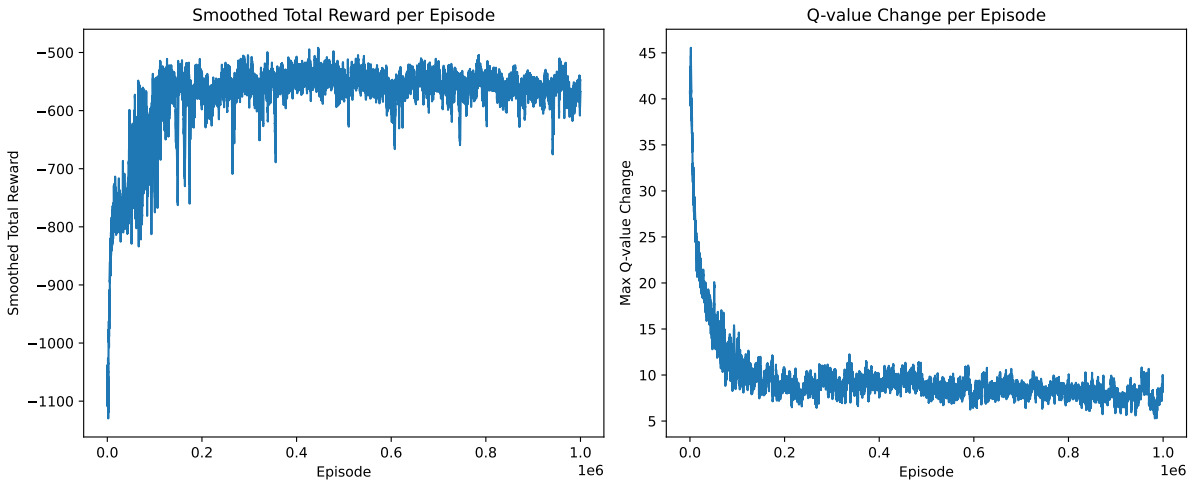

Figure 4.6: Optimal Lot Sizes and Cost per Episode in Generated Data Set (2 periods)

Scenario 2: Three-period Problem

In this scenario, we extend the solution approach described above to solve the three-period problem. The detailed results are presented as follows:

SOLS method

For the SOLS method, the optimal lot sizes for the three periods are 254, 314, and 266 units, respectively. The average cost on the test data for this method is 70.69. Similar to the two-period problem, this cost may be higher compared to the other two methods, as SOLS relies on static yield distribution models and does not incorporate the variability in yields or enable dynamic inventory adjustments across periods.

DOLS method

For the DOLS method, the optimal lot sizes are determined based on the yield rates and inventory levels at each period. In the first period, the optimal lot size is fixed

at 350 units. As shown in Figure 4.7 (left frame), the optimal lot size in the second period is adjusted as follows: if the yield rate exceeds 0.77, the lot size is set to 300 units; if the yield rate falls between 0.77 and 0.65, the lot size remains 350 units; otherwise, it is increased to 400 units. For the third period, as illustrated in Figure 4.7 (right frame), the optimal lot sizes are dynamically adjusted depending on both the yield rate and the inventory levels at that period. Consistent with previous results, as the yield rate increases, the inventory level also increases, leading to a reduction in the optimal lot size for the subsequent period.

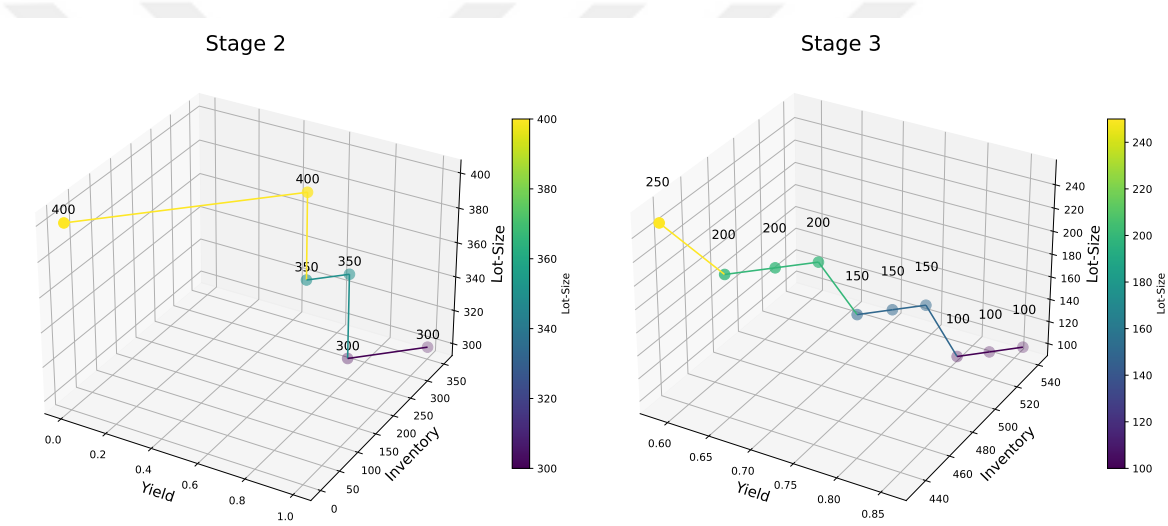

Figure 4.7: Three-period lot sizing rule for generated data set using the DOLS method

The minimum expected total cost for the DOLS method is 29.99, demonstrating a significant reduction compared to the SOLS method (about 58% cost reduction).

RLLS method

For the RLLS method, the optimal lot sizes are determined based on the yield rates and inventory levels at each period. In the first period, the optimal lot size is fixed at 350 units. As shown in Figure 4.8 (left frame), the optimal lot size in the second

period is adjusted as follows: if the yield rate exceeds 0.71, the lot size is set to 300 units; if the yield rate falls between 0.71 and 0.57, the lot size remains 350 units; otherwise, it is increased to 400 units. For the third period, as illustrated in Figure 4.8 (right frame), the optimal lot sizes are dynamically adjusted depending on both the yield rate and the inventory levels at that period. Consistent with previous results, as the yield rate increases, the inventory level also increases, leading to a reduction in the optimal lot size for the subsequent period. But compared to the DOLS method, the lot size quantity stays in much higher levels especially for low yield rates.

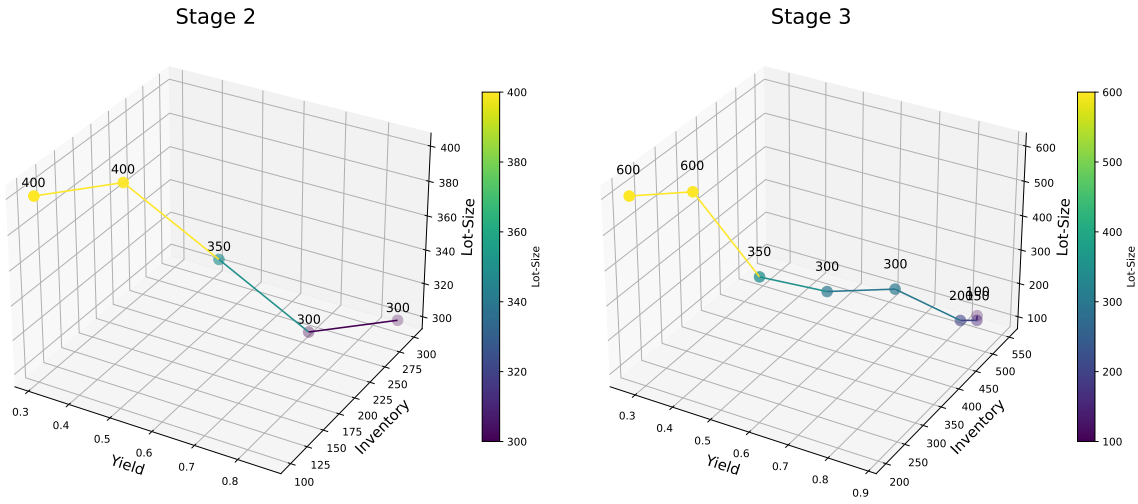

Figure 4.8: Three-period lot sizing rule for generated data set using the RLLS method

The total expected cost for the RLLS method on the test set is 27.84, which is slightly lower than DOLS, however, the RLLS method significantly outperformed SOLS, showcasing its suitability for addressing uncertain yield distributions. Figure 4.9 shows the smoothed total reward and Q-value across episodes. As depicted, the

Q-values converge to a stable range after approximately 20,000 episodes, with no further improvement observed. In the left panel, the average award stabilizes at the same range.

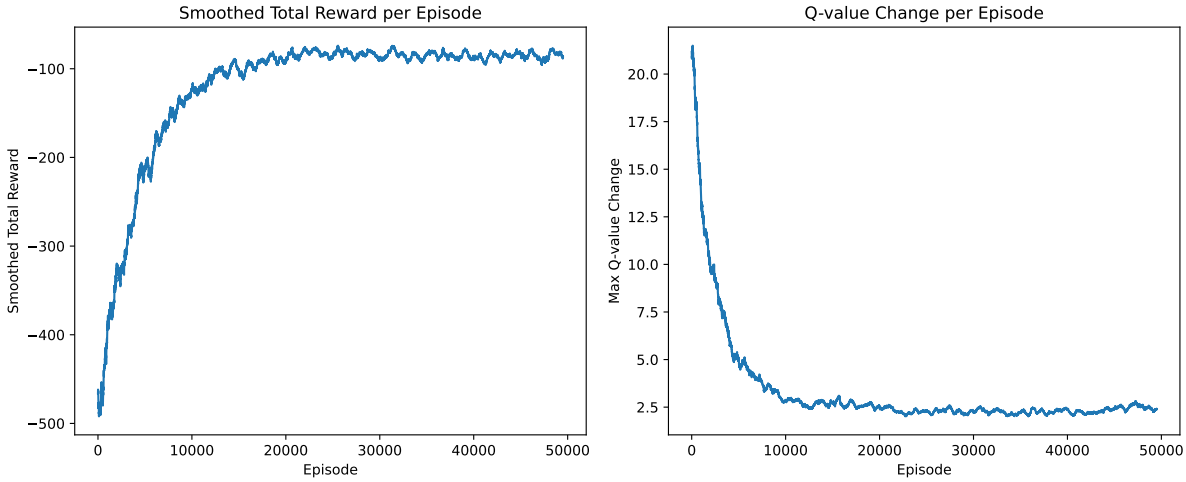

Figure 4.9: Optimal Lot Sizes and Cost per Episode in Generated Data Set (3 periods)

4.7.2 SECOM Data Set

Scenario 1: Two-period problem

The models are provided with the SECOM data set to determine the optimal lot size quantities for each period and calculate the corresponding costs on the test set. The results are presented as follows:

SOLS method

For the SOLS method (static optimization), the optimal lot sizes for the two periods are 259 and 400 units, respectively. The average cost on the test data for this method is 578.32. Like the previous section, this method may result in a higher overall cost

compared to the other methods due to its reliance on a static yield distribution model that may not fully capture variability in yields across periods.

DOLS method

For the DOLS method, the optimal lot size in the first period is 260 units. As shown in Figure 4.10, the optimal lot size in the second period does not vary regardless of the yield rate and inventory level (I_1) at the end of period 1. This indicates that the lot sizing rule under the DOLS method is insensitive to the yield rate. The likely reason for this behavior is the low quality of the data and the fitted distribution, as the DOLS method relies on distributions as input.

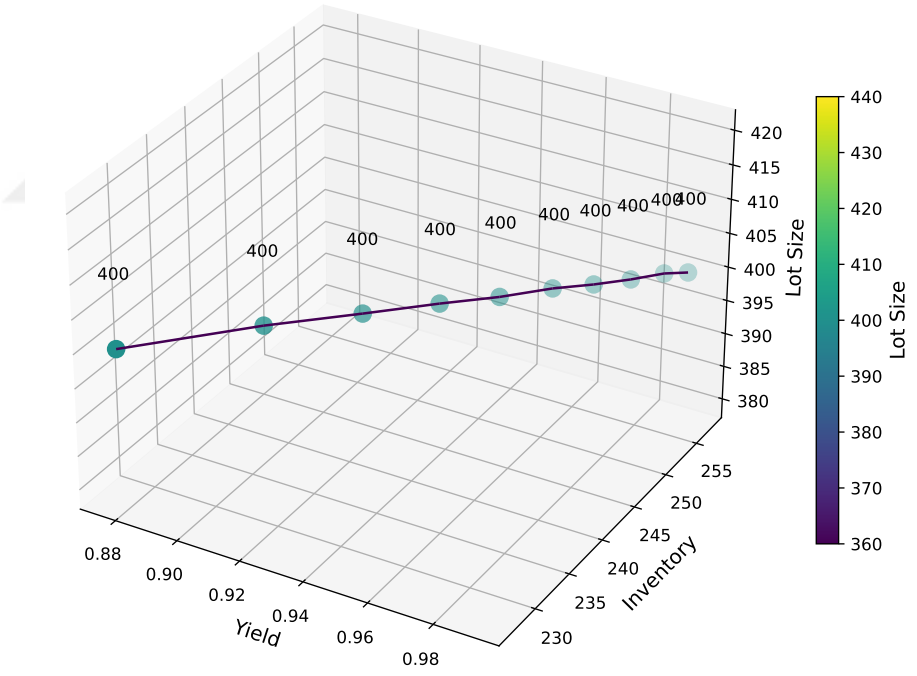

Figure 4.10: Two-period lot sizing rule for SECOM data set using the DOLS method

The average cost on the test data for the DOLS method is 363.97, indicating a substantial reduction in total cost compared to the SOLS method (about 37% cost reduction). Furthermore, as demonstrated by the lot sizing rule, an increase in the

yield rate results in a corresponding decrease in the lot size.

RLLS method

The RLLS method yields the following optimal policies. The optimal lot size in period 1 is 390 units. As shown in Figure 4.11, unlike the DOLS method, the optimal lot size in period 2 varies and is sensitive to both the yield rate and inventory levels. The lot sizing rule increases as the yield rate and inventory levels decrease.

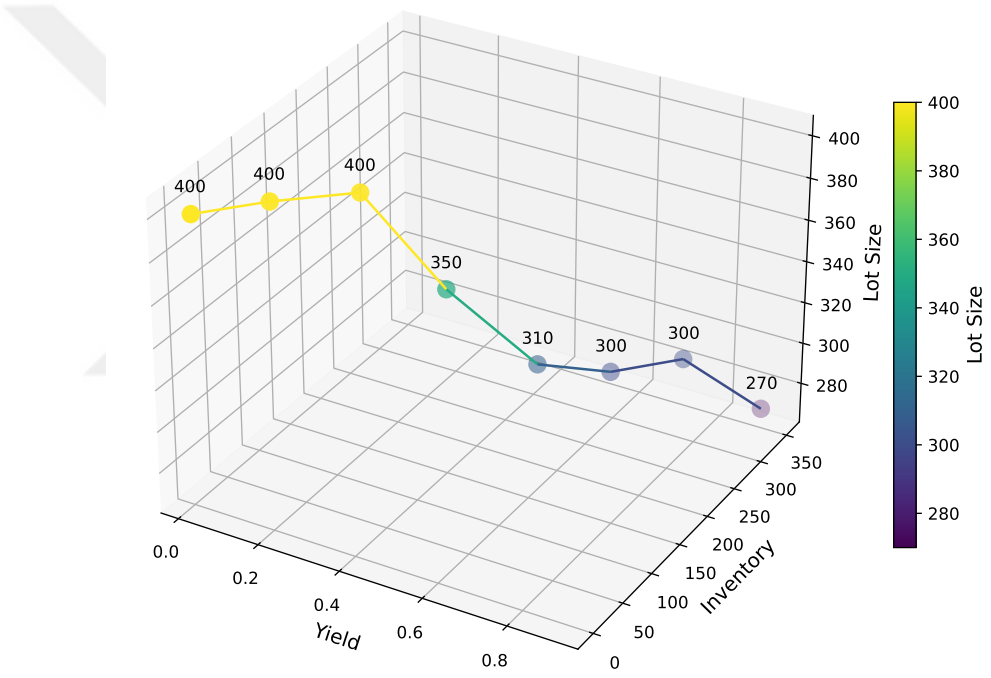

Figure 4.11: Two-period lot sizing rule for SECOM data set using the RLLS method

The average cost on the test data for the RLLS method is 351.51. While the RLLS method slightly outperformed DOLS, it significantly surpassed SOLS, highlighting the effectiveness of reinforcement learning in addressing the challenges posed by uncertain yield distributions. Figure 4.12 illustrates the smoothed total reward and Q-value across episodes. As shown, compared to the generated data set, the Q-values for the SECOM data set exhibit more noise, which may be attributed to

the quality of the provided data set. However, on average, the Q-values converge to a stable range after approximately 50,000 episodes, with no further improvement observed. Similarly, in the left panel, the average reward stabilizes within the same range.

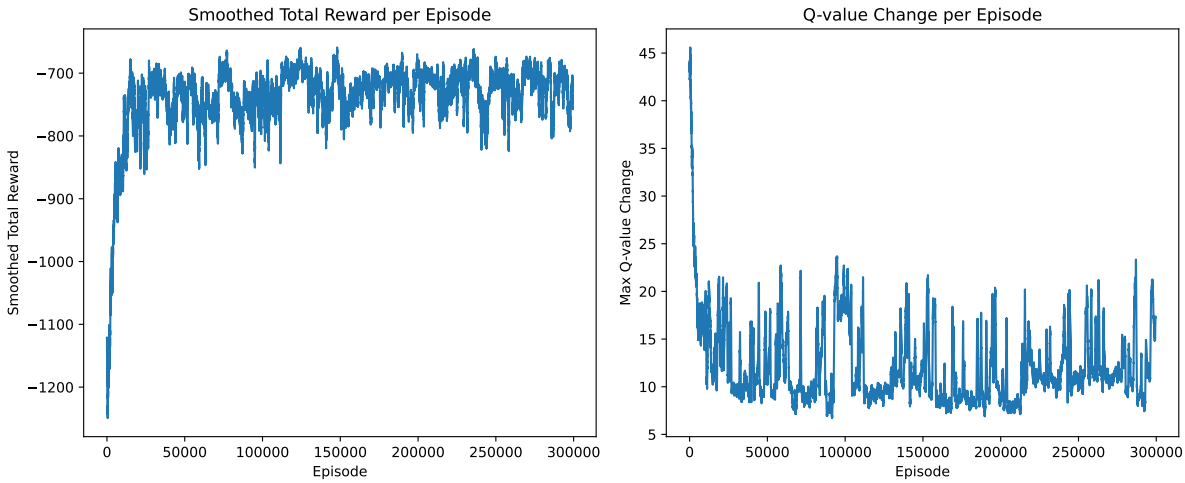

Figure 4.12: Optimal Lot Sizes and Cost per Episode in Generated Data Set (2 periods)

Scenario 2: Three-period Problem

In this scenario, we extend the solution approach described above to solve the three-period problem. The detailed results are presented as follows:

SOLS method

For the SOLS method, the optimal lot sizes for the three periods are 178, 400, and 100 units, respectively. The average cost on the test data for this method is 46.12. Similar to the two-period problem, this cost may be higher compared to the other two methods, as SOLS relies on static yield distribution models and does not

incorporate the variability in yields or enable dynamic inventory adjustments across periods.

DOLS method

For DOLS method, the optimal lot sizes are determined based on the yield rates and inventory levels at each period. In the first period, the optimal lot size is fixed at 350 units. In the second period, as shown in Figure 4.13 (left frame), the optimal lot size is adjusted as follows: if the yield rate exceeds 0.94, the lot size is set to 200 units; if the yield rate falls between 0.94 and 0.79, the lot size remains 250 units; otherwise, it is increased to 300 units. For the third period, as shown in Figure 4.13 (right frame), the optimal lot sizes are dynamically adjusted depending on both the yield rate and inventory levels at that period. However, limited variability is observed in this period, likely due to the quality of the data set and the fitted distribution.

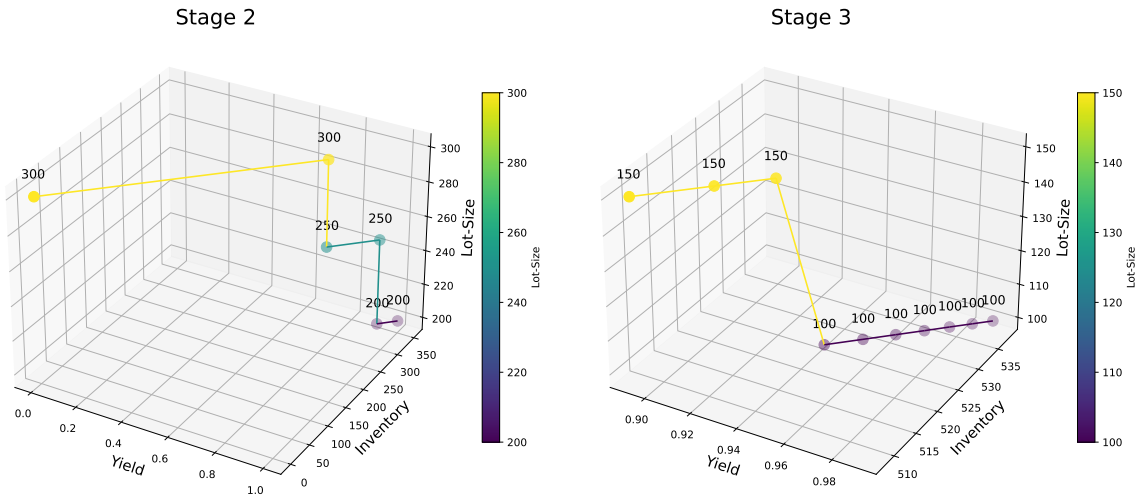

Figure 4.13: Three-period lot sizing rule for SECOM data set using the DOLS method

The minimum expected total cost for the DOLS method is 57.27. In this case, the DOLS model does not outperform the SOLS method, which could be attributed to the quality of the data set.

RLLS method

For the RLLS method, the optimal lot sizes are determined based on the yield rates and inventory levels at each period. In the first period, the optimal lot size is fixed at 200 units. As shown in Figure 4.8 (left frame), the optimal lot size in the second period is adjusted as follows: if the yield rate exceeds 0.75, the lot size is set to 350 units; if the yield rate falls between 0.75 and 0.51, the lot size remains 380 units; otherwise, it is increased to 400 units. For the third period, as illustrated in Figure 4.14 (right frame), the optimal lot sizes are dynamically adjusted depending on both the yield rate and the inventory levels at that period. Consistent with previous results, as the yield rate increases, the inventory level also increases, leading to a reduction in the optimal lot size for the subsequent period. But compared to the DOLS method, the lot size quantity stays in much higher levels especially for low yield rates.

The total expected cost for the RLLS method on the test set is 39.13, which is significantly outperformed SOLS and DOLS around 15% and 32% in terms of cost reduction on test set, showcasing its suitability for addressing uncertain yield distributions. Figure 4.15 shows the smoothed total reward and Q-value across episodes. As depicted, the Q-values converge to a stable range after approximately 50,000 episodes, with no further improvement observed. In the left panel, the average award stabilizes at the same range.

Table 4.1 summarizes the results of applying three methods to two data sets for determining the optimal lot size quantities in two scenarios: two-period and three-period problems. As previously discussed, the SOLS method determines static optimal lot sizes for each period, whereas the DOLS and RLLS methods compute dynamic optimal lot sizes based on the yields and inventory from previous periods. The table presents the average lot sizes ($\bar{Q}$) across all possible states and their cor-

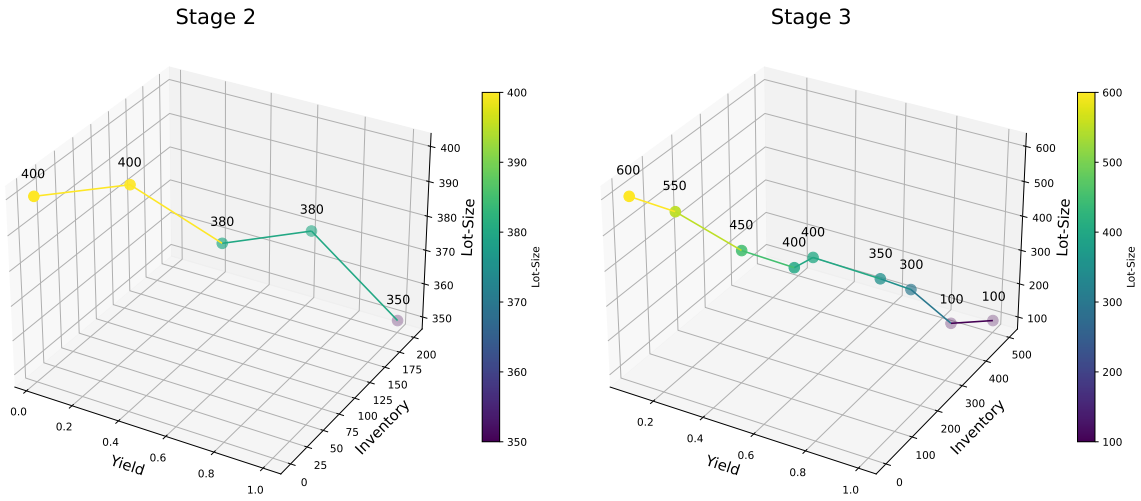

Figure 4.14: Three-period lot sizing rule for SECOM data set using the DOLS method

responding costs. The lowest cost for each scenario and data set is highlighted in bold. Notably, except for the two-period scenario using the generated data set, the RLLS method consistently outperforms the SOLS and DOLS methods, achieving cost reductions ranging from 15% to over 60%. Additionally, for the two-period scenario under the generated data set, the RLLS method produces results comparable to those of the DOLS method.

4.8 Conclusion

In this chapter, we investigate a multi-period single-item lot sizing problem, where a contractual agreement requires the delivery of D items to customers across n production periods. In each period, we utilize a data set of random yields, which influence both the production output at that period and the resulting inventory levels in subsequent periods. The primary objective is to derive a data-driven rule for

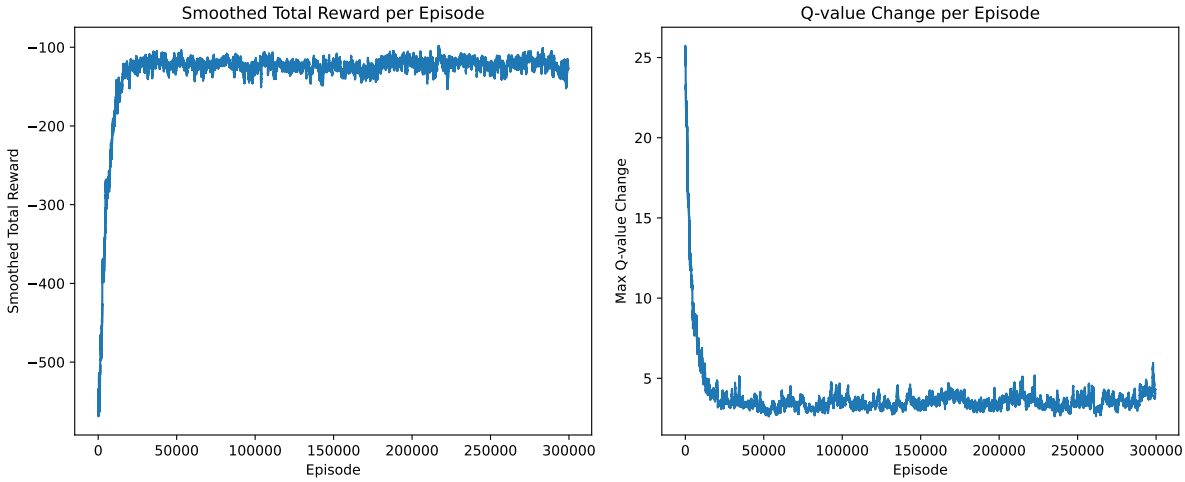

Figure 4.15: Optimal Lot Sizes and Cost per Episode in Generated Data Set (3 periods)

Table 4.1: Performance Comparisons on Two and Three periods for Generated and SECOM Data Sets

Data set	**Method**	**Two periods**			**Three periods**			
		Q_1	$\bar{Q}_2$	**Cost**	Q_1	$\bar{Q}_2$	$\bar{Q}_3$	**Cost**
Generated	SOLS	100	743	504.36	254	314	266	70.69
	DOLS	100	731	**264.48**	350	350	160	29.99
	RLLS	150	805	279.40	350	350	325	**27.84**
SECOM	SOLS	259	400	578.32	178	400	100	46.12
	DOLS	260	400	363.97	350	250	111	57.27
	RLLS	390	341	**351.51**	200	382	361	**39.13**

determining optimal lot sizes under unknown yield distributions. This approach contrasts with conventional methods in the literature that typically assume known yield

distributions. Our rule is designed to minimize the expected underage and overage costs at the end of the final period, incorporating holding costs for intermediate periods and variable production costs, while ensuring that the total customer demand is satisfied. This chapter contributes to the literature by introducing reinforcement learning as a novel approach to address the multi-period lot sizing problem with stochastic yields and unknown distributions. To evaluate the effectiveness of our proposed approach, we compare its performance against established static and dynamic optimization methods. The results demonstrate that our method provides a flexible and adaptive framework for decision-making in complex production systems characterized by stochastic yields.

We evaluate two scenarios: a two-period lot sizing problem and a three-period lot sizing problem, using two distinct data sets. The proposed method, RLLS, consistently outperformed existing methods from the literature, which rely on static and dynamic optimization techniques. In the best cases, our approach can reduce the total costs up to 60% compared to the benchmarks. Furthermore, unlike conventional approaches, our method does not require prior knowledge of yield distributions at each period. The RLLS approach generates dynamic lot sizing rules that determine optimal lot sizes based on inventory levels and yield outcomes from preceding periods. This adaptability makes RLLS a powerful tool for decision-makers in stochastic production environments.

Chapter 5

CONCLUSION

In this thesis, we addressed key challenges in data-driven decision-making and optimization under uncertainty by developing innovative methods that integrate machine learning, optimization. Across three distinct but interrelated chapters, we tackled problems in decision-making, production and inventory management, and multi-period lot sizing, demonstrating the applicability and efficacy of our approaches in diverse domains. The overarching goal was to provide practical, interpretable, and effective solutions for real-world problems characterized by uncertainty and asymmetric costs.

In Chapter 2, we introduced four novel methods to address data-driven decision-making problems under asymmetric cost structures. These methods leveraged machine learning models and optimization techniques to derive decision rules. By prioritizing cost minimization over error reduction, our methods demonstrated substantial improvements, reducing costs by up to 95% compared to traditional benchmarks. These findings underscore the importance of explicitly incorporating cost asymmetry into decision-making models, particularly in high-stakes contexts such as healthcare.

Chapter 3 extended the focus to data-driven single-stage lot sizing problems in production and inventory management. Here, we explored the integration of machine learning-based predictions with optimization frameworks to determine feature-dependent optimal lot sizes. Using state-of-the-art predictive models, we demonstrated that incorporating additional feature data into decision-making significantly reduces overage and underage costs. Among the methods tested, the combination of random forest algorithms with theoretical optimization insights emerged as the most effective, providing a robust and interpretable framework for prescriptive analytics.

Building on the single-stage framework, Chapter 4 addressed the more complex data-driven multi-period lot sizing problem with stochastic yields. We proposed a novel reinforcement learning-based approach that does not require prior knowledge of yield distributions. Instead, our method dynamically adapts lot sizing decisions at each period based on observed yield outcomes and inventory levels. Evaluations across two- and three-period scenarios demonstrated the superiority of the reinforcement learning method, which consistently outperformed static and dynamic optimization benchmarks, reducing total costs by up to 60%. This adaptability and flexibility make the proposed approach a powerful tool for decision-makers in stochastic production environments.

Together, the three chapters of this thesis contribute to the growing body of research on data-driven decision-making under uncertainty. By integrating machine learning and optimization, we bridged the gap between predictive and prescriptive analytics, providing actionable insights and practical solutions for complex problems. Future research could expand upon this work by exploring scalability, incorporating richer datasets, and applying these methods to other domains, such as agricultural planning and logistics. The approaches developed in this thesis lay a strong foundation for continued advancements in decision-making and optimization, empowering decision-makers to navigate uncertainty with confidence and precision.

BIBLIOGRAPHY

Aharon Ben-Tal, L. E. G. and Nemirovski, A. (2009). *Robust Optimization*. Princeton University Press.

Al-Kharaz, M., Ananou, B., Ouladsine, M., Combal, M., and Pinaton, J. (2019). Quality prediction in semiconductor manufacturing processes using multilayer perceptron feedforward artificial neural network. In *2019 8th international conference on systems and control (ICSC)*, pages 423–428. IEEE.

Ban, G.-Y. and Rudin, C. (2019). The big data newsvendor: Practical insights from machine learning. *Operations Research*, 67(1):90–108.

Bennett, K. P. (1992). Decision tree construction via linear programming. *Proceedings of the 4th midwest artificial intelligence and cognitive science society conference*, pages 97–101.

Bennett, K. P. and Blue, J. A. (1996). Optimal decision trees. *Rensselaer Polytechnic Institute*.

Bertsimas, D. and Dunn, J. (2017). Optimal classification trees. *Springer*, 106:44.

Bertsimas, D. and Kallus, N. (2020). From predictive to prescriptive analytics. *Management Science*, 66(3):1025–1044.

Biau, G. and Devroye, L. (2010). On the layered nearest neighbour estimate, the bagged nearest neighbour estimate and the random forest method in regression and classification. *Journal of Multivariate Analysis*, 101(10):2499–2518.

Bibak, B. (2024). bbml: A python package for fitting classification with customized cost matrix. https://pypi.org/project/bbml/. Accessed: 2024-08-18.

Birge, J. R. and Louveaux, F. V. (1997). *Introduction to Stochastic Programming*. Springer.

Bitran, G. R. and Gilbert, S. M. (1994). Co-production processes with random yields in the semiconductor industry. *Operations Research*, 42(3):476–491.

Bookbinder, J. H. and Tan, J. Y. (1988). Strategies for the probabilistic lot-sizing problem with service-level constraints. *Management Science*, 34(9):1096–1108.

Bowman, E. H. (1955). The scheduling of economic lot sizes in production systems. *Journal of the Operations Research Society of America*, 3(1):67–78.

Breiman, L. (2001). Random forests. *Machine Learning*, 45(1):5–32.

Breiman, L., Friedman, J., Olshen, R., and Stone, C. (1984a). *Classification and Regression Trees*. Wadsworth & Brooks/Cole Advanced Books & Software.

Breiman, L., Friedman, J. H., Olshen, R. A., and Stone, C. J. (1984b). Classification and regression trees.

Carey, J. C. and Hall, J. C. (1995). The use of the logistic model in genetic counseling: An application of genetic models to the study of familial aggregation of asthma. *Journal of Genetic Counseling*, 4(3):145–156.

Charnes, A. and Cooper, W. W. (1959). Chance-constrained programming. *Management Science*, 6:73–79.

Chen, Z. and Xie, W. (2021). Regret in the newsvendor model with demand and yield randomness. *Production and Operations Management*, 30:4176–97.

Chollet, F. et al. (2015). Keras. https://keras.io.

Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine learning*, 20(3):273–297.

Cox, D. R. (1958). The regression analysis of binary sequences. *Journal of the Royal Statistical Society: Series B (Methodological)*, 20(2):215–242.

Cox, L.A., Q. Y. . K. W. (1989). Heuristic least-cost computation of discrete classification functions with uncertain argument values. *Annals of Operations Research*, 21:1–29.

Czako, Z., Sebestyen, G., and Hangan, A. (2021). Automaticai – a hybrid approach for automatic artificial intelligence algorithm selection and hyperparameter tuning. *Expert Systems with Applications*, 182:115225.

Dantzig, G. B. (1955). Linear programming under uncertainty. *Management Science*, 1:197–206.

Diefenbach, J. (2023). *Design and Control of Stochastic Manufacturing Systems*. PhD Dissertation, University of Mannheim.

Domingos, P. and Pazzani, M. (1997). On the optimality of the simple bayesian classifier under zero-one loss. *Machine Learning*, 29(2-3):103–130.

Duda, R. O., Hart, P. E., and Stork, D. G. (2001). *Pattern Classification*. Wiley, 2nd edition.

Elkan, C. (2001). The foundations of cost-sensitive learning. In *Proceedings of the 17th International Joint Conference on Artificial Intelligence*, volume 1, pages 973–978.

Erkip, N. K. (2023). Can accessing much data reshape the theory? inventory theory under the challenge of data-driven systems. *European Journal of Operational Research*, 308(3):949–959.

Friedman, J. (2001a). Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, 29(5):1189–1232.

Friedman, J. H. (2001b). Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, 29(5).

Gerchak, Y. and Parlar, M. (1990). Yield randomness, cost tradeoffs and diversification in the EOQ model. *Naval Research Logistics*, 37:341–354.

Gerchak, Y., Vickson, R., and Parlar, M. (1988). Periodic review production models with variable yield and uncertain demand. *IIE Transactions*, 20:144.

Graves, S. C. (1986). A multi-echelon inventory model for a repairable item with one-for-one replenishment. *Management Science*, 32(10):1247–1256.

Green, D. M. and Swets, J. A. (1966). The theory of signal detectability. *Wiley*.

Grosfeld-Nir, A. and Gerchak, Y. (2004). Multiple lotsizing in production to order with random yields: Review of recent advances. *Annals of Operations Research*, 126:43–69.

Grosfeld-Nir, A., G. Y. (2004). Multiple lotsizing in production to order with random yields: Review of recent advances. *Annals of Operations Research*, 126:43–69.

Gupta, Y. P. and Keung, Y. (1992). Comparative analysis of lot-sizing models for multi-stage systems: A simulation study. *International Journal of Production Research*, 30:2251–2264.

Gurobi Optimization, LLC (2024). Gurobi Optimizer Reference Manual.

Haberman, S. (1999). Haberman's Survival. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C5XK51.

Harris, F. W. (1913). How many parts to make at once. *Factory, The Magazine of Management*, 10(2):135–136.

Hastie, T., Tibshirani, R., Friedman, J. H., and Friedman, J. H. (2009). *The elements of statistical learning: data mining, inference, and prediction*, volume 2. Springer.

Henig, I. and Gerchak, Y. (1990). The structure of periodic review policies in the presence of random yield. *Operations Research*, 38:634–643.

Hu, C., Lu, J., Liu, X., and Zhang, G. (2018). Robust vehicle routing problem with hard time windows under demand and travel time uncertainty. *Computers & Operations Research*, 94.

Janosi, Andras, S. W. P. M. and Detrano, R. (1988). Heart Disease. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C52P4X.

Kall, P. and Mayer, J. (2005). *Stochastic Linear Programming: Models, Theoryand Computation*. Springer Verlag, New York.

Karlin, S. (1958). One stage model with uncertainty. in K. J. Arrow, S. Karlin and H. Scarf (Eds.) Studies in the Mathematical Theory of Inventory and Production, pp. 109-143 Stanford: Stanford University Press.

Keze Wang, Dongyu Zhang, Y. L. R. Z. and Lin, L. (2017). Cost-effective active learning for deep image classification. *IEEE Transactions on Circuits and Systems for Video Technology*, 27(12):2591–2600.

Kleywegt, A. J., Shapiro, A., and Homem-de Mello, T. (2001). The sample average approximation method for stochastic discrete optimization. *SIAM Journal on Optimization*, 308(3):479–502.

LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., and Jackel, L. D. (1989). Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1(4):541–551.

Lee, D.-H., Yang, J.-K., Lee, C.-H., and Kim, K.-J. (2019). A data-driven approach to selection of critical process steps in the semiconductor manufacturing process considering missing and imbalanced data. *Journal of Manufacturing Systems*, 52:146–156.

Lemaître, G., Nogueira, F., and Aridas, C. K. (2017). Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning. *Journal of Machine Learning Research*, 18(17):1–5.

Levi, R., Roundy, R. O., and Shmoys, D. B. (2007). Provably near-optimal sampling-based policies for stochastic inventory control models. *Mathematics of Operations Research*, 32(4):821–839.

Lin, Y. and Yongho, J. (2006). Random forests and adaptive nearest neighbors. *Journal of the American Statistical Association 101*, (474):578–90.

Liu, C., Letchford, A. N., and Svetunkov, I. (2022). Newsvendor problems: An integrated method for estimation and optimisation. *European Journal of Operational Research*, 300(2):590–601.

Liu, X.-Y. and Zhou, Z.-H. (2006a). On multi-class cost-sensitive learning. *Proceedings of the 21st National Conference on Artificial Intelligence (AAAI)*.

Liu, X.-Y. and Zhou, Z.-H. (2006b). Training cost-sensitive neural networks with methods addressing the class imbalance problem. *IEEE Transactions on Knowledge and Data Engineering*, 18(1):63–77.

Liyanage, L. H. and Shanthikumar, J. G. (2005). A practical inventory control policy using operational statistics. *Operations Research Letters*, 33(4):341–348.

Llewellyn, D. W. (1959). Production scheduling in a continuous system with random yields. *Journal of Industrial Engineering*, 10:203–213.

Loh, W.-Y. (2011). Classification and regression trees. *WIREs Data Mining and Knowledge Discovery*, 1(1):14–23.

Mandl, C. and Minner, S. (2023). Data-driven optimization for commodity procurement under price uncertainty. *Manufacturing & Service Operations Management*, 25(2):371–390.

Mandl, C., Nadarajah, S., Minner, S., and Gavirneni, S. (2022). Data-driven storage operations: Cross-commodity backtest and structured policies. *Production and Operations Management*, 31(6):2438–2456.

McCann, M. and Johnston, A. (2008a). SECOM. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C54305.

McCann, M. and Johnston, A. (2008b). Secom dataset uci machine learning repository.

Mnih, V., Kavukcuoglu, K., Silver, D., et al. (2015). Human-level control through deep reinforcement learning. *Nature*, 518:529–533.

Naoki Abe, B. Z. and Langford, J. (2004). An iterative method for multi-class cost-sensitive learning. *Proceedings of the 2004 SIAM International Conference on Data Mining*.

Natekin, A. and Knoll, A. (2013). Gradient boosting machines, a tutorial. *Frontiers in Neurorobotics*, 7(21).

Neghab, D. P., Khayyati, S., and Karaesmen, F. (2022). An integrated data-driven method using deep learning for a newsvendor problem with unobservable features. *European Journal of Operational Research*, 302(2):482–496.

Neumann, J. V. and Morgenstern, O. (1944). *Theory of Games and Economic Behavior*. Princeton University Press.

Norouzi, M., Collins, M., Johnson, M. A., Fleet, D. J., and Kohli, P. (2015). Efficient non-greedy optimization of decision trees. *Proceedings of the Advances in Neural Information Processing Systems*, page 1729–1737.

Notz, P. M. and Pibernik, R. (2022). Prescriptive analytics for flexible capacity management. *Management Science*, 68(3):1756–1775.

Okyay, H., Karaesmen, F., and Özekici, S. (2014). Newsvendor models with dependent random supply and demand,. *Optimization Letters*, 8:983–999.

Oroojlooyjadid, A., Snyder, L. V., and Takáč, M. (2020). Applying deep learning to the newsvendor problem. *IISE Transactions*, 52(4):444–463.

Özekici, S. and Parlar, M. (1999). Inventory models with unreliable suppliersin a random environment. *Annals of Operations Research*, 91:123–136.

Pearl, J. (1988). Probabilistic reasoning in intelligent systems: Networks of plausible inference. *Morgan Kaufmann*.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al. (2011). Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830.

Platt, J. C. (1999). Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In *Advances in Large Margin Classifiers*, pages 61–74. MIT Press.

Quezada, F., Gicquel, C., and Kedad-Sidhoum, S. (2020). A multi-stage stochastic integer programming approach for a multi-echelon lot-sizing problem with returns and lost sales. *Computers & Operations Research*, 113:104785.

Ramezanian, R. and Saidi-Mehrabad, M. (2013). Hybrid simulated annealing and mip-based heuristics for stochastic lot-sizing and scheduling problem in capacitated multi-stage production system. *Applied Mathematical Modelling*, 37:10122–10134.

Rockafellar, R. T. and Uryasev, S. (2000). Optimization of conditional value-at-risk. *Journal of Risk*, 2:21–41.

Romano, R., Davino, C., and Næs, T. (2014). Classification trees in consumer studies for combining both product attributes and consumer preferences with additional consumer characteristics. *Food Quality and Preference*, 33:27–36.

Sachs, A.-L. (2015). The data-driven newsvendor with censored demand observations. In *Retail Analytics*, pages 35–56. Springer.

Salman H. Khan, Munawar Hayat, M. B. F. A. S. and Togneri, R. (2018). Cost-sensitive learning of deep feature representations from imbalanced data. *IEEE Transactions on Neural Networks and Learning Systems*, 29(8):3573–3587.

Scarf, H. (1958a). A min-max solution of an inventory problem. *Studies in the Mathematical Theoryof Inventoryand Production. Stanford University Press, Stanford, CA*.

Scarf, H. (1958b). A min-max solution of an inventory problem. *Studies in the Mathematical Theory of Inventory and Production*.

Scarf, H. (1959). Bayes solutions of the statistical inventory problem. *The Annals of Mathematical Statistics*, 30(2):490–508.

Scornet, E. (2015). Random forests and kernel methods.

Seyfi, S., Yanıkoğlu, , and Yılmaz, G. (2023). Multi-stage scenario-based stochastic programming for managing lot-sizing and workforce scheduling at vestel. *Annals of Operations Research*.

Shih, W. (1980). Optimal inventory policies when stockouts result from defective products. *International Journal of Production Research*, 18:677–686.

Son, N. H. (1998). From optimal hyperplanes to optimal decision trees. *Fundamenta Informaticae*, 34:145–174.

Song, J.-S. and Zipkin, P. (1993a). Inventory control in a fluctuating demand environment. *Operations Research*, 41(2):351–370.

Song, J.-S. and Zipkin, P. (1993b). Inventory control in a fluctuating demand environment. *Operations Research*, 41(2):351–370.

Stranieri, F., Fadda, E., and Stella, F. (2024). Combining deep reinforcement learning and multi-stage stochastic programming to address the supply chain inventory management problem. *International Journal of Production Economics*, 268:109099.

Susto, G. A., Pampuri, S., Schirru, A., Beghi, A., and De Nicolao, G. (2015). Multi-step virtual metrology for semiconductor manufacturing: A multilevel and regularization methods-based approach. *Computers & Operations Research*, 53:328–337.

Sutton, R. S. and Barto, A. G. (2018). *Reinforcement learning: An introduction*. MIT press.

Turney, P. D. (1995). Cost-sensitive classification: Empirical evaluation of a hybrid genetic decision tree induction algorithm. *Journal of Artificial Intelligence Research*, 2:369–409.

Van Parys, B. P., Esfahani, P. M., and Kuhn, D. (2020). From data to decisions: Distributionally robust optimization is optimal. *Management Science*.

Viaene, S. and Dedene, G. (2004). Cost-sensitive learning and decision making revisited. *European Journal of Operational Research*, 166:212–220.

Wagner, H. M. and Whitin, T. M. (1958). Dynamic version of the economic lot size model. *Management Science*, 5(1):89–96.

Wan, G. and Zhan, Y. (2021). Multi-level, multi-stage lot-sizing and scheduling in the flexible flow shop with demand information updating. *International Transactions in Operational Research*, 28:890–905.

Wolberg, William, M. O. S. N. and Street, W. (1995). Breast Cancer Wisconsin (Diagnostic). UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C5DW2B.

Yano, C. and Lee, H. (1995a). Lot sizing with random yields: A review. *Operations Research*, 43:311–334.

Yano, C. A. and Lee, H. L. (1995b). Lot sizing with random yields: A review. *Operations Research*, 43(2):311–334.

Zadrozny, B. and Elkan, C. (2001). Learning and making decisions when costs and probabilities are both unknown. *Proceedings of the 7th International Conference on Knowledge Discovery and Data Mining (KDD)*.

Zadrozny, B. and Langford, J. (2005). Cost-sensitive learning by cost-proportionate example weighting. *Proceedings of the 3rd IEEE International Conference on Data Mining*.

Appendix A

MIP MODEL FOR OPTIMAL DECISION TREE (MIPODT)

In this section, we present the MIP formulation for decision tree by considering the asymmetric cost. Suppose we aim to construct an optimal decision tree with a maximum depth of D. Given this depth, we can construct the maximal tree of such depth, consisting of $T = 2^{(D+1)} - 1$ nodes, which we index as $t = 1, ..., T$. Figure A.1 illustrates the maximal tree of depth 2.

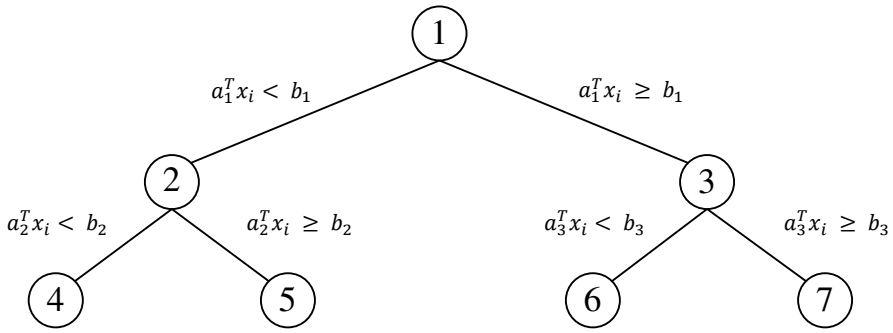

Figure A.1: Classification tree with $D = 2$

In our notation, we use $\mathcal{P}(t)$ to denote the parent node of node t, while $\mathcal{A}(t)$ represents the set of ancestors of node t. Additionally, we define $\mathcal{L}(t)$ as the set of ancestors of t, where the left branch has been followed on the path from the root node to t. Similarly, we define $\mathcal{R}(t)$ as the set of ancestors of t, where the right branch has been followed. This can be expressed as $\mathcal{A}(t) = \mathcal{R}(t) \cup \mathcal{L}(t)$. For instance, considering the tree in Figure A.1, we have $\mathcal{L}(5) = \{1\}$, $\mathcal{R}(5) = \{2\}$, and $\mathcal{A}(5) = \{1, 2\}$.

We divide the nodes in the tree into two distinct sets:

- Branch nodes: These nodes, denoted as Nodes $t \in \mathcal{T}_B = 1, ..., T/2$, apply a split of the form $a^T x < b$. Points that satisfy this split will follow the left branch in the tree, while those that do not will follow the right branch.

- Leaf nodes: These nodes, denoted as Nodes $t \in \mathcal{T}_L = T/2 + 1, ..., T$, are responsible for making a class prediction for each point that falls into the corresponding leaf node.

The proposed MIP model for optimal decision tree with asymmetric costs is presented as follows:

Sets:	
T	Set of all tree nodes
$\mathcal{P}(t)$	Parent node of node t
$\mathcal{A}(t)$	Set of ancestors of node t
$\mathcal{R}_t$	Set of ancestors of node t from the right branches
$\mathcal{L}_t$	Set of ancestors of node t from the left branches
Parameters:	
D	Depth of tree
n	number of observations in training set
p	Number of features
K	Number of defined classes
α	A value that controls to prune a node or not, so-called complexity parameter
N_{min}	Minimum number of points assigned to each branch node
ϵ	small constants to change the inequality to be non-strict
e_{ik}	1, if class of i-th point is k $\forall\, i \in n, k \in K$; 0, otherwise
mc_{kr}	cost matrix associated for misclassifications class k and r $\forall\, k, r \in K$
Decision Variables:	
C_{kt}	1, if class k assigned to node t $\forall\, k \in K, t \in \mathcal{T}_L$; 0, otherwise
L_t	1, if any point assigned to node t $\forall\, t \in \mathcal{T}_L$; 0, otherwise
Z_{it}	1, if point i assigned to node t $\forall\, i \in n, t \in \mathcal{T}_L$; 0, otherwise
a_{jt}	1, if feature j is evaluated in node t $\forall\, j \in p, t \in \mathcal{T}_B$; 0, otherwise
b_t	the right-hand-side of splitting rule in node t $\forall\, t \in \mathcal{T}_B$
d_t	1, if node t applies a split $\forall\, t \in \mathcal{T}_B$; 0, otherwise
N_{kt}	number of points of label k in node t $\forall\, k \in K, t \in \mathcal{T}_L$
M_{kt}	number of predicted points of label k in node t $\forall\, k \in K, t \in \mathcal{T}_L$
Q_{kt}	difference between actual and predicted points of label k in node t $\forall\, k \in K, t \in \mathcal{T}_L$
O_{kr}	number of actual points of label k that predicted as class r $\forall\, k, r \in K$

$$\min \quad \sum_{k \in K} \sum_{r \in K} mc_{kr}\, O_{kr} + \alpha \sum_{t \in \mathcal{T}_B} L_t \tag{A.1}$$

subject to

$$\sum_{j \in p} a_{jt} = d_t \qquad \forall\, t \in \mathcal{T}_B \tag{A.2}$$

$$b_t \leq d_t \qquad \forall\, t \in \mathcal{T}_B \tag{A.3}$$

$$d_t \leq d_{p_{(t)}} \qquad \forall\, t \in \mathcal{T}_B / 1 \tag{A.4}$$

$$Z_{it} \leq L_t \qquad \forall\, i \in n, t \in \mathcal{T}_L \tag{A.5}$$

$$\sum_{i \in n} Z_{it} \geq N_{min}\, L_t \qquad \forall\, t \in \mathcal{T}_L \tag{A.6}$$

$$\sum_{t \in \mathcal{T}_L} Z_{it} = 1 \qquad \forall\, i \in n \tag{A.7}$$

$$\sum_{j \in p} a_{jm} x_{ij} \geq b_m - (1 - Z_{it}) \qquad \forall\, i \in n, t \in \mathcal{T}_L, m \in \mathcal{R}_t \tag{A.8}$$

$$\sum_{j \in p} a_{jm}(x_{ij} - \epsilon_j + \epsilon_{min}) - \epsilon_{min} \leq b_m + (1 + \epsilon_{max}) \times \times (1 - Z_{it}) \qquad \forall\, i \in n, t \in \mathcal{T}_L, m \in \mathcal{L}_t \tag{A.9}$$

$$\sum_{k \in K} C_{kt} = L_t \qquad \forall\, t \in \mathcal{T}_L \tag{A.10}$$

$$N_{kt} = \sum_{i \in n} e_{ik}\, Z_{it} \qquad \forall\, k \in K, t \in \mathcal{T}_L \tag{A.11}$$

$$M_{kt} = \sum_{i \in n} C_{kt}\, Z_{it} \qquad \forall\, k \in K, t \in \mathcal{T}_L \tag{A.12}$$

$$Q_{kt} \geq N_{kt} - M_{kt} \qquad \forall\, k \in K, t \in \mathcal{T}_L \tag{A.13}$$

$$O_{kr} \geq \sum_{t \in \mathcal{T}_L} Q_{kt}\, C_{rt} \qquad \forall\, k \in K, r \in K \tag{A.14}$$

$$C_{kt}, L_t, Z_{it} \in [0, 1] \qquad \forall\, i \in n,\ k \in K,\ t \in \mathcal{T}_L \tag{A.15}$$

$$a_{it}, d_t \in [0, 1] \qquad \forall\, i \in n, t \in \mathcal{T}_B \tag{A.16}$$

$$b_t \in \mathbb{R} \qquad \forall\, t \in \mathcal{T}_B \tag{A.17}$$

To track the split applied at a branch node $t \in \mathcal{T}_B$, we introduce variables a_{pt} and b_{pt}. Our model is designed for univariate decision trees, such as CART, where each node's hyperplane split involves only a single variable. Therefore, we enforce this constraint by setting the elements of a_{pt} as binary variables that sum to 1.

In order to allow the option of not splitting at a branch node, we utilize indicator variables $d_t \in \{0, 1\}$ to track whether a node t applies a split. If a branch node does not apply a split, we set $a_{pt} = 0$ and $b_{pt} = 0$. Consequently, this forces all points to follow the right split at that node since the condition for the left split, $0 < 0$, is never satisfied. By doing so, we can terminate tree growth early without introducing additional variables to account for points reaching the branch node. Instead, we direct them all in the same direction down the tree to end up in the same leaf node. The constraints (A.2), and (A.3) ensure the enforcement of these conditions.

Subsequently, The constraint (A.4) imposes the hierarchical structure on the tree to ensure its integrity. This means that a branch node is only allowed to apply a split if its parent node also applies a split.

We introduce indicator variables Z_{it} to keep track of the points assigned to each leaf node in the decision tree. Additionally, we use indicator variables L_t to ensure that each leaf contains a minimum number of points, denoted by N_{min}. The constraint (A.5) enforces that each point x_i is assigned to a specific leaf node t and the constraint (A.6) guarantees that each leaf contains at least N_{min} points. Furthermore, The constraints (A.7) enforce that each point x_i is assigned to exactly one leaf node. By incorporating these constraints into the model, we ensure the proper allocation of data points to individual leaf nodes, maintaining the specified minimum number of points at each leaf while guaranteeing that each point is assigned to only one leaf in the decision tree.

To enforce the required splits dictated by the structure of the tree when assigning points to leaves, the constraints (A.8), and (A.9) are introduced that ensure the consistency between the assigned leaf nodes and the splits applied in the tree.

To make a single class prediction at each leaf node that contains points, we introduce binary variables C_{kt} to track the prediction of each node t. The variable

C_{kt} takes the value 1 if the class prediction at node t is k, and 0 otherwise, as shown in the constraints (A.10). These constraints ensure that for every leaf node containing points, there is precisely one class prediction made, and each point is associated with a class prediction that corresponds to the leaf node it is assigned to in the decision tree. By doing so, we achieve the objective of making a single class prediction at each leaf node that contains points.

We define N_{kt} as the count of data points with label k in node t, constraint (A.11). To facilitate this, we introduce indicator parameters e_{ik}, which are set to 1 if the class of the i-th data point is k, for all i in the set of data points otherwise, e_{ik} is set to 0. Moreover, we utilize M_{kt} to represent the count of data points predicted as class k in node t, constraint (A.12). Additionally, we introduce Q_{kt} to quantify the discrepancy between the actual count and the predicted count of data points with class label k in node t, constraint (A.13). Lastly, O_{kr} serves to indicate the number of data points belonging to class k that are predicted as class r, constraint (A.14).

The objective function (A.1) minimizes the misclassification cost associated with different error types. Additionally, the complexity of the tree can be controlled by adjusting the value of α; increasing α prevents some nodes from being split. The solution to the model provides the values of the variables a_{jt} and b_t, which are used to construct the decision tree and assign labels to unseen events.

Appendix B

OPTIMAL HYPERPLANE DECISION TREE (OHDT)

The goal is to find the simplest linear rule to assign a decision to each data point i. So, we introduce two binary variables Z_{iA} and Z_{iB} and apply them to the above condition as follows. Here, a_j is continuous, taking values in the interval $[-1, 1]$. Moreover, β_0 can assume both positive and negative values within the range $[-1, 1]$. So, the range of $ax - \beta_0$ is restricted to $[-2, 2]$, necessitating $M = 2$ to guarantee that the constraint is inherently met when $Z_{it} = 0 \;\; \forall\, t \in A, B$.

$$\sum_{j \in p} a_j x_{ij} < \beta_0 + 2(1 - Z_{iA}) \qquad \forall\, i \in n \tag{B.1}$$

$$\sum_{j \in p} a_j x_{ij} \geq \beta_0 - 2(1 - Z_{iB}) \qquad \forall\, i \in n \tag{B.2}$$

$$Z_{iA} + Z_{iB} = 1 \qquad \forall\, i \in n \tag{B.3}$$

In the preceding equations, it is denoted that when the decision denoted as A is applied to the data point i, the variable Z_{iA} assumes a value of one, while for all other cases, it takes on the value of zero. Similarly, when the decision labeled as B is assigned to the data point i, Z_{iB} is set to one. In order to enforce the constraint that each data point receives only a single decision assignment, Equation B.3 is employed for this purpose.

Furthermore, it is assumed that there must be assigned at least one data point to each decision category A or B. This constraint is introduced by incorporating the following equation:

$$\sum_{i \in n} Z_{it} \geq 1 \qquad \forall\, t \in A, B \tag{B.4}$$

To regulate the number of input features utilized in formulating the linear decision rule, it is imperative to introduce the following two constraints. The objective is to select a set of D features with the intention of reducing model complexity, eliminating extraneous features, and enhancing the model's resilience. In this context, the binary variable V_j plays a pivotal role in constraining the number of features and M is considered as a big penalty value, as elucidated in the ensuing equations:

$$-MV_j \leq a_j \leq MV_j \qquad \forall\, j \in p \tag{B.5}$$

$$\sum_{j \in p} V_j = D \tag{B.6}$$

As previously explained, the overarching goal is to minimize the cost associated with assigning an incorrect decision label to each data point indexed as i, upon revelation of its true label. To achieve this objective, we incorporate the ensuing constraints:

$$\sum_{k \in K} C_{kt} = 1 \qquad \forall\, t \in \{A, B\} \tag{B.7}$$

$$g_{kt} = \sum_{i \in n} E_{ik} Z_{it} \qquad \forall\, k \in K, t \in \{A, B\} \tag{B.8}$$

$$h_{kt} = \sum_{i \in n} C_{kt} Z_{it} \qquad \forall\, k \in K, t \in \{A, B\} \tag{B.9}$$

$$q_{kt} \geq g_{kt} - h_{kt} \qquad \forall\, k \in K, t \in \{A, B\} \tag{B.10}$$

$$O_{kr} \geq \sum_{t \in \mathcal{T}_L} q_{kt} C_{rt} \qquad \forall\, k \in K, r \in K \tag{B.11}$$

To make a single decision class prediction at each node that contains points, we introduce binary variables C_{kt} to track the prediction of each node t. The variable C_{kt} takes the value 1 if the class prediction at node t is k, and 0 otherwise. These constraints ensure that for every node containing points, there is precisely one class prediction made, and each point is associated with a class prediction that corresponds to the node it is assigned to in the decision tree. By doing so, we achieve the objective of making a single class prediction at each leaf node that contains points.

We define g_{kt} as the count of data points with label k in node t. To facilitate this, we introduce indicator parameters E_{ik}, which are set to 1 if the class of the i-th data point is k, for all i in the set of data points otherwise, E_{ik} is set to 0. Moreover, we utilize h_{kt} to represent the count of data points predicted as class k in node t. Additionally, we introduce q_{kt} to quantify the discrepancy between the actual count and the predicted count of data points with class label k in node t. Lastly, O_{kr} serves to indicate the number of data points belonging to class k that are predicted as class r.

The optimization model for attaining an optimal decision based on the observed label is as presented below:

Model:

$$\min \quad \sum_{k\in K}\sum_{r\in K} O_{kr}\, mc_{kr} \tag{B.12}$$

subject to

$$\sum_{j\in p} a_j x_{ij} < \beta_0 + 2(1 - Z_{iA}) \qquad \forall\, i \in n \tag{B.13}$$

$$\sum_{j\in p} a_j x_{ij} \geq \beta_0 - 2(1 - Z_{iB}) \qquad \forall\, i \in n \tag{B.14}$$

$$Z_{iA} + Z_{iB} = 1 \qquad \forall\, i \in n \tag{B.15}$$

$$\sum_{i\in n} Z_{it} \geq 1 \qquad \forall\, t \in \{A, B\} \tag{B.16}$$

$$\sum_{k\in K} C_{kt} = 1 \qquad \forall\, t \in \{A, B\} \tag{B.17}$$

$$-\, MV_j \leq a_j \leq MV_j \qquad \forall\, j \in p \tag{B.18}$$

$$\sum_{j\in p} V_j = D \tag{B.19}$$

$$g_{kt} = \sum_{i\in n} E_{ik}\, Z_{it} \qquad \forall\, k \in K, t \in \{A, B\} \tag{B.20}$$

$$h_{kt} = \sum_{i\in n} C_{kt}\, Z_{it} \qquad \forall\, k \in K, t \in \{A, B\} \tag{B.21}$$

$$q_{kt} \geq g_{kt} - h_{kt} \qquad \forall\, k \in K, t \in \{A, B\} \tag{B.22}$$

$$O_{kr} \geq \sum_{t\in\{A,B\}} q_{kt}\, C_{rt} \qquad \forall\, k \in K, r \in K \tag{B.23}$$

$$a_j, b_0 \in [-1, 1] \qquad \forall\, j \in p \tag{B.24}$$

$$C_{kt}, Z_{it} \in \{0, 1\} \qquad \forall\, i \in n,\, k \in K,\, t \in \{A, B\} \tag{B.25}$$

$$V_j \in \{0, 1\} \qquad \forall\, j \in p \tag{B.26}$$

Appendix C

DATA DESCRIPTION AND PROCESSING

The performance of the proposed methods for finding the efficient decision rule is demonstrated using four distinct publicly available data sets:

The 1^{st} data set is obtained from a semiconductor manufacturing process (SECOM) and is available on the UCI machine learning repository (McCann and Johnston, 2008a). In this data set, features correspond to various process measurements in a semiconductor wafer fabrication process. The SECOM data set contains 1567 observations with 590 features. Each observation corresponds to a wafer, and every wafer has its yield as binary data: pass or fail (0 or 1).

The 2^{nd} data set pertains to breast cancer diagnosis for 569 patients, comprising 30 features which is available on the UCI machine learning repository (Wolberg and Street, 1995). The response variable, diagnosis, categorizes cancer into two types: malignant (1) and benign (2), which benign is less dangerous than malignant type. These features are derived from digitized images of fine needle aspirates (FNA) of breast masses, encapsulating characteristics of the cell nuclei present in the image.

The 3^{rd} data set focuses on the detection of heart disease in patients, with the presence of the disease quantified as an integer value ranging from 0 (no presence) to 4 which is available on the UCI machine learning repository (Janosi and Detrano, 1988). Historical experiments utilizing the Cleveland database have primarily aimed at distinguishing between the presence (values 1, 2, 3, 4) and absence (value 0) of heart disease. This particular data set is composed of 13 features, derived from various examinations of 303 patients.

The 4^{th} data set contains cases from a study that was conducted between 1958 and 1970 at the University of Chicago's Billings Hospital on the survival of 306 patients who had undergone surgery for breast cancer, which is available on the UCI

machine learning repository (Haberman, 1999). This data set is characterized by four attributes: the age of the patient at the time of surgery, the year of operation, the number of positive axillary nodes detected, and the survival status post-surgery. The survival status is categorized into two classes: '1' indicating patients who survived for 5 years or more after the operation, and '2' for those who passed away within 5 years following their surgery.

C.1 Pre-processing and Feature Selection

Before any analysis, the data sets must undergo preprocessing, crucial for handling large data with missing values ('NaN'). In this study, missing values are imputed with the feature's mean. Additionally, features with constant values across all instances are removed using a variance threshold, leading to the deletion of 116 constant features from the SECOM data set. No constant features were identified in the other data sets.

To address class imbalance, such as the disproportionate number of pass (1,463) and fail (104) wafers in SECOM, data balancing techniques like the Synthetic Minority Oversampling Technique (SMOTE) are employed. This approach, implemented via the imblearn library (Lemaître et al., 2017), generates new observations for the minority class, facilitating a more robust analysis. The other data set did not exhibit significant imbalance, thus no additional balancing techniques were necessary.

Given the varying scales of features, normalization is necessary to enhance model efficiency, particularly in optimization models. Here, we normalize each feature by dividing by its maximum value.

Solving Mixed Integer Programming (MIP) models can be computationally intensive. For instance, the proposed MIP model for the full breast cancer data set required 884 seconds to solve when the tree depth was set to 2 (Figure C.1). To reduce computational time, we applied feature selection with cross-validation, focusing on the most impactful features identified by a standard decision tree model. This approach reduced computational time by approximately 60%, from 884 seconds to 353 seconds, without compromising model accuracy. The selected features were

then incorporated into the proposed MIP model. This process was applied to all data sets to achieve results within a reasonable runtime.

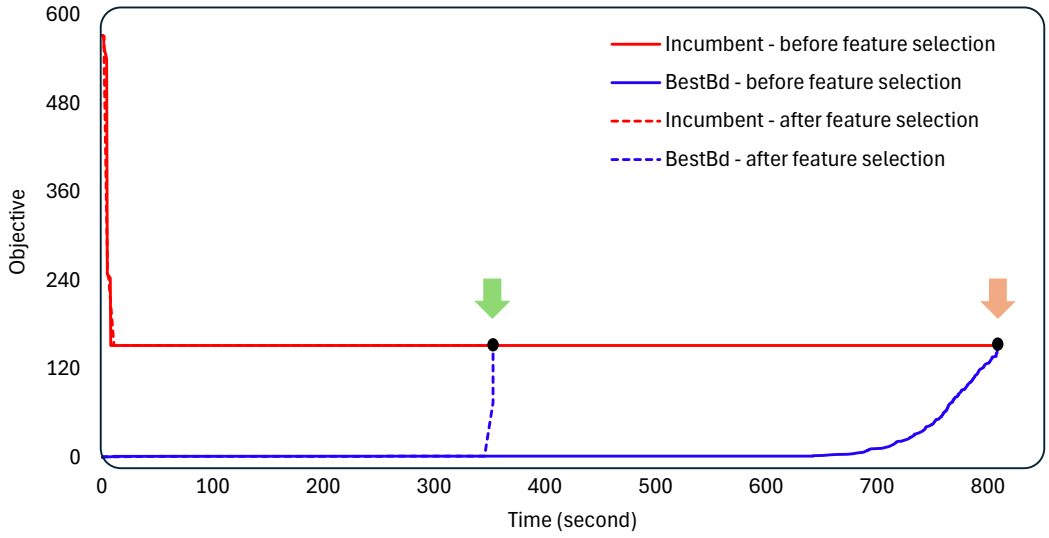

Figure C.1: Impact of feature selection on run time

C.2 Ratio of Asymmetric Cost

In all the proposed methods, asymmetric cost matrix plays a crucial role in determining the decision rule, particularly in relation to the overall cost associated with different error types. To systematically assess the performance of the proposed models, we introduce three levels of asymmetric cost, denoted as D. At $D = 0$, the cost ratio between the highest and lowest values is set at 1:1, representing a symmetric or balanced cost environment. In this scenario, we expect similar outcomes between the proposed models and existing heuristic approaches. $D = 1$ introduces a cost ratio of approximately 5:1, where the highest cost is five times greater than the lowest. $D = 2$ further increases this ratio to approximately 50:1, where the highest cost is fifty times greater than the lowest. $D = 3$ further increases this ratio to approximately 10000:1, where the highest cost is ten thousand times greater than the lowest.

For each data set, the cost matrices are provided for $D = 0$, $D = 1$, $D = 2$, and $D = 3$. These matrices are critical in understanding the trade-offs made by the decision-making models when different costs are assigned to different types of errors. The matrices are arranged in a row for each data set, with the specific values reflecting the relative importance or penalty of false positives and false negatives, among other error types.

SECOM data set

$$\text{D=0: } \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \quad \text{D=1: } \begin{bmatrix} 0 & 9 \\ 6 & 0 \end{bmatrix} \quad \text{D=2: } \begin{bmatrix} 0 & 900 \\ 6 & 0 \end{bmatrix} \quad \text{D=3: } \begin{bmatrix} 0 & 60000 \\ 6 & 0 \end{bmatrix}$$

Breast Cancer data set

$$\text{D=0: } \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \quad \text{D=1: } \begin{bmatrix} 0 & 2 \\ 10 & 0 \end{bmatrix} \quad \text{D=2: } \begin{bmatrix} 0 & 20 \\ 1000 & 0 \end{bmatrix} \quad \text{D=3: } \begin{bmatrix} 0 & 20 \\ 20000 & 0 \end{bmatrix}$$

Heart Disease data set

$$\text{D=0: } \begin{bmatrix} 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 \end{bmatrix} \quad \text{D=1: } \begin{bmatrix} 0 & 2 & 5 & 10 & 15 \\ 4 & 0 & 6 & 8 & 14 \\ 9 & 8 & 0 & 10 & 13 \\ 15 & 14 & 12 & 0 & 7 \\ 20 & 18 & 16 & 14 & 0 \end{bmatrix} \quad \text{D=2: } \begin{bmatrix} 0 & 2 & 5 & 100 & 150 \\ 4 & 0 & 6 & 8 & 14 \\ 9 & 8 & 0 & 100 & 130 \\ 150 & 140 & 120 & 0 & 7 \\ 2000 & 1800 & 1600 & 1400 & 0 \end{bmatrix}$$

$$\text{D=3: } \begin{bmatrix} 0 & 2 & 5 & 100 & 150 \\ 4 & 0 & 6 & 8 & 14 \\ 9 & 8 & 0 & 1000 & 1300 \\ 1500 & 1400 & 1200 & 0 & 70 \\ 20000 & 18000 & 16000 & 14000 & 0 \end{bmatrix}$$

Haberman data set

$$\text{D=0: } \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \quad \text{D=1: } \begin{bmatrix} 0 & 2 \\ 10 & 0 \end{bmatrix} \quad \text{D=2: } \begin{bmatrix} 0 & 20 \\ 1000 & 0 \end{bmatrix} \quad \text{D=3: } \begin{bmatrix} 0 & 20 \\ 200000 & 0 \end{bmatrix}$$

Appendix D

THE MODEL WITHOUT FEATURES AND WITH FULL INFORMATION

Let us review the results for the special case of a model without features and full distributional information that are used in the study. There is a rich literature for this case, and we exploit some of these results for the case with limited data and feature information. The expected cost minimization problem is under full information on the distributions of the random variables $Y$and D is:

$$\min_{Q} z(Q) = \mathbb{E}\left[\left(b(D - QY)^{+} + h(QY - D)^{+}\right)\right] \tag{D.1}$$

The solution to the problem (D.1) with full distributional information on D and Y dates back to Shih (1980), Gerchak et al. (1988) and Henig and Gerchak (1990). We take the point of view from Okyay et al. (2014), which addresses the case with possibly dependent random demand and yield. Our objective function is slightly different than the one in Okyay et al. (2014), but we can follow the approach therein to differentiate the right hand side of (D.1) with respect to Q and conclude that the optimal lot size (order quantity) Q^* is the solution of:

$$\frac{\mathbb{E}[Y \mathbb{1}_{\{D \leq Q^* Y\}}]}{\mathbb{E}[Y]} = \frac{b}{b+h} \tag{D.2}$$

Note that $\mathbb{1}_{\{D \leq Q^* Y\}}$ corresponds to the indicator function of the subscripted event and takes the value of 1 whenever $D \leq Q^* Y$ and takes the value of 0 otherwise. Okyay et al. (2014) show that whenever D and Y are jointly continuously distributed random variables, a solution to the above equation exists and that it is unique under an additional regularity condition. Unfortunately, there appears to be no obvious closed form for the optimal lot size in general. But, (D.2) can be solved numerically.

Appendix E

PROOF OF PROPOSITION 1

Proof of Proposition 1: Consider two lot size decisions, Q_1 and Q_2, where $Q_1 < Q_2$. To establish the theorem, evaluate the difference in costs, $C(Q_2;Y) - C(Q_1;Y)$, across three distinct scenarios: when demand divided by yield, D/Y, falls into one of the following intervals - less than Q_1, between Q_1 and Q_2, and greater than Q_2. For clarity, these intervals are redefined in terms of actual demand, D, as $D < Q_1Y$, $Q_1Y \leq D \leq Q_2Y$, and $D > Q_2Y$, considering that the yield, Y, ranges from 0 to 1.

We can express the following for the cost difference:

$$\begin{aligned} C(Q_2;D) - C(Q_1;D) &= -b(Q_2-Q_1)y\mathbb{1}_{\{D>Q_2Y\}} + h(Q_2-Q_1)Y\mathbb{1}_{\{D<Q_1Y\}} \\ &\quad +[Q_1b + Q_2h - (b+h)\frac{D}{Y}]Y\mathbb{1}_{\{Q_1Y\leq D\leq Q_2Y\}} \end{aligned} \tag{E.1}$$

In (E.1), the domain where $D > Q_2Y$ can be expressed in terms of the left-hand domain $D > Q_1Y$, with the additional requirement of subtracting the supplementary cost incurred in the second region for the range where $Q_1Y \leq D \leq Q_2Y$. Consequently, we arrive at the following formulation:

$$\begin{aligned} C(Q_2;D) - C(Q_1;D) &= -b(Q_2-Q_1)Y\mathbb{1}_{\{D>Q_1Y\}} + h(Q_2-Q_1)Y\mathbb{1}_{\{D<Q_1Y\}} \\ &\quad +[(b+h)(Q_2-\frac{D}{Y})]Y\mathbb{1}_{\{Q_1Y\leq D\leq Q_2Y\}} \end{aligned} \tag{E.2}$$

Now, let us assume that $Q_1 = Q^*$ represents the optimal lot size. From (D.2), it is deduced that:

$$\mathbb{E}[Y\mathbb{1}\{D < Q_1Y\}] = \frac{\mathbb{E}[Y]\cdot b}{b+h}$$

and

$$\mathbb{E}[Y\mathbb{1}\{D > Q_1 Y\}] = \frac{\mathbb{E}[Y]\cdot h}{b+h}.$$

Substituting these expressions into (E.2) results in the cancellation of the initial two terms. Consequently, the expected cost difference simplifies to:

$$EC(Q_2;D) - EC(Q_1;D) = (b+h)\mathbb{E}(Q_2 - \frac{D}{Y})]\mathbb{E}[Y\mathbb{1}_{\{Q_1 Y \le D \le Q_2 Y\}}] \tag{E.3}$$

Finally, we repeat similar steps for the area denoted as $D < Q_1 Y$ which can be expressed utilizing the region $D < Q_2 Y$. However, it is imperative to account for and subtract the additional incurred cost associated with the second region in instances where $Q_1 Y \le D \le Q_2 Y$. Consequently, we have:

$$\begin{aligned} C(Q_2;D) - C(Q_1;D) = & -b(Q_2 - Q_1)Y\mathbb{1}_{\{D > Q_2 Y\}} + h(Q_2 - Q_1)Y\mathbb{1}_{\{D < Q_2 Y\}} \\ & - [(b+h)\left(\frac{D}{Y} - Q_1\right)]Y\mathbb{1}_{\{Q_1 Y \le D \le Q_2 Y\}} \end{aligned} \tag{E.4}$$

Now, if we assume that $Q_2 = Q^*$ is the optimal lot size. We can use the optimality conditions to (D.2) to further simplify (E.4). Consequently, the expected cost difference is obtained as follows:

$$EC(Q_1;D) - EC(Q_2;D) = (b+h)\mathbb{E}(\frac{D}{Y} - Q_1)]\mathbb{E}[Y\mathbb{1}_{\{Q_1 Y \le D \le Q_2 Y\}}] \tag{E.5}$$

Appendix F

ESTIMATION AND OPTIMIZATION WITHOUT FEATURES

A simple standard data-driven approach is to ignore the feature information and formulate and solve the empirical risk minimization problem using demand and yield realizations in the training sample within a Sample Average Approximation (SAA) framework. To this end, let us define auxiliary decision variables u_i and o_i that correspond to the underage and overage quantities for sample S for observation i. The formulation below finds the optimal quantity that would minimize the empirical cost in the training sample:

$$\min \hat{z}(Q) = \frac{1}{n}\left(\sum_{i=1}^{n}(bo_i + hu_i)\right) \tag{F.1}$$

$$u_i \geq d_i - Qy_i \qquad \forall\, i,\ i = 1, 2, \ldots, n \tag{F.2}$$

$$o_i \geq Qy_i - d_i \qquad \forall\, i,\ i = 1, 2, \ldots, n \tag{F.3}$$

$$u_i, o_i \geq 0 \qquad \forall\, i,\ i = 1, 2, \ldots, n \tag{F.4}$$

$$Q \geq 0 \tag{F.5}$$

The solution of the above linear program gives the lot size Q^*_{SAA} that minimizes the empirical cost in the training sample when the feature data is ignored. One could then use this single lot size for future ordering decisions (i.e. out of sample). This is a reasonable data-based solution and the formulation is a linear program which can be efficiently solved at a large scale but does not exploit the information that may be provided by the features. We use this SAA solution as a benchmark for other methods that use features.

Remark: Instead of solving the above LP, one can use the optimality condition

(3.13) taking the empirical joint distribution as the estimator for the joint distribution of D and Y. However, the LP allows us to implement other data-based approaches as outlined in the next subsection.

Appendix G

ESTIMATION AND OPTIMIZATION WITH LINEAR RULES

G.1 A Linear Rule for Demand and Yield

Let us assume that the optimal lot size is a linear function of the demand and yield features and can be expressed in the following form:

$$Q_{LR,i} = \beta_0 + \beta_1 x_{i,1} + ... + \beta_r x_{i,r} + \beta_{r+1} u_{i,1} + ... + \beta_{r+s} u_{i,s} \tag{G.1}$$

As already noted in Ban and Rudin (2019), the linear lot sizing rule can be extended further by taking non-linear transformations of the features. The following LP uses the values of the predictors and the corresponding yield and demand observations on the training data to compute the optimal linear rule (determined by the coefficients $(\beta_0, \beta_1, ..., \beta_{r+s})$ for the order quantity in (G.1).

$$\min \hat{z}(\beta_0, \beta_1, \ldots, \beta_{r+s}) = \frac{1}{n}\left(\sum_{i=1}^{n}(bo_i + hu_i)\right) \tag{G.2}$$

$$\text{s.t.} \quad u_i \geq d_i - Q_i y_i \qquad \forall\, i = 1, 2, \ldots, n \tag{G.3}$$

$$o_i \geq Q_i y_i - d_i \qquad \forall\, i = 1, 2, \ldots, n \tag{G.4}$$

$$Q_i = \beta_0 + \sum_{j=1}^{r} \beta_j x_{i,j} + \sum_{k=1}^{s} \beta_{k+r} u_i^k \qquad \forall\, i = 1, 2, \ldots, n \tag{G.5}$$

$$\beta_j \text{ unrestricted} \qquad \forall\, j = 0, 1, \ldots, r+s$$

$$u_i, o_i, Q_i \geq 0 \qquad \forall\, i = 1, 2, \ldots, n$$

Let the optimal values of the decision variables be: $\beta_0^*, \beta_1^*, ..., \beta_{r+s}^*$. The decision

rule for a new observation in the sample is:

$$Q_{new} = \beta_0^* + \sum_{j=1}^{r} \beta_j^* x_{new,j} + \sum_{k=r+1}^{r+s} \beta_k^* u_{new}^{k-r}$$

G.2 A Linear Rule without Yield Features (ERM1)

It is plausible that for a given yield value, the optimal lot size is a linear function of the demand features. Further, if yield features are not available or cannot be transformed easily into a linear model (G.1) simplifies to:

$$Q_i = \beta_0 + \sum_{j=1}^{r} \beta_j x_{i,j}$$

In addition, if demand and yield are known to be independent, we can expand the sample by considering all possible combinations of observations of demand (d_i) and yield (y_k). This would lead to the following alternative LP formulation:

$$\min \hat{z}(Q_i, \beta_0, \beta_1, \ldots, \beta_r) = \frac{1}{n^2} \left(\sum_{i=1}^{n} \sum_{k=1}^{n} (bo_{ik} + hu_{ik}) \right)$$

$$\begin{aligned}
\text{s.t.} \quad & u_{ik} \geq d_i - Q_i y_k && \forall i, k = 1, 2, \ldots, n && \text{(G.6)} \\
& o_{ik} \geq Q_i y_k - d_i && \forall i, k = 1, 2, \ldots, n && \text{(G.7)} \\
& Q_i = \beta_0 + \sum_{j=1}^{r} \beta_j x_{i,j} && \forall i = 1, 2, \ldots, n && \text{(G.8)} \\
& u_{ik}, o_{ik} \geq 0 && \forall i, k = 1, 2, \ldots, n && \text{(G.9)} \\
& \beta_j \text{ unrestricted} && \forall j = 0, 1, \ldots, r \\
& Q_i \geq 0 && \forall i = 1, 2, \ldots, n
\end{aligned}$$

Let the optimal values of the decision variables be: $\beta_0^*, \beta_1^*, ..., \beta_r^*$. The decision rule for a new observation in the sample is:

$$Q_{ERM1}(new) = \beta_0^* + \sum_{j=1}^{r} \beta_j^* x_{new,j}$$

Appendix H

DATA AND BRIEF SUMMARY OF NON-LINEAR METHODS

H.1 Data Description and Processing

The performance of the proposed strategies for finding the optimal lot size is demonstrated on two different types of data sets: some synthetic data sets and a modified version of the SECOM data set, which is publicly available (McCann and Johnston, 2008b) (and partially presented in Table 3.1).

H.1.1 Synthetic Data Sets

We generate a data set of 2000 observations with 10 features for the demand and 10 features for the supply side to evaluate the performance of the strategies in a controlled environment. In particular, to observe the effects of model reduction (i.e. elimination of useless features to manage the bias-variance trade-off), some of our generated features are dummies that do not have any dependence on the responses. We generate the demand according to a linear model:

$$d_i = a_0 + \sum_{j=1}^{r} a_j x_{i,j} + \epsilon_i$$

where ϵ_i are normally distributed with mean zero and standard deviation, $\sigma_i = 5$. The number of observations equals $n = 2000$ and the number of demand features equals $r = 10$. Here, $a_0 = 0, a_1 = 10, a_2 = -2, a_3 = 20, a_4 = 5, a_5 = 1, a_6 = a_7 = a_8 = a_9 = a_{10} = 0$. We draw, $x_{i,j}$'s independent from a uniformly distribution between 0 and 10. Note that we assume that only the first five features have a dependence on the demand and the remaining five features enable us to check whether the model identification during training is correct.

Moreover, we generate the feature-based random yield based on a logit form with the following parameters that give us continuous values between 0 and 1:

$$y_i = \frac{e^{b_0+\sum_{k=1}^{s} b_k y_{i,k}+\epsilon_i}}{1+e^{b_0+\sum_{j=1}^{s} b_j y_{i,k}+\epsilon_i}}$$

where ϵ_i are normally distributed with mean zero and standard deviation, $\sigma_i = 0.05$ and the number of yield features $s = 10$. In this equation, y_i^j's are uniformly distributed between 0 and 1. Here, $b_0 = 0, b_1 = 0.5, b_2 = -0.5, b_3 = 1, b_4 = -1, b_5 = 2, b_6 = b_7 = b_8 = b_9 = b_{10} = 0$. We assume that only the first five features have a dependence on the yield.

H.1.2 A Modified SECOM Data Set for Yield Features

To represent a realistic data set for yield-related features, we employ the SECOM data set that is obtained from a semiconductor manufacturing process and is available on the UCI machine learning repository (McCann and Johnston, 2008b). In this data set, features correspond to various process measurements in a semiconductor wafer fabrication process. The SECOM data set contains 1567 observations with 590 features. Each observation corresponds to a wafer, and every wafer has its yield as binary data: pass or fail. There are 1463 pass wafers and 104 fail wafers, which means the data set is imbalanced. For such high yield rates, the optimization problem is not critical. But Diefenbach (2023) reports that in certain cases the observed yield rates in semi-conductor manufacturing might be much lower. To keep the feature information from the data set with the goal of testing optimization for lower yield rates, we sample the data set to generate a more balanced set with a lower yield rate. This is done by synthesizing new observations from the existing observations, a frequently used data augmentation approach. In particular, we use the Synthetic Minority Oversampling Technique, or SMOTE for short. By using this ML tool (used library from imblearn (Lemaître et al., 2017)), we balance the SECOM date set for further analysis. Since SECOM data set only considers the yield side, we add four more features regarding the data and time of production, and we assume that those correspond to the demand features.

Finally, the case in (Diefenbach, 2023) considers batch production and has uncertain yield rates similar to the models presented in Sections 3.3 and 3.4. We therefore convert the 0-1 outcome to a yield rate (between 0 and 1) using a Random Forest estimator on the original data. A part of the modified data set can be seen in Table H.1. In contrast with Table 3.1, the first column now reports a likelihood estimate, which we use as a proxy for the the yield rate of a batch. We present further details on feature reduction in the next subsection.

Table H.1: A part of SECOM data: estimated yield ($\hat{y}_i$) by using random forest classifier and selected 21 features ($u_{i,k}$) based on the most important ones.

Observation i	**Yield** ($\hat{y}_i$)	$u_{i,1}$	$u_{i,2}$	$u_{i,3}$	$u_{i,4}$	**...**	$u_{i,20}$	$u_{i,21}$
1	0.98	3030.93	10.04	61.29	-1.73	...	0.32	3.06
2	0.98	3095.78	9.26	78.25	0.81	...	0.27	2.01
3	0.22	2932.61	9.31	14.37	23.82	...	0.19	4.09
4	0.89	2988.72	9.69	76.90	24.38	...	0.17	2.90
5	0.80	3032.24	10.34	76.39	-12.29	...	0.22	3.18
6	0.71	2946.25	9.24	59.94	28.30	...	0.32	2.26
7	0.99	3030.27	10.00	74.46	2.11	...	0.20	2.20
8	0.93	3058.88	9.68	78.09	29.13	...	0.35	92.59
9	0.89	2967.68	9.70	61.10	26.98	...	0.32	2.49
⋮	⋮	⋮	⋮	⋮	⋮	...	⋮	⋮
1752	0.19	2936.42	10.66	64.25	22.00	...	0.51	64.87
1753	0.20	2911.79	9.93	78.12	2.39	...	0.49	3.39
1754	0.92	2993.45	9.95	79.25	-3.148	...	0.30	3.40

H.2 Pre-processing and Feature Selection

The SECOM data reports measurements/signals that are recorded along with the corresponding results of the quality test (0 or 1). We use the reported measurements

as features corresponding to the yield. Prior to any analysis, it is necessary to clean the original data set. As in almost all big data sets, the existence of missing values (or 'NaN') is relatively common. We substitute such values with the average value of each feature (column). Another problem is the existence of some features with constant values. These features are eliminated by a variance-threshold tool. In all, 116 columns are detected and deleted as constant features.

After the initial cleaning, there are still 474 remaining features, a large number with respect to the number of observations. We next focus on reducing the dimension of the data set and determining the most significant features. To reduce the model, we then utilize the Recursive Feature Elimination with Cross-Validation (RFECV) (Czako et al., 2021) method and the random forest algorithm as a prediction tool. The feature selection by RFECV is time-consuming, but the analysis enables us to reduce the number of features to 21. The fitted random forest classification with respect to 21 selected top features and binary yield as response gives the accuracy of 93.6% on the original data (as in Table 3.1. Note that Lee et al. (2019) also investigates the SECOM data set with a data mining perspective, but for a better control of the approach and to benefit from more recent tools, we implemented our own model reduction.

As a consistency check, We also perform RFECV on the synthetic data set where there are 10 significant features and 10 irrelevant (dummy) ones. The method correctly identifies the 10 significant features and eliminates the rest.

Here, we provide some information on the methods used to estimate yield as a function of the features.

H.3 Random Forests

Random forest algorithm (RF) (Breiman, 2001) is a flexible tool for modeling interactions in high dimensional data by constructing a large number of regression trees and averaging their predictions to arrive at a point prediction. A simple structure of the random forest algorithm is shown in Figure 3.4.1. The typical implementation of the RF algorithm is related to some local averaging algorithms like kernel

estimators (Scornet, 2015) or nearest neighbors method (Biau and Devroye, 2010). The RF method can be viewed as an adaptive neighborhood regression procedure that where the prediction (estimation of the conditional mean) is formulated as a weighted average of the observed response variables in the neighbourhood (Lin and Yongho, 2006).

In this study, for extracting the distribution of the demand and yield based on the given information in the historical data set, we generate $k = 100$ random regression trees for each estimation, which give us a rule for estimating the demand and yield. The remaining parameters of the random forest regression model are configured with their default values. The *max depth* parameter is set to *None*, allowing nodes to expand until all leaves are pure. Additionally, the *max features* parameter is set to the total number of features in the data set, ensuring all available features are considered during each tree construction. We then use a different bootstrapped training sample to generate each tree to obtain 100 different predictions.

H.4 Gradient Boosting

Gradient boosting regression (GBR) is a powerful and flexible tool for modeling interactions in high-dimensional data by sequentially constructing an ensemble of weak learners, typically regression trees, and combining their predictions to arrive at a point prediction. The typical implementation of the GBR algorithm relates to boosting methods, where the prediction model is built in a stage-wise fashion, and new trees are added to correct the residual errors of the previous ones (Friedman, 2001b). GBR can be viewed as a forward learning ensemble method that optimizes a loss function by adding predictors to the ensemble, one at a time, thereby improving the accuracy of the model (Natekin and Knoll, 2013).

In this study, we generate an ensemble of $n = 100$ regression trees to estimate the yield rate in the out-of-sample set. Each tree is constructed to predict the residuals of the previous trees, which helps in minimizing the overall error. The remaining parameters of the gradient boosting regression model are configured with their default values. The *learning rate* parameter is set to *0.1*, which controls the contribution

of each tree to the final prediction. Additionally, the *max depth* parameter is set to *3*, limiting the number of nodes in each tree to prevent overfitting. We then use different subsets of the training data to generate each tree to obtain 100 different predictions.

H.5 Artificial Neural Network (ANN)

Artificial neural network (ANN) based predictors are machine learning methods that are at the heart of deep learning algorithms. The structure and the title are inspired by the human brain, mimicking the way that biological neurons signal to one another. A typical ANN consists of node layers containing an input layer, one or multiple hidden layers, and an output layer. Each neuron (node) connects to another and has an associated weight and threshold. If the output of any individual node is above the specified threshold value, that node is activated, sending data to the next layer of the network. Otherwise, no data is passed along to the next layer of the network. A simple structure of the artificial neural network is shown in Figure H.1.

ANN is fitted to a given set of data points by changing the weights of the arcs that connect the nodes. ANN relies on training data to learn and improve its accuracy over time. However, once these learning algorithms are fine-tuned for accuracy, they are powerful tools for prediction and classification.

In this study, an Artificial Neural Network (ANN) model is designed and implemented for analysis. The architecture of the ANN includes one input layer, followed by three hidden layers, and an output layer consisting of a single neuron.

The hidden layers are structured with 4, 6, and 4 units respectively, providing complexity and depth to the model. The kernel initializer parameter is set to *normal*, ensuring appropriate weight initialization. The optimizer utilized for training the model is *Adam*, a popular optimization algorithm in deep learning.

To train the model, a batch size of 20 was employed, which determines the number of samples processed before the model's internal parameters are updated. The training process is repeated for 100 epochs, representing the number of times

the entire training data set is passed forward and backward through the model. This ANN architecture and training configuration were chosen to effectively capture and learn complex patterns in the input data, facilitating accurate predictions and analysis.

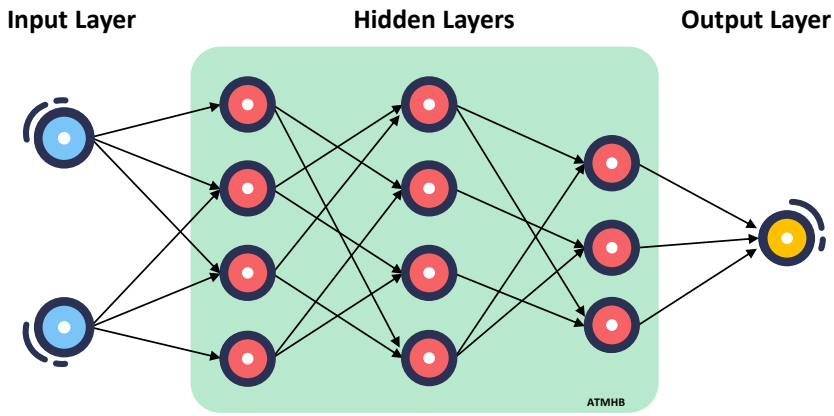

Figure H.1: Artificial Neural Network Structure

Appendix I

RELATIONSHIP BETWEEN THE LOT SIZE AND EXPECTED YIELD RATE

To demonstrate the relationship between the optimal lot size and the expected yield, we generated 20 series of cases with varying expected yields ranging from 0.1 to 1, which in each case there are 300 observations. In addition, we assumed that the demand is fixed at 100 units, with overage and underage costs set at 10 and 15, respectively. The yield values are generated using the following rule:

$$y_{ij} = a_j + a_0\epsilon_{ij}$$

Where i refers to each observation and j refers to each set. The ϵ_{ij} values are normally distributed with a mean of zero and a standard deviation, $\sigma = 1$. It is assumed that $a_0 = 5$ and a_j's values are real numbers between 0.05 and 1, with a step size of 0.05. Accordingly, the expected yield for each set is calculated. Then, we determined the corresponding optimal order quantities for each case. As depicted in Figure I.1, as the expected yield increases, the optimal lot size decreases, though not in a linear manner.

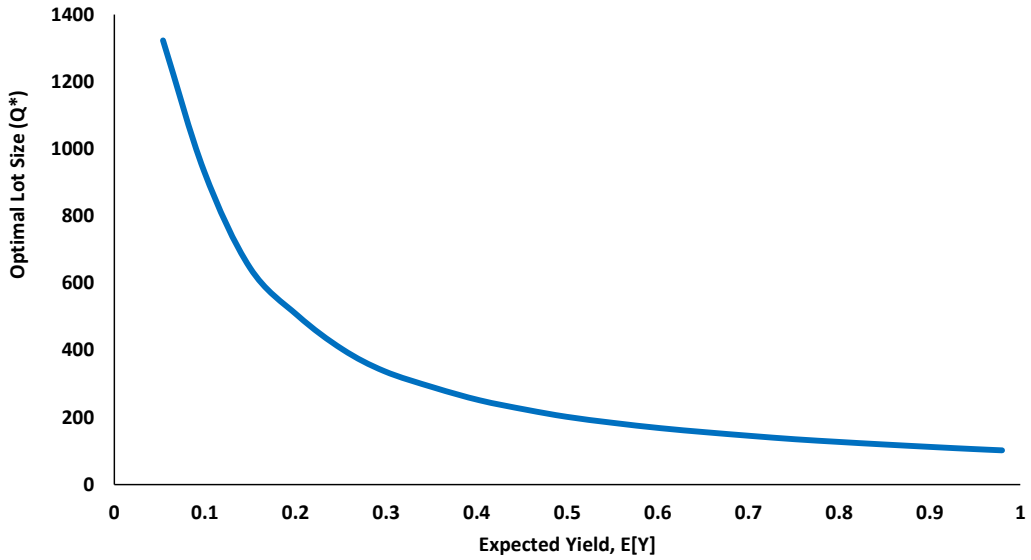

Figure I.1: Relationship between the lot size and expected yield rate

Appendix J

THE CASE OF UNIFORMLY DISTRIBUTED YIELD

Let's consider the case with a uniformly distributed yield between $\underline{y}$ and $\bar{y}$ where $(0 \leq \underline{y} < \bar{y} \leq 1)$. The probability density function of the yield rate is then given by: $f_Y(y) = 1/(\bar{y} - \underline{y})$ for $(\underline{y} \leq y \leq \bar{y})$ and $f_Y(y) = 0$ otherwise.

We then have:

$$E[Y] = \frac{\underline{y} + \bar{y}}{2}$$

and

$$E[Y\mathcal{I}_{d \leq QY}] = \frac{(-(d^2/(2Q^2)) + \bar{y}^2/2)}{(\bar{y} - \underline{y})}$$

Recall that the optimality condition is given as:

$$\frac{E[Y\mathcal{I}_{d \leq Q^*Y}]}{E[Y]} = \frac{b}{b+h} \equiv \alpha$$

Solving the above, for Q^*, we obtain:

$$Q^* = \frac{d}{\sqrt{\bar{y}^2 - \alpha(\bar{y}^2 - \underline{y}^2)}} \tag{J.1}$$

For a given Q, We can also compute:

$$E[(QY - d)^+] = \frac{d^2/2Q - d\bar{y} + Q\bar{y}^2/2}{\bar{y} - \underline{y}} \tag{J.2}$$

and

$$E[((d - QY)^+] = \frac{d^2/2Q - d\underline{y} + Q\underline{y}^2/2}{\bar{y} - \underline{y}}. \tag{J.3}$$

Plugging in the expression for Q^* from (J.1) in (J.2) and (J.3) we can obtain complicated but explicit expressions for expected overage and underage under Q^*.

Finally, if Y depends on some features $\mathbf{U}$ and if the dependence leads to a uniform distribution for each feature vector, we can compute $Q^*|\mathbf{u}$ and $E[c(Q^*|\mathbf{u})]$

using the above. In addition, if we have the probability mass function of the features $P(\mathbf{U} = \mathbf{u})$, we can compute the expected cost per period as:

$$E[C^*] = \sum_{\mathbf{u}} E[c(Q^*|\mathbf{U})]P(\mathbf{u}).$$