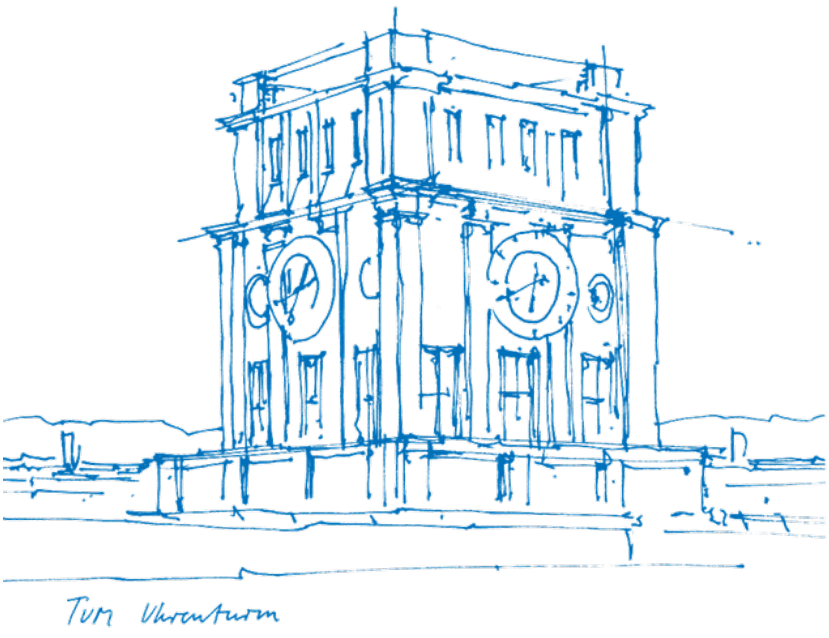
Chair of Information Systems and Business Process Management (i17)
Department of Computer Science
TUM School of Computation, Information and Technology
Technical University of Munich

Bachelor's Thesis in Informatics

Ruan Viljoen

Automated Data-Driven Performance Analysis in Simulation Model Discovery



Chair of Information Systems and Business Process Management (i17)
Department of Computer Science
TUM School of Computation, Information and Technology
Technical University of Munich

Bachelor's Thesis in Informatics

Ruan Viljoen

Automated Data-Driven Performance Analysis in Simulation Model Discovery

Automatisierte datengetriebene Leistungsanalyse bei der Entdeckung von Simulationsmodellen

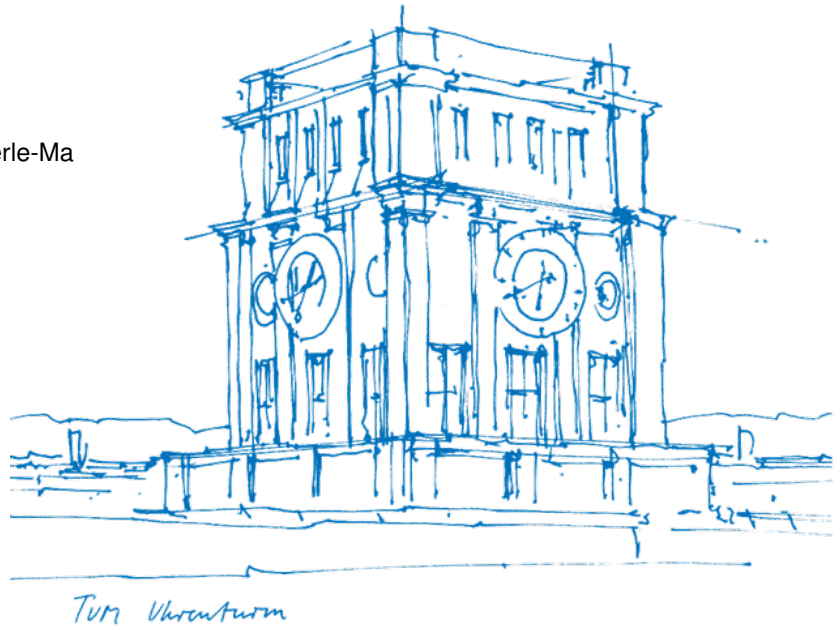
Thesis for the Attainment of the Degree
Bachelor of Science

at the TUM School of Computation, Information and Technology,
Department of Computer Science,
Chair of Information Systems and Business Process Management (i17)

Examiner
Prof. Dr. Stefanie Rinderle-Ma

Supervised by
Michel Kunkler

Submitted on
15.06.2024



Declaration of Academic Integrity

I confirm that this bachelor's thesis is my own work and I have documented all sources and material used.

I am aware that the thesis in digital form can be examined for the use of unauthorized aid and in order to determine whether the thesis as a whole or parts incorporated in it may be deemed as plagiarism. For the comparison of my work with existing sources I agree that it shall be entered in a database where it shall also remain after examination, to enable comparison with future theses submitted. Further rights of reproduction and usage, however, are not granted here.

This thesis was not previously presented to another examination board and has not been published.

Garching, 15.06.2024 Ruan Viljoen

Abstract

This thesis investigates a method to identify overarching patterns in business processes using Gaussian Mixture Models (GMMs) and Decision Trees. The primary objective is to develop a generalized method for automating the analysis and prediction of execution and starting times within business processes, thereby enhancing decision-making and resource optimization.

The research explores the application of GMMs, Decision Trees, and statistics to uncover patterns and relationships in temporal process data. By segmenting data based on influential features and modeling these segments with GMMs, we aim to improve the generality and accuracy of predictive models.

This approach could be particularly useful for resource-constrained organizations and businesses, providing accessible and cost-effective tools for process optimization. If such models can be used as benchmarks, they can help identify patterns that affect all organizations dealing with a particular business process. The findings demonstrate significant improvements in model accuracy and generality, contributing practical solutions and advancing the democratization of business process analytics. This may empower organizations to make data-driven decisions and potentially improve operational efficiency.

Keywords: *Process Mining, Simulation Model Mining, Gaussian Mixture Models, Decision Trees, Business Process Optimization.*

Contents

1 Introduction **8**

1.1 Motivating Example . . . 10

1.2 Research Questions . . . 11

1.3 Contribution . . . 12

1.4 Methodology . . . 12

Artifacts . . . 13

Problem Identification and Motivation . . . 14

Objectives of a Solution . . . 14

Design and Development . . . 14

Demonstration . . . 15

Evaluation . . . 15

Communication . . . 15

1.5 Structure . . . 16

2 Related Work **16**

2.1 Fundamentals . . . 16

Business Process Management . . . 16

Statistical Functions and Fitting . . . 18

Machine Learning . . . 18

The BIC Score . . . 20

2.2 Closely Related Approaches . . . 21

Machine Learning in Business Process Monitoring . . . 21

Business Process Simulation with Differentiated Resources . . . 22

Discovering Simulation Models . . . 23

3 Solution Design **25**

3.1 Part 1: Initial GMM Fitting and Visualization for the Entire Dataset . . . 27

3.2 Part 2: Recursive Partitioning and GMM Fitting on Subsets . . . 27

3.3 Part 3: Filtering Single-Component-Optimal Subsets . . . 29

3.4 Part 4: Comparative Analysis and Visualization . . . 29

3.5 Part 5: Hyperparameter tuning and Complexity Analysis 29

4 Implementation **30**

4.1 Part 1: Data Extraction, GMM Fitting, and Visualization 30

Data Loading and Preprocessing 31

Gaussian Mixture Model Fitting 31

4.2 Part 2: Detailed Exploration through Recursive Partitioning 32

Implementation Steps 32

Implementation and Results 34

4.3 Part 3: Single-Component Optimal Fit Analysis 34

4.4 Execution Times Scenario 37

Entire Dataset 37

Partitioning Subset 38

Single Component Subset 38

5 Evaluation **40**

5.1 Part 4: Comparative Analysis of Model Performance 40

5.2 Part 5: Hyperparameter Tuning and Complexity Analysis 43

5.3 Complexity Analysis 45

6 Discussion **47**

6.1 Contributions 47

Evaluation and Validation of Predictive Models 47

Implications 47

6.2 Practical Applications 48

6.3 Limitations 48

6.4 Future Research Directions 49

7 Conclusion **50**

Bibliography **52**

A Appendix **54**

List of Tables

1 BIC Values for Different Implementations with Depth = 4 and Threshold = 1000 . . 54
2 BIC vs. Recursion Depth at Threshold = 100 54
3 BIC vs. Threshold at Depth = 4 54
4 Run Time vs. Max Depth 55
5 Run Time vs. Threshold 55

List of Figures

1 Probability Distribution of Event Starting times over the course of a day 11
2 The Overall Workflow 26
3 The workflow of Part 1 30
4 Comparison of histogram data with the Gaussian Mixture Model fitting and overall density estimation. 31
5 Workflow of Part 2 33
6 Histogram of a subset of the data after partitioning with the Gaussian distribution fit from the Gaussian Mixture Model. 34
7 Workflow of Part 3 35
8 Histogram of a subset of the data optimally modeled using one component with the Gaussian distribution fit from the Gaussian Mixture Model. 36
9 Comparison of histogram data with the Gaussian Mixture Model fitting and overall density estimation. 37
10 Histogram of a subset of the data after partitioning with the Gaussian distribution fit from the Gaussian Mixture Model. 38
11 Histogram of a subset of the data optimally modeled using one component with the Gaussian distribution fit from the Gaussian Mixture Model. 39
12 Optimal GMM Components at Threshold = 100, Depth = 4 40
13 Comparison of BIC Values for Different Implementations using the time of day for event start times as the target variable 41
14 Comparison of BIC Values for Different Implementations using the process durations as the target variable 42
15 BIC vs. Threshold at Depth = 4 43

16 BIC vs. Recursion Depth 44
17 Run Time vs. Threshold of the recursive segmentation 45
18 Run Time vs. Max Depth 46

Introduction

Business process optimization refers to making business processes as efficient as possible. A business process can be defined as a number of activities performed in a coordinated manner in an organization or technical environment to achieve a business goal [1]. These sets of activities that constitute a business process can be optimized in a number of ways, depending on the target feature whose value is optimized for - this might include minimizing the execution time of a process for example, in which case the target feature is the execution time. The most common goals of this process are long-term effectiveness, cost-efficiency, and market competitiveness for the organizations involved [2].

Within the domain of understanding business processes, Process Mining has become an important topic which encompasses a spectrum of techniques dedicated to extracting insights from event logs to improve business processes [3]. Process Mining can be divided into 4 subfields: process discovery, conformance checking, process reengineering and operational support [4]. Process discovery aims to define process models using event logs. Conformance checking includes a set of techniques to confirm that the event logs and generated process models are in fact equivalent. Process reengineering takes these process models and analyses them, which then informs how the underlying process might be changed or redesigned. Finally, operational support acts as a proactive mechanism to help organizations to not only understand historical process patterns but also to intervene and optimize processes in real-time, ensuring that operational activities align with strategic goals and adapt to changing conditions.

Simulation model mining is the next step. Simulation model mining is the process of using process mining techniques to extract non-trivial and useful information from process execution logs, with the goal of creating comprehensive simulation models. These simulation models represent the various perspectives of a process, such as control-flow, data, performance, and resource perspectives. Simulation models are often represented as colored petri nets (CPNs) which are a handy way to visualize processes [5]. Simulation model mining is important because it allows for the extraction of useful information from process execution logs, provide a controlled environment to analyze and evaluate the performance of different designs or scenarios without the need for costly and time-consuming real-world experiments, and helps in gaining insights into the characteristics of a

process, such as control-flow, data, performance, and resource perspectives. Simulation models allow for the testing of different strategies, which aids in decision-making, and simulation model mining is particularly useful for solving decision problems, such as identifying resource bottlenecks, by providing a way to explore different scenario simulations and make informed choices based on the outcomes of these simulations [5].

Using simulation model mining, model execution times for various resources can be modeled, which is of particular importance since execution time is one of the largest cost contributors and opportunities for optimization on a broad spectrum of organizations. These simulation models provide a representation that can then be analyzed for patterns in execution times or process starting times, for example. In doing so there exists the possibility of finding possible correlations between execution time and other factors that may help organizations in forecasting, identifying bottlenecks and take remedial steps that would either cut costs or optimize output. The research in this thesis is concerned with finding appropriate functions that may be used to model and forecast the relationship between execution times and other parameters to give organizations the tools to aid in making important decisions.

Various research fields have been concerned with fitting these models, but this thesis aims to find a way of automatically finding generalizable relationships between execution times and certain parameters that may differ case-by-case. In doing so, organizations can use this method to find relationships in their own data, and gaining valuable insights that would greatly improve decision-making and resource optimization. Concept drift, where data streams change over time, is a concern for predictive analysis [6], and the approach to be outlined could possibly help detect concept drift.

One proposed method for analyzing the relationships with process execution times have been to use Queuing Theory Models [7]. In this thesis, I will explore the use of classical statistical models as well as machine learning algorithms to try to find models for analyzing execution times in a predictive way.

Machine learning algorithms have been used for business process optimization in manufacturing [8] and therefor will be considered in this thesis. The models derived from the chosen algorithms' output will be assessed and compared. Accuracy will be measured using various metrics to be able to evaluate the final results in varying situations.

Motivating Example

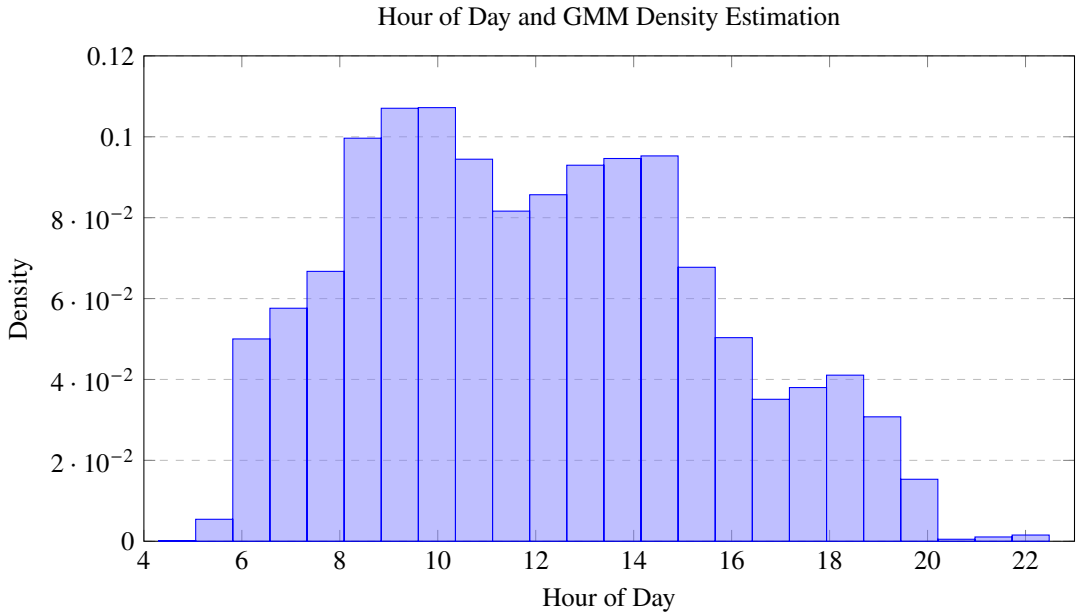
Almost all organizations make decisions at some point with the aim of improving the efficiency of their operations. Larger organizations often benefit from a wealth of resources and talent in data analytics and business process modeling, aiding them in these efforts. However, many smaller enterprises lack these resources, hindering decision-making around process efficiency. Consequently, these smaller enterprises would greatly benefit from having standardized methods of analyzing process efficiency within their operations. The motivation behind this thesis is to address this need by democratizing access to data analytics and business process management.

Let's consider a concrete example to illustrate this point. Take, for instance, the BPI 2017 Challenge [9], which focused on the loan application process of a Dutch financial institute. Figure 1 presents the probability density of process execution times. This visualization not only offers insights into the operational efficiency of the financial institution but also highlights potential bottlenecks and peak performance periods. By understanding the distribution of these execution times, decision-makers can predict future productivity more accurately and make informed decisions based on data-driven insights.

This approach allows for a more nuanced analysis of process data by potentially differentiating between resources types or other classifications to break the data down into valuable insights, automatically. By modeling the distribution of execution times, as in the figure, and then applying some automated method of decomposing it, we can uncover useful insights, empowering organizations to analyze business processes in a variety of scenarios without in-depth knowledge of data analytics or the processes themselves, thereby offering a cost-effective and accessible solution for process optimization.

Ultimately, this thesis aims to bridge the gap between the demand for business process models in resource-constrained environments and the barriers that prevent their large-scale adoption. By exploring a generalized method of generating models, this research seeks to provide a feasible and impactful solution, transcending the barriers posed by the costs and expertise typically associated with data analytics.

Figure 1
Probability Distribution of Event Starting times over the course of a day



Research Questions

- RQ 1: Which statistical, or machine learning algorithms are most suitable for modeling and analyzing execution times and instance starting times of business processes?

  Description: This question will guide the research in evaluating a select set of algorithms for predictive analysis of the execution times and relationships between resources, with a focus on identifying a practical algorithm that provides fundamental analytics that enterprises can follow as a "rule of thumb" for operational efficiency.

- RQ 2: How can decision trees be used to mitigate inaccuracies in simulation models?

  Description: Statistical and Machine Learning models have varying strengths when it comes to modeling data. Statistical methods in particular are favorable for their simplicity, but this might make them less accurate when modeling a relationship within a specific dataset. Machine Learning algorithms, on the other hand, might perform better in certain cases or domains, only to perform less optimally in others. Decision trees hold promise in helping to mitigate both of these effects, helping identify when each method is appropriate to use. This research question will consider how decision trees can help achieve this.

- RQ 3: Which factors can affect the accuracy and generalizability of simulation models?

Description: Among the factors that could be investigated are concept drift, bias, variance, errors in the data and the sample size of the data, along with a host of others. Naturally these factors might play a greater or lesser role in different datasets, and this research question will focus on identifying the particular factors in a few datasets that have a negative effect on the accuracy and generalizability of the simulation models considered in the research.

Contribution

This research aims to make a contribution to the field of business process optimization, particularly in the application of simulation model mining and the use of statistical and machine learning algorithms for predictive analysis.

The primary contribution of this thesis is the development of a generalized method that can automatically discover relationships between execution times and various parameters in business processes. This method stands to benefit organizations by providing a tool for analyzing business processes in diverse scenarios without in-depth knowledge of data analytics or the processes themselves, thereby democratizing access to process optimization tools.

The thesis aims to advance the field of business process optimization by offering novel insights and practical tools that enhance the ability of organizations to analyze and improve their operational efficiency. The research findings are expected to have both theoretical significance and practical implications, particularly for organizations seeking cost-effective and accessible methods for process optimization.

Methodology

The thesis follows the Design Science Research Process (DSRP) proposed by Peffers [10], emphasizing the systematic creation of practical solutions for real-world problems. Unlike traditional research in natural sciences that primarily seeks to understand phenomena, DSR focuses on developing and evaluating tangible artifacts, such as models or methods, to address specific challenges, involving key steps for methodical artifact development and validation [10].

The DSRP model, outlined by Peffers and colleagues, guides research in artifact design and evaluation, comprising six essential steps. It begins with problem identification and motivation, rec-

ognizing gaps in existing knowledge, followed by outlining objectives for a solution, providing a foundation for subsequent design and development. This phase shapes the proposed artifact by drawing on prior literature and theoretical frameworks [10].

A crucial stage in the DSRP model is the demonstration, where the developed artifact is practically applied in real-world scenarios or case studies to showcase its effectiveness and relevance. Subsequently, an evaluation assesses the alignment of the artifact with predefined objectives. The final step involves communicating research findings and artifacts to both academic and practitioner communities [10].

DSR perfectly suits the goals of this thesis, which is all about designing a method. This research is focused on creating practical and widely applicable models, and DSR's systematic approach to designing these models provides a solid foundation. Unlike typical research, DSR emphasizes making tangible tools that address specific challenges, making it a practical fit for our project [10].

DSR's cyclic and iterative process is well-suited for the dynamic nature of optimizing business processes. It allows us to continually improve our method based on real-world applications and evaluations, which is crucial for adapting to the changing needs of small enterprises. The key steps in the DSR process—problem identification, solution development, practical application, evaluation, and communication—align seamlessly with the thorough approach required for this research. Essentially, DSR gives us an effective method not only to contribute to theoretical knowledge but also to provide practical, usable solutions for improving business processes in resource-constrained environments.

Artifacts

In Peffers' DSRP, an artifact is a tangible or intangible creation designed to solve a specific problem. Artifacts can take various forms, including models, methods, constructs, or instantiations. They are developed through a systematic process of design and evaluation, aimed at addressing identified issues or gaps in current knowledge or practice. The purpose of an artifact is to provide practical, innovative solutions that can be applied in real-world scenarios, enhancing understanding and improving existing systems or processes.

Models to Fit Process Data: This artifact involves creating models derived from classical statistical methods and machine learning algorithms. These models are designed to analyze and forecast execution times within business processes. Method of Decomposing These Models: The second artifact comprises a method for decomposing a model that fits process data into clusters or distribution combinations, revealing the underlying structure of the model. Code for Producing Models: The third artifact includes the necessary code used to generate the first two artifacts, facilitating the implementation of the models and decomposition method.

Problem Identification and Motivation

The problem identification arises from a clear gap in addressing the demand for customized business process models, especially in resource-constrained environments like those faced by small enterprises. Motivation stems from the recognition that these entities lack tailored solutions due to barriers such as limited expertise, time constraints, and financial limitations.

In this phase, I will thoroughly examine existing challenges faced by resource-constrained organizations, emphasizing the barriers hindering the creation of customized business process models. The goal is to provide a comprehensive understanding of the motivation driving this research.

Objectives of a Solution

The primary objective is to explore and develop industry-standard and generalized models accessible to any organization, including resource-constrained ones. The aim is to bridge the existing gap by providing effective tools for business process optimization without the need for extensive customization, enabling greater process efficiency information at a low cost.

During this phase, I will outline specific objectives for creating standardized models, considering the unique needs of resource-constrained organizations. I will establish clear criteria for success, ensuring that the developed models are widely applicable and accessible.

Design and Development

This phase involves utilizing advanced techniques such as process mining, simulation model mining, statistics, and machine learning algorithms. These methodologies are employed to extract meaningful insights from existing data, leading to the creation of comprehensive simulation models.

In this phase, I will apply a combination of process mining, simulation model mining, and statistical and machine learning approaches to develop models that effectively capture and represent business process data. In doing so, I will use the PM4PY package developed by the Fraunhofer Institute for Applied Information Technology [11], an open-source python library that supports the cutting edge of process mining algorithms.

Demonstration

The developed simulation models are applied and demonstrated in real-world scenarios, emphasizing their utility in analyzing execution times and understanding the intricate relationships between different resources within a business process. The focus is on showcasing practical applications and benefits in operational contexts.

During this phase, I will execute real-world demonstrations of the developed models, providing concrete examples of their application in diverse business settings by using real datasets. The goal is to illustrate the practical impact and effectiveness of the models in optimizing business processes.

Evaluation

The evaluation stage is crucial for assessing the accuracy and effectiveness of the predictive models. Factors like bias and variance are considered to ensure that the models perform reliably across varying conditions and scenarios. Various accuracy metrics are used to compare different algorithms.

In this phase, I will rigorously evaluate the developed models, considering factors such as bias and variance. the accuracy of the models on each case considered using the Mean Squared Error, or other appropriate standard metrics.

Communication

The final step involves sharing the research findings and the developed models with a dual audience – the academic community, including students, researchers and other players. Communication aims to contribute valuable insights to the academic discourse while also providing practical, applicable solutions for enterprises seeking efficient business process models. This two-fold communication strategy ensures the research has a broad impact, influencing both theoretical advancements and real-world applications.

During this phase, I will present my thesis and research findings. This involves presenting the key concepts in the thesis before discussing the research findings themselves, to lay the groundwork for understanding what the thesis is about.

Structure

- **Related Work:** Reviews existing literature and research relevant to business process optimization, process mining, and the application of statistical and machine learning algorithms in this field.
- **Solution Design:** Details the proposed methodology and approach for automating simulation model analysis, including algorithmic strategies and data handling techniques.
- **Implementation:** Describes the practical aspects of implementing the proposed solution, covering software choices, algorithm development, and setup of the experimental environment.
- **Evaluation:** Presents the results of testing and assessing the proposed solution, including data analysis, interpretation of results, and comparison of different methodologies.
- **Discussion:** Offers a critical analysis of the findings in relation to the initial research questions and the broader context of existing literature, discussing implications, limitations, and potential areas for future research.
- **Conclusion:** Summarizes the key findings, reflects on the research contributions to the field, and provides concluding remarks on the significance and impact of the study.
- **Appendix:** Contains supplementary materials that support the thesis, such as detailed data tables, code listings, and additional experimental results.

Related Work

Fundamentals

Business Process Management

Business process management (BPM) is about designing, optimizing, and automating business processes across an organization to improve the overall efficiency, including reducing costs, and increase customer satisfaction[12].

In the rapidly evolving landscape of the private and public sector, organizations are consistently striving to refine their operations, and achieve sustainable growth. Business Process Management

(BPM) can play a role in realizing these goals, no matter the domain of the activities of a particular organisation. BPM is a strategic discipline that systematically orchestrates the design, execution, monitoring, and continual improvement of business processes [12]. It often provides a framework for organizing end-to-end activities that generate value for an organization, encompassing various concepts and techniques such as process modeling, analysis, and automation, and in doing so fosters innovation and overall improvements [12].

A few concepts and definitions are key to Business Process Management. Firstly, processes are defined as interrelated activities that transform inputs into outputs and in doing so create value [12]. Secondly, the overall contribution to creating value or value stream encapsulates all of the activities involved in transforming a product or service from raw material to the final delivery of the end product or service. Lastly, the BPM lifecycle outlines a structured framework guiding the journey through process design, implementation, monitoring, and continuous improvement [12].

There are a number of key techniques that help drive BPM forward and have become foundational. Business Process Modeling (also BPM) involves a graphical representation of a business process, providing a visual understanding of its structure, flow, and information flows in a format that is similar to a colored petri net (CPN). Next, Process Analysis delves into the current state of a business process, identifying strengths, weaknesses, opportunities, and threats, and paves the way forward for improvement. Lastly, Process Improvement is a structured approach to enhancing performance through data-driven insights and collaboration with key partners and stakeholders, while Process Automation involves the use of technology to automate tasks within a process, leading to improvements [12].

There are a number of advantages that go with adopting BPM practices and techniques within an organisation. This adoption can lead to increased efficiency and effectiveness, resulting in reduced costs, shorter cycle times, and improved customer satisfaction. BPM enhances organizational agility, enabling adaptation to changing market conditions, and finally, by offering improved visibility and control over processes, BPM also minimizes the risk of errors, fraud, and other operational disruptions, safeguarding organizational integrity [12].

Statistical Functions and Fitting

Statistical functions and fitting play a pivotal role in data analysis, enabling researchers to extract meaningful information and insights from complex datasets [13], [14]. These techniques allow us to identify patterns in data and make informed decisions based on the evidence [13], [15]. Statistical functions serve as mathematical tools that represent and describe data patterns [13], [16]. They encapsulate the underlying structure and relationships within a dataset, transforming raw data into interpretable forms [13]. These functions can be broadly categorized into two main types: descriptive and inferential [13].

Descriptive statistics summarize the overall characteristics of a dataset, providing concise metrics of central tendency, dispersion, and shape [13]. Examples of this include measures like mean, median, mode, standard deviation, range, and quartiles. These metrics provide a snapshot of the data's distribution and identify key trends [15]. Inferential statistics, on the other hand, delve deeper into the dataset, drawing inferences about the population from which the data was sampled [15].

Normal distribution fitting is one of the most common statistical fitting techniques [13]. It involves fitting a normal distribution to a set of data points. This helps us to understand the central tendencies, dispersions, and shapes of the data [16]. Other common fitting methods include polynomial fitting, exponential fitting, logarithmic fitting, and power law fitting [13]. Each method is tailored to specific data types and relationships [16].

Statistical fitting plays a crucial role in regression analysis and machine learning. Regression analysis is a statistical method that is used to model the relationship between a number of independent variables and a dependent variable or target variable[13]. Statistical fitting is used to fit a mathematical function to the data, and the resulting model can be used to make predictions about the dependent variable given the value of the independent variables, regardless of whether these values were part of the original 'training' data considered. This ties statistical fitting to Machine Learning, which we explore below.

Machine Learning

In the realm of big data we live in today, where data is abundant, Machine Learning (ML) stands out as a pivotal tool to enable the extraction of information [17]. At its core, ML refers to a set of techniques designed to enable computers to identify patterns in data automatically. These discernible

patterns are then leveraged to predict future data trends and make informed decisions in uncertain situations.

Probability theory forms the bedrock of ML. Whether predicting future outcomes, determining the optimal model to explain data, or deciding the next measurement to perform, probability theory is a versatile foundation in addressing uncertainty that is then built upon to build ML frameworks. While closely related to statistics, ML distinguishes itself by placing a strong emphasis on probabilistic modeling and inference, offering a unified perspective on the entire field and relying on computational power to execute much more calculations that traditional statistics typically deals with [17].

Types of Machine Learning. Supervised Learning is the first and most straightforward type of Machine Learning. Machine Learning is broadly categorised into two main types. In supervised learning, the goal is to learn a mapping from inputs to outputs, based on labeled training sets of values. Classification, a subtype of supervised or predictive learning, involves mapping inputs to categorical outputs. This process finds applications in pattern recognition and regression tasks. A notable example discussed is linear regression, a method for predicting numerical values based on input data [17].

Unsupervised learning deals with uncovering patterns in data without the aid of labeled outputs[17]. Tasks associated with unsupervised learning include clustering, where data points are grouped based on similarities without any particular 'target variable' being defined. These techniques help reveal intrinsic relationships within the data without explicit guidance[17].

Reinforcement Learning is inspired by behavioural psychology. In reinforcement learning, a system interacts with an environment or dataset that represents that environment, learning through trial and error. Feedback in the form of rewards or penalties guides the system in refining its actions over time, much like animals do in reality. Reinforcement learning has applications in various domains, including gaming and robotics[17].

Bias-Variance Trade-Off. A central challenge in ML is navigating the bias-variance trade-off, a delicate balance influencing a model's predictive performance. This trade-off involves managing two sources of error: bias, stemming from overly simplistic assumptions, and variance, arising from excessive model complexity. As model complexity increases, it tends to capture noise in the

training data, resulting in high variance and poor generalisation to new data[17]. Conversely, overly simplistic models may introduce bias by oversimplifying underlying patterns.

The trade-off is similar to walking a tightrope, aiming for a model that generalises well to unseen data while capturing the inherent complexity of underlying relationships[17].

Model Selection and Hyperparameters. In the realm of Machine Learning (ML), practical considerations involve selecting an appropriate model and fine-tuning hyperparameters. This process is akin to choosing the right tool and adjusting its settings for optimal performance[17].

ML models can be compared to a toolbox, each serving a specific purpose. Parametric models offer efficiency with predefined structures but come with assumptions, while non-parametric models are more adaptable but computationally demanding. The decision between the two significantly impacts a model's flexibility and ability to generalize, as elucidated by [17].

Hyperparameters act as adjustable components in a model, similar to tuning the settings on a camera. These adjustments fine-tune the model's performance, allowing it to adapt and make accurate predictions[17].

Overfitting occurs when a model becomes too fixated on the training data[17], akin to delving into excessive details in a story. Oversimplification, conversely, is akin to presenting a generic narrative that lacks nuance. Techniques like cross-validation serve as checks to strike a balance, ensuring the model neither fixates excessively nor oversimplifies, making it a reliable tool for discerning patterns in data[17].

In essence, the efficacy of an ML model lies not just in its complexity but in the meticulous selection of the right model, precise adjustment of hyperparameters, and judicious avoidance of overfitting and oversimplification – a delicate balance for meaningful data interpretation[17].

The BIC Score

The Bayesian Information Criterion (BIC) is a statistical method used to evaluate and select models based on Bayesian probability principles. It operates within a maximum likelihood estimation framework and computes a score that helps in choosing models that are both general and a good fit for the data. The BIC score is derived by considering the log-likelihood of the model, the number

of parameters it incorporates, and the total number of observations. L in the formula represents the log likelihood, n the sample size and d the number of features.

$$\text{BIC} = -2 \cdot \ln(\hat{L}) + k \cdot \ln(n)$$

This criterion specifically penalizes the complexity of the model, thus often favoring models that achieve a good balance between simplicity and the ability to fit the data. As the sample size increases, BIC's preference for simpler models becomes more pronounced, making it a valuable tool for model selection in various statistical and machine learning applications.

Closely Related Approaches

In this section, details of various approaches applied to business processing and simulating business process data more broadly, are discussed. These approaches are grounded in a variety of different disciplines, each of which is titled and listed below. Readers should note that this is not an exhaustive list, but an attempt to paint a picture of what has been tried before in regards to simulation models and analysis.

Machine Learning in Business Process Monitoring

Kratsch et al. (2021) in their seminal paper, "Machine Learning in Business Process Monitoring: A Comparison of Deep Learning and Classical Approaches Used for Outcome Prediction," embark on a comparative study that evaluates the effectiveness of deep learning versus classical machine learning approaches within the realm of business process monitoring [18]. This study is particularly relevant in the context of predictive modeling in business processes, a theme that resonates with the broader focus of this thesis.

The authors methodically contrast the two strands of machine learning methodologies, providing insights into their applicability and performance in predicting business process outcomes. Their findings convey an understanding of the strengths and limitations inherent in different machine learning approaches, including but not limited to various supervised learning algorithms. This comparative analysis serves as an essential reference for researchers seeking to employ machine learning techniques for predictive analytics in business environments, and offers cases where this has worked well [18].

From this research, several relevant insights emerge that are potentially applicable to this thesis. Firstly, the effectiveness of deep learning in handling complex, non-linear relationships in data can be leveraged in developing sophisticated models for business process optimization. Secondly, the comparative advantage of classical machine learning approaches in scenarios with limited data or simpler relationships provides a guideline for model selection based on the specific characteristics of the business process.

Business Process Simulation with Differentiated Resources

The paper "Business Process Simulation with Differentiated Resources: Does it Make a Difference?" by López-Pintado and Dumas investigates an interesting aspect of business process simulation largely ignored before: the use of differentiated resources, challenging the traditional approach of treating resources as uniform entities within a category[19]. This paradigm shift addresses the limitations in accuracy of simulation models that assume homogeneous resource performance and availability.

Differentiated resources in business process simulation imply modeling each resource—human labor, machinery, software systems—individually, acknowledging their unique efficiencies, capacities, availabilities, and other specific attributes. This nuanced approach better mirrors real-world complexities of business processes, enhancing the fidelity of simulation models.

The authors propose an automated method to derive these models from event logs, a significant advancement for creating data-grounded simulations. This methodology involves defining specific weekly calendars and notations, adapting existing methods to discover resource availability calendars from event logs, and revising confidence and support metrics suitable for this context [19]. The process further includes the allocation of activities to resources based on these logs and the use a function to estimate differentiated resource performance. This function adjusts event durations according to each resource's calendar, building histograms from these durations, and applying curve-fitting to find appropriate probability distributions for each resource-activity pair [19].

Empirical evaluation shows that models with differentiated resources more closely replicate actual business process dynamics, such as cycle times and resource utilization patterns, compared to models with undifferentiated resources. This level of detail and realism is crucial for precise decision-making and optimization in business process management [19].

In the context of this thesis, which aims to enhance business process optimization through predictive modeling, the concept of differentiated resources is highly relevant. Incorporating these principles into this thesis' simulation models could lead to more accurate predictions and effective optimization strategies. By acknowledging the unique characteristics of each resource, we can develop models that not only reflect real-world scenarios more accurately but are also more capable of identifying optimization opportunities in business processes.

Discovering Simulation Models

Rozinat et al. (2009) introduce a novel approach to business process simulation, focusing on the automatic creation of simulation models using process mining techniques. This methodology represents a significant shift from traditional manual model creation which is very cumbersome, leveraging process execution logs to build comprehensive models. These models encapsulate multiple dimensions of a business process, including control-flow, data, performance, and organizational perspectives, integrated into a Colored Petri Net (CPN) [5].

Process mining, the cornerstone of their approach, begins with control-flow discovery, which automatically constructs a process model from an event log, capturing the causal relations between activities within a business process. The model is visualized using a Petri net, representing dynamic transitions and states based on recorded activities and cases [5].

The subsequent stage involves decision point analysis, which identifies data dependencies affecting the routing of cases in the process. This analysis discovers rules for different routes within the process based on data attributes associated with log events [5].

Performance analysis further enriches the model by incorporating execution and waiting times for activities, and probabilities for alternative paths. This data is derived from the log, assuming a normal distribution for times and a Poisson arrival process for case generation [5].

Another vital aspect is organizational perspective mining, which discovers the relationships between resources, roles, and activities. By analyzing activity frequencies, resource groups or roles are identified, providing insights into the organizational structure within the process [5].

Rozinat et al.'s methodology facilitates the construction of detailed and objective simulation models, grounded in real data. These models are useful in evaluating different process designs and exploring the effects of potential redesigns. The application of process mining techniques in this context marks a significant advancement in business process simulation and management [5].

Solution Design

The aim of the research conducted in this thesis is to find a method of discovering generalized models within business process data. This means that we are looking for a way to explore data in an automated way, that yields results that allow decision-makers to make generalizations that are valid. The initial assumption is that modeling a dataset using normal distribution components would yield more generally applicable models, and the hypothesis is then that finding subsets in this dataset that can be modelled using fewer normal components than the overall subset will further improve the generality of the resulting model. To be able to determine whether this hypothesis is corrected, the BIC Score is measured for a model of the overall dataset, and each subsequent set of subsets. The average BIC Score of these sets of subsets is then used to be able to compare them to the overall dataset and each other. An overall improvement will yield a better average BIC score. In this section the method referred to above is explained in different parts, each representing a step in the overall process of implementing the approach or evaluating its efficacy. The method employs Gaussian Mixture Models, Decision Trees and filtering to produce the resulting models.

It shares the spirit of something referred to as distributional regression in other research, which is that normal distributions within data is a useful basis on which to make predictions without being overly reliant on the available data, and so the prediction will be generally applicable to future values added to the dataset, but which pertain to the same business case and business processes. This helps achieve the goal of providing decision-makers with models that represent the data, but may be used for extrapolation - something that supervised learning algorithms are not well-suited to, resulting in an overfitted model that cannot generalize. Having a dataset that is normally distributed means that additional samples of data that relate to the same processes will continue to result in a normally distributed dataset - this is the key assumption behind why normally distributed datasets are preferred in this thesis and largely relies on the logic of the law of large numbers.

The process to try to identify these normally distributed subsets within a larger dataset involves modeling the original dataset with a Gaussian Mixture Model (GMM), then taking the original dataset and segmenting it based on some of the other features in a recursive way, always choosing the most important feature for segmentation. This means that for a dataset that has feature A with possible values 1 and 2, two different subsets will be generated - one with all the records that have

feature A value 1 and one with all the records that have feature A value 2. The choice of feature is based on which feature has the greatest influence in predicting the target variable, and will be explained in greater detail later on. Once these subsets are generated, they are modelled with a GMM as well, and the subsets' models are compared to the original dataset GMM, using a specific metric the BIC score, to determine which is best. Once this process has been completed several times resulting in subsets that may be segmented based on various features, all those subsets that can be optimally modelled using only 1 component in its GMM are filtered out, and these subsets are compared to the segmented subsets before filtering on the basis of their BIC scores - again to determine if the resulting models do a better job of predicting values in a generally applicable way. Each step of the approach is detailed below, along with the logic behind the implementation.

Figure 2
The Overall Workflow

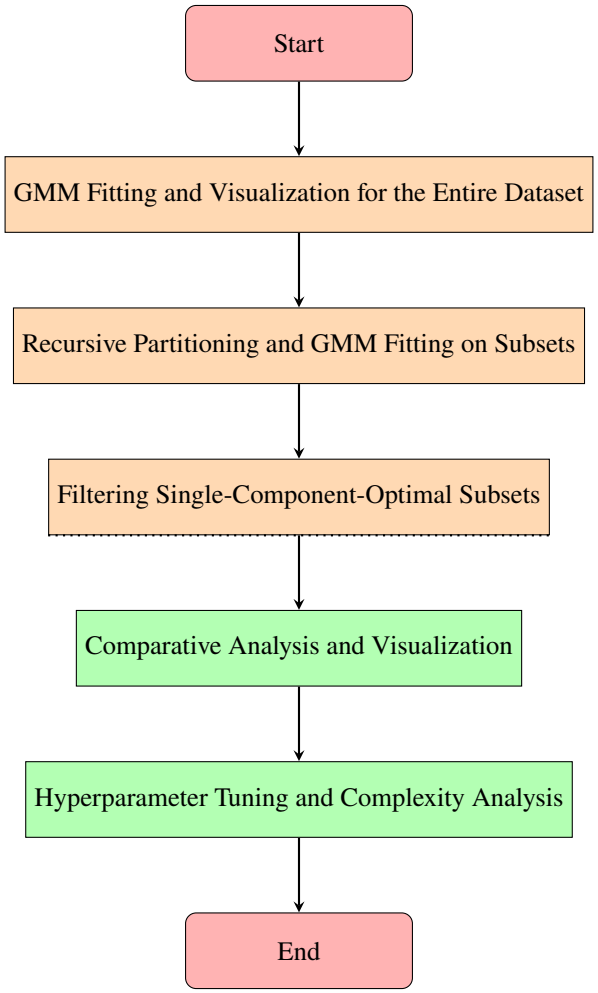


Figure 2 gives an overview of the parts that went into the implementation and evaluation of the method in question.

Part 1: Initial GMM Fitting and Visualization for the Entire Dataset

This part of the analysis begins with the extraction of data from the event log, including all features related to a particular record in the business process data. One of these features is selected as the target variable, i.e. the feature that will be used as the dependent variable in the models and predictions that follow. In our analysis the target variable relates to some temporal feature such as process execution or start times. Following this extraction, a Gaussian Mixture Model (GMM) is fitted to the entire dataset to analyze the distribution of the target variable over time. The Bayesian Information Criterion (BIC) is used as the metric to optimize the model for a given target variable. The target variable value is used in the log likelihood calculation of the BIC score, and then evaluated to select the optimal number of components for the GMM, aiming to balance model complexity with fit quality. The resulting GMM is visualized alongside a histogram of the data, providing an initial graphical representation of the event distribution within the dataset. This visualization helps to set a baseline understanding of the data, which is crucial for subsequent, more detailed analyses. The initial BIC score value will also be used to show that subsequent models generated after the next few steps in the process are better models than the original.

Part 2: Recursive Partitioning and GMM Fitting on Subsets

This segment embarks on dividing the initial dataset into subsets through a methodical approach driven by decision trees, which identify significant features for segmentation. The process of doing so works as follows:

1. **Feature Identification Using Decision Trees**: A decision tree regressor is used to model the original dataset, resulting in a model that optimizes the Mean Squared Error (MSE) between the regressor's predictions and the actual values of the target variable. This is a common metric to use in supervised learning algorithms to measure the distance between the predictions and the actual values.

   Next, handling categorical data is a key step; the function identifies and encodes non-numeric features using LabelEncoder, which replaces categories with a numerical representation essential for decision tree operations. The model also excludes other features, like the timestamps

used in the target variable and record ids, to concentrate the learning process on relevant attributes.

The function then divides the dataset into training and testing subsets, using 80% for training to fit the model and 20% for validation, ensuring that the model's performance is evaluated on unseen data. During the training phase, the DecisionTreeRegressor is employed, which selects features and split points at each node by determining which reduce the mean squared error (MSE) the most, effectively partitioning the data into homogeneously valued groups.

After training, the feature importances are quantified. The most important feature is defined here as the feature with the greatest influence on the mean squared error calculated by the decision tree regression. To determine this involves calculating the extent to which each feature decreases the overall MSE, with these reductions weighted by the number of samples affected at each split. This results in a normalized importance score for each feature, highlighting their respective contributions to prediction accuracy. In this way, this Decision Tree Regressor can be used to determine the most important feature each time recursive partitioning happens. Once the most important feature is determined, subsets are created based on the possible values of the feature in question.

2. **Recursive Partitioning**: Upon identifying the most important feature, the dataset is partitioned based on this feature. Recursive partitioning involves splitting the dataset into subsets based on the values of the identified feature. If the feature is categorical and has a set number of possible values, this process is easy. When the feature is numeric in nature, intervals are automatically identified in the numeric data using GMMs and then the subsets are generated based on these identified clusters. This means that a number of subsets are created from the original dataset using this most important feature to divide them. Each subset will have records with a specific value of this most important feature, or records that have a value within a specific interval in the case of numeric data. This process is continued recursively, until either the resulting subset is smaller than a certain threshold, e.g. 500 records, or the partitioning has been done for a certain number of times, the maximum recursion depth.
3. **Gaussian Mixture Modeling (GMM)**: For each subset obtained through recursive partitioning, a Gaussian Mixture Model is fitted. Each one's BIC score is calculated and the average of all their BIC scores is compared with the original dataset model's BIC score to see if this step yielded improvements in the models. Comparing the BIC score of the original dataset with this average BIC score for the segmented subsets answers the question of whether the

recursive partitioning based on the most important feature identified actually resulted in a more homogeneous and predictable dataset.

Part 3: Filtering Single-Component-Optimal Subsets

In this analysis phase, we take our quest for generally applicable subsets one step further. Subsets that are best modeled with a single GMM component are identified within the set of subsets found after the recursive partitioning process. By filtering these subsets, the investigation emphasizes data segments where a singular distribution component suffices, streamlining the model. This simplification allows for a focused examination of specific data behaviors, contrasting with the more complex models necessary for other segments. The average BIC score across these singularly modeled subsets is calculated, offering a measure of model efficacy and simplicity. Additionally, the visualization of a randomly chosen subset, fitted with its optimal single-component GMM, illustrates the distribution fit, enhancing understanding of the data's inherent structure.

Part 4: Comparative Analysis and Visualization

The final segment of our analysis undertakes a comparative review, presenting the BIC scores of three models: the complete dataset, the detailed segmented subsets, and those subsets apt for single-component GMM that were generated by part 1, 2 and 3, respectively. This comparison assesses each model's fit and complexity, and helps to show the reader that each step yields a more robust and valuable model than the previous step. Furthermore, a pie chart depicting the ratio of subsets that were optimally modelled using one component is shown, to indicate the fraction of subsets that reach this 'ideal'.

Part 5: Hyperparameter tuning and Complexity Analysis

The analysis process involves exploring the impact of varying threshold and maximum depth values on the execution time of a recursive partitioning algorithm. This method aims to understand the computational complexity and identify optimal hyperparameters for the Gaussian Mixture Model (GMM) fitting within the data segmentation context. We define a range of threshold and max depth values and systematically measure the execution time required for recursive partitioning and subsequent GMM fitting across these parameter values. This allows us to visualize the trade-offs between computational expense (execution time) and model granularity (controlled by threshold and depth).

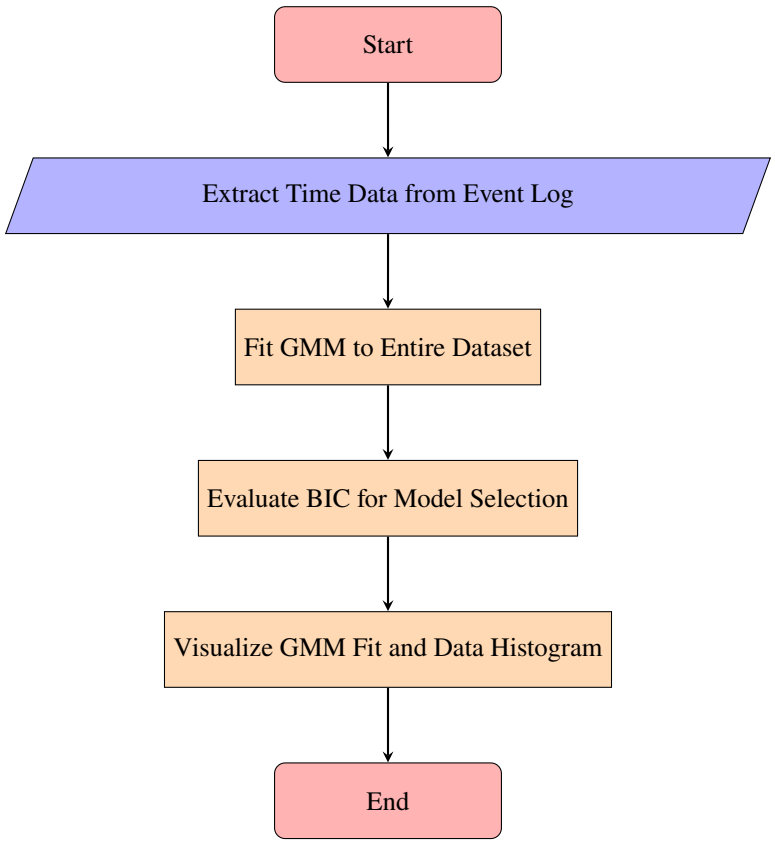
Implementation

In this section, the detailed process outlined in the Solution Design is actually implemented in code and the results shown. This section is again organised into the various parts mentioned in the Solution Design, with descriptions of the implementation specific detail where relevant and a flow chart outlining the steps already explained in the previous section. The implementation is first explained using the process start times as the target variable, and thereafter, at the end of this chapter, the results are shown for the scenario where process durations are the target variable.

Part 1: Data Extraction, GMM Fitting, and Visualization

Part 1 is concerned with extracting the data so that a first attempt model can be generated, with the "hour of day" for event occurrences as a target variable in this section. The process below outlines the implementation details of Part 1.

Figure 3
The workflow of Part 1



Data Loading and Preprocessing

The approach begins with loading the BPI 2017 event log into a Pandas DataFrame, setting the stage for in-depth analysis. We then engage in feature engineering to extract meaningful insights:

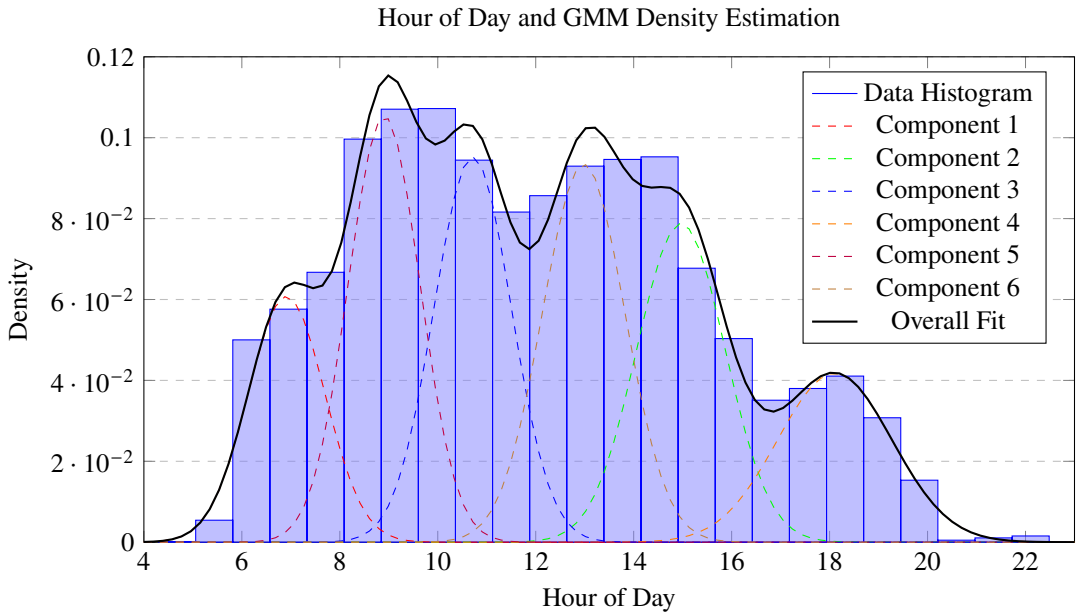
- The "hour of day" analysis entails extracting the time of day from event timestamps, aiming to discern patterns in event frequencies over the course of a day.

Gaussian Mixture Model Fitting

To model the distribution of these features, we use Gaussian Mixture Models (GMMs), optimizing the model based on the Bayesian Information Criterion (BIC), to ensure the generality of the result and avoiding overfitting the data. This technique is closely related to the concept of distributional regression, and is briefly shown in action below.

```
# For "hour of day":
X_hour_of_day = data['hour_of_day'].values.reshape(-1, 1)
gmm_hour = GaussianMixture(n_components=n,
    covariance_type='full').fit(X_hour_of_day)
```

Figure 4
Comparison of histogram data with the Gaussian Mixture Model fitting and overall density estimation.



The density estimation plots of the GMMs provide a visual representation of the data distribution and allows us to conduct an analysis of temporal patterns within the BPI 2017 dataset. Here we can see that there are clusters in the data that are homogeneous or distributed in a predictable way, but as it stands the data is too broad to be able to use these models for targeted decision-making. The goal now is to dissect these datasets to identify subsets that would be better suited to this purpose, and this involves the next step: the recursive partitioning of these datasets.

Part 2: Detailed Exploration through Recursive Partitioning

With preliminary insights gained from GMM analysis in part 1, we can now delve deeper into the dataset via the recursive partitioning process outlined in the Solution Design section. A Decision Tree is used to identify features for segmentation. This is done by finding the features that have the largest impact on predicting the target variable 'hour of day' with the goal of uncovering more nuanced patterns within the event log by separating the data into subsets.

Implementation Steps

The approach is systematic, involving several key steps to dissect the dataset into more homogeneous segments that might offer better insights that the mangled original datasets.

1. **Feature Identification using Decision Trees**: We employ a Decision Tree Regressor to pinpoint the most significant features that influence the time of the day that processes are started. This critical step ensures our segmentation is data-driven. On each iteration of the recursion algorithm running, this function again identifies the 'most important feature' considering the new subsets generated by the previous iteration.
2. **Recursive Partitioning**: Following feature identification, the dataset is divided based on the selected features, with this process repeating recursively within each subset. This method uncovers subsets with shared characteristics, allowing for a more detailed analysis.
3. **Gaussian Mixture Modeling on Segments**: GMMs are fitted to each segment to model the distribution of the targeted features, highlighting unique patterns within specific dataset segments. A limit of 11 components is set for the GMM model in this implementation.

Figure 5
Workflow of Part 2

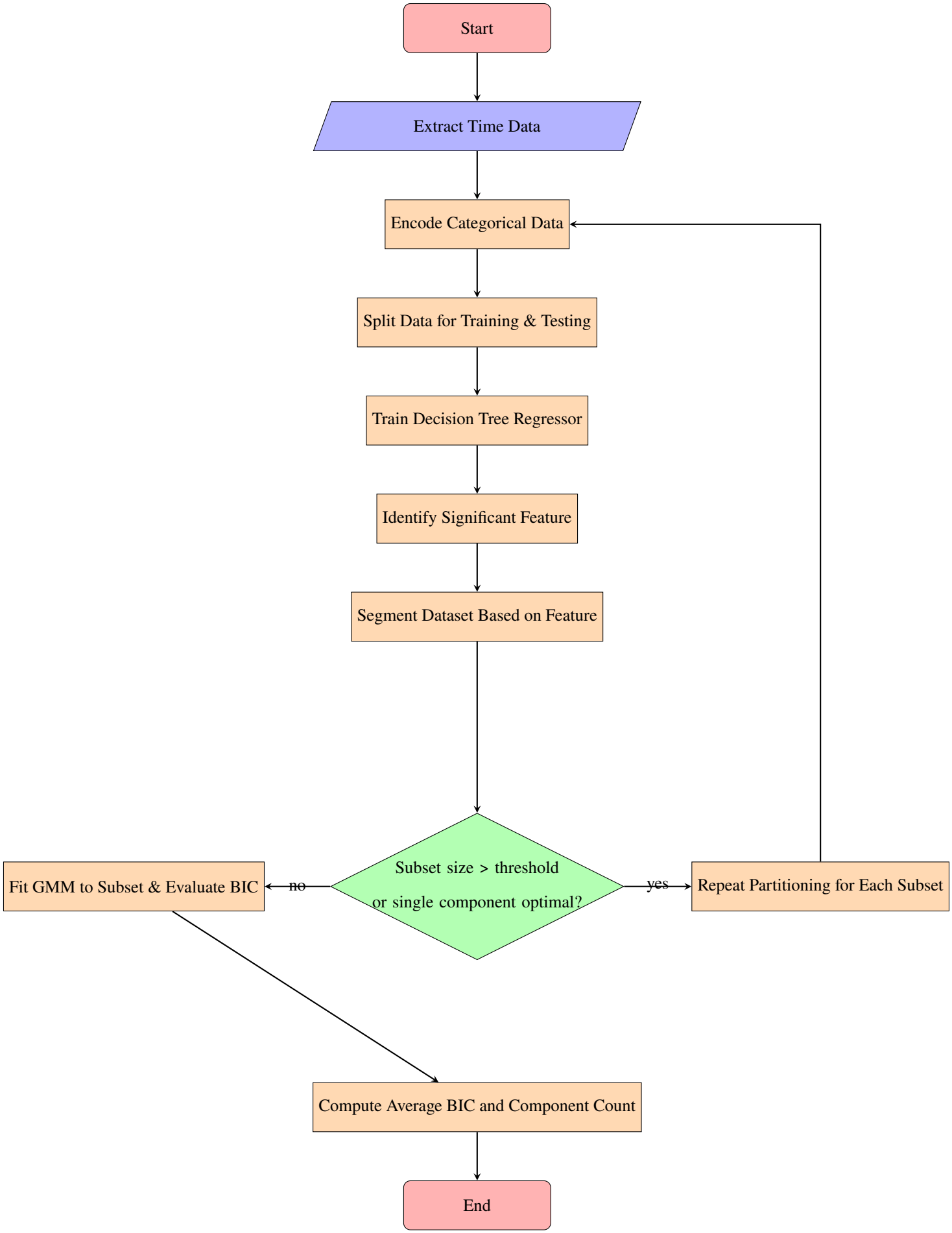
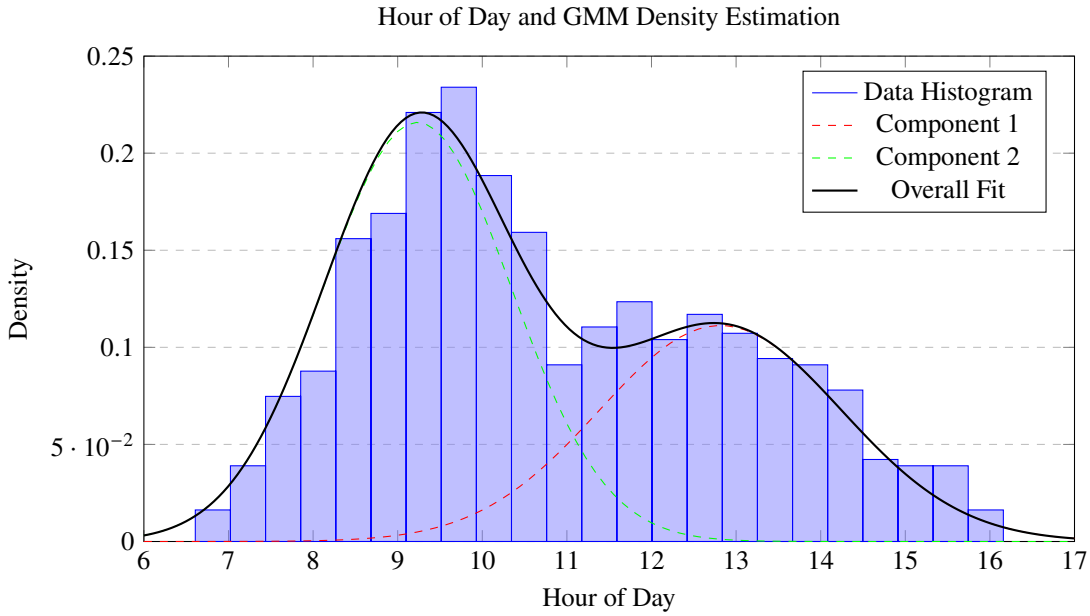


Figure 6
Histogram of a subset of the data after partitioning with the Gaussian distribution fit from the Gaussian Mixture Model.



Implementation and Results

In Figure 6 are the results of one such a subset uncovered by the recursive segmentation process. Readers are encouraged to note the BIC scores are now significantly improved, and the data is more homogeneous.

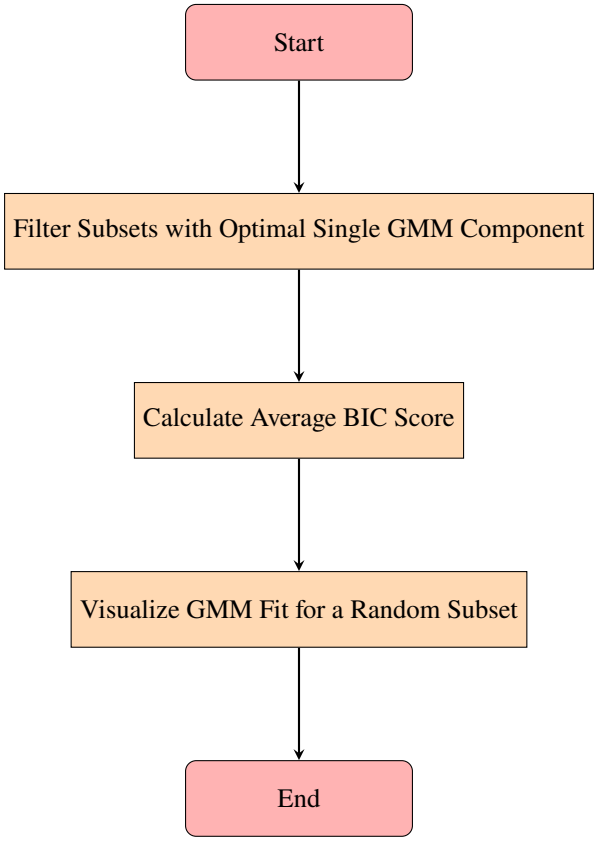
Here we can see that our approach has already yielded an improved model, where fewer clusters in the data exist and the model has a better BIC Score than the model based on the entire dataset shown before. We can start to use the data more effectively for decision-making because it provides a coherent and more normally distributed model. This particular subset indicates that the processes in question in this subset are more likely to be started in the morning than the afternoon. Some subsets identified and modelled in this way will obviously remain overly complex, and likely never lead to a generalizable subset and model, but in many cases - like the one shown above - this segregation of the data is useful.

Part 3: Single-Component Optimal Fit Analysis

Continuing from the segmentation results, we shift our focus to the examination of subsets among the identified set of subsets which are optimally modelled by a single Gaussian Mixture Model (GMM)

component. This analysis targets a subset of data segments identified in the recursive partitioning process, narrowing our lens to those segments for which a singular GMM component provides the best fit.

Figure 7
Workflow of Part 3

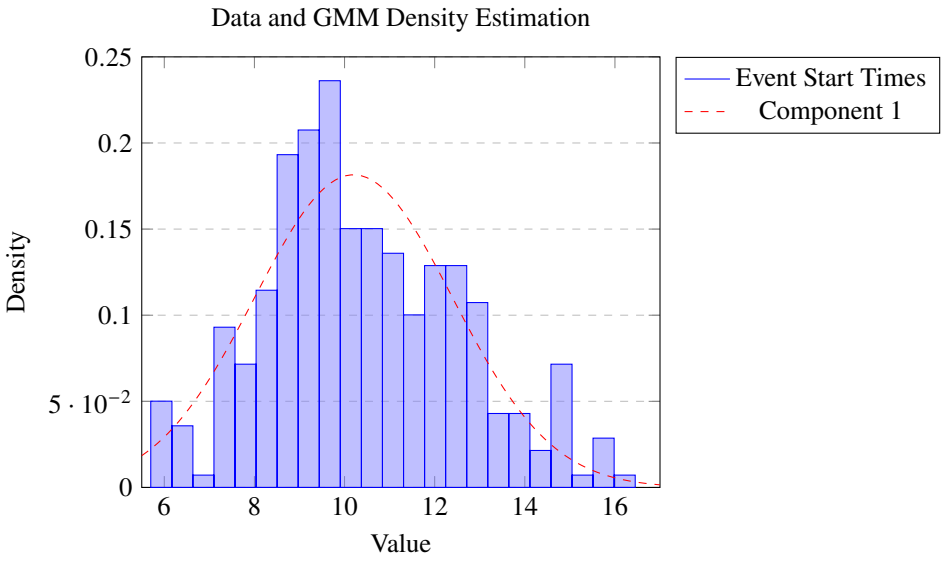


This approach is a simple, but particularly useful for identifying homogeneous segments within the process. The implication here is that these subsets are normally distributed and the most useful for decision-making. From the histogram and model above we can see that the processes in this subset are normally distributed over a typical working day start around 6am. This might help decision-making around when productivity is optimal during the day, and help management make operational decisions with this in mind.

The plot in Figure 8 . A manager looking at this subset can see a clear normal distribution of the data and make decisions accordingly, knowing that this distribution applies to all process instances that fulfil a series of conditions which were used in decomposition to arrive at this point. This

result completes the approach that started with a broad dataset, continued to segment this dataset and finally filtered out the subsets that are optimally modelled using a single component.

Figure 8
Histogram of a subset of the data optimally modeled using one component with the Gaussian distribution fit from the Gaussian Mixture Model.



Steps to achieve the result in Figure 8

1. **Subset Filtering**: First we filter out the subsets among all the subsets from part 2 where a single-component GMM achieves the lowest Bayesian Information Criterion (BIC) score, ensuring focus on segments with minimal internal variance. BIC score is considered a good metric for balancing generalizability and fit when considering models, in particular clustering models.
2. **BIC Score Evaluation**: We meticulously calculate and analyze the BIC score for each filtered subset. A lower BIC signifies a model's capacity to effectively capture the underlying data distribution. A BIC score difference of more than 10 between models indicating a strong improvement [20].

Implementation Detail. A snippet of the implementation for identifying these optimally fit subsets is shown here:

```
def filter_and_calculate_bic_single_component_subsets(final_partitions):
    single_component_subsets = []
```

```
for partition in final_partitions:
    X = partition['data']
    gmm = GaussianMixture(n_components=1).fit(X)
    bic_score = gmm.bic(X)
    if bic_score < predefined_threshold:
        single_component_subsets.append((partition, bic_score))
```

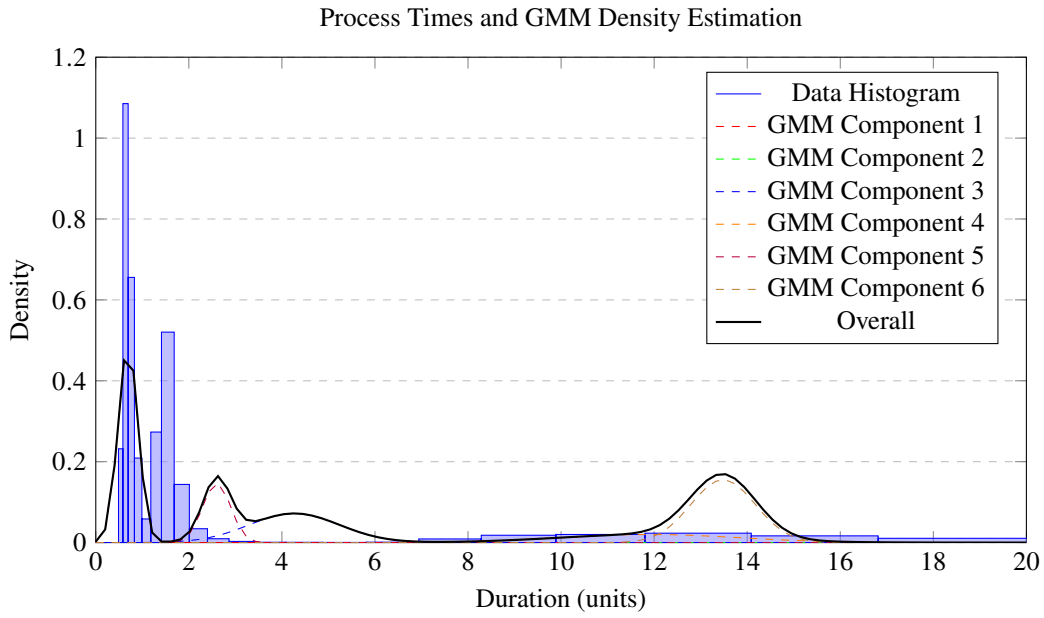
Execution Times Scenario

This section provides the data and graphics that result from applying the same method as outlined above to the BPI 2017 dataset, but changing the target variable to be the process duration times. To extract these process durations, the timestamp of all events in the dataset were taken, and a petri net was extracted to visualize the dataflow using the PM4PY library. Adjacent events in this process flow were used to calculate the process durations.

Entire Dataset

Figure 9

Comparison of histogram data with the Gaussian Mixture Model fitting and overall density estimation.



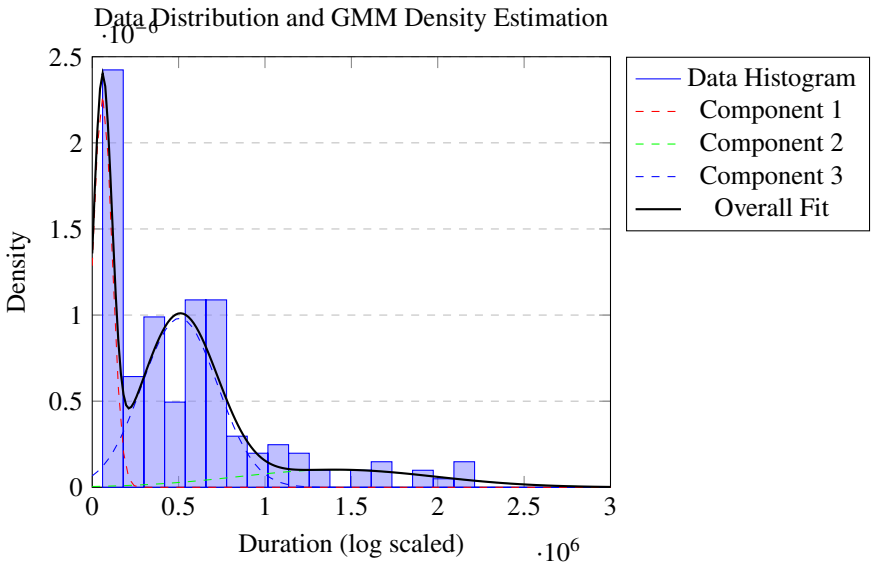
The initial dataset with the process durations as target variables modelled by a GMM yields the results shown in Figure 9. This shows a very messy figure which does not have clearly discernible

underlying components and would be difficult to use in decision-making. In such a case, the usefulness of decomposing the dataset into more homogeneous subsets becomes quite clear. The BIC (Bayesian Information Criterion) score for the model in Figure 10 is calculated as BIC = 584247.77

Partitioning Subset

A subset derived from the recursive partitioning algorithm, and yields the GMM below. Here the data is more coherent, with distinguishable components that underlie the data. Had this data been presented to a manager, they would be able to discern the general trends underlying the data and make decisions accordingly. Further clarification on why certain clusters exist seems not to be derivable by additional partitioning.

Figure 10
Histogram of a subset of the data after partitioning with the Gaussian distribution fit from the Gaussian Mixture Model.

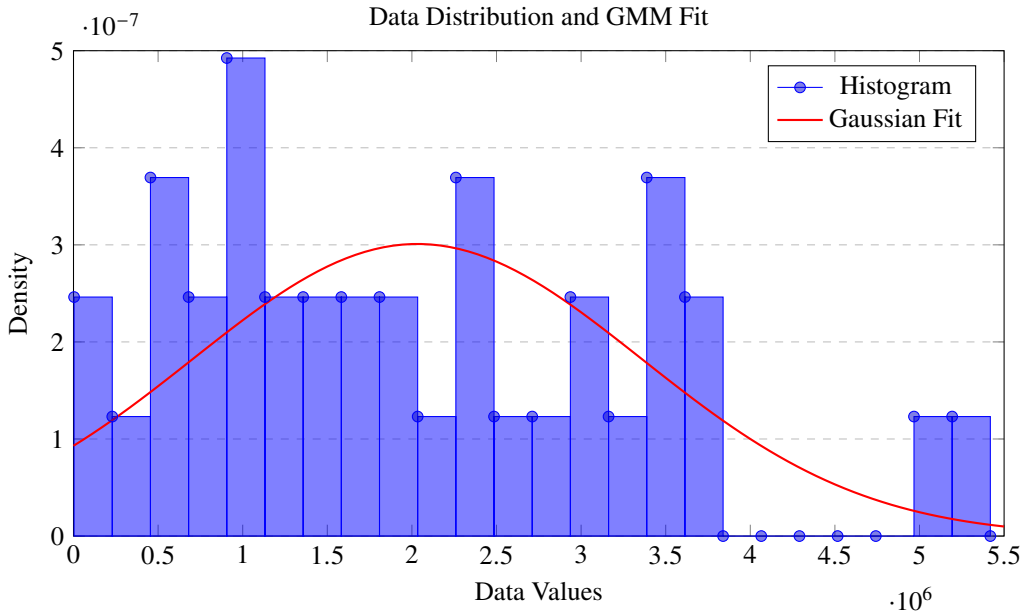


Single Component Subset

The set of subsets is filtered to yield those which are optimally fitted by a single GMM component. One of these is pictured below. Here the data has been decomposed up to a point where there exists only one optimal gaussian component and again this may be useful for decision-making.

Figure 11

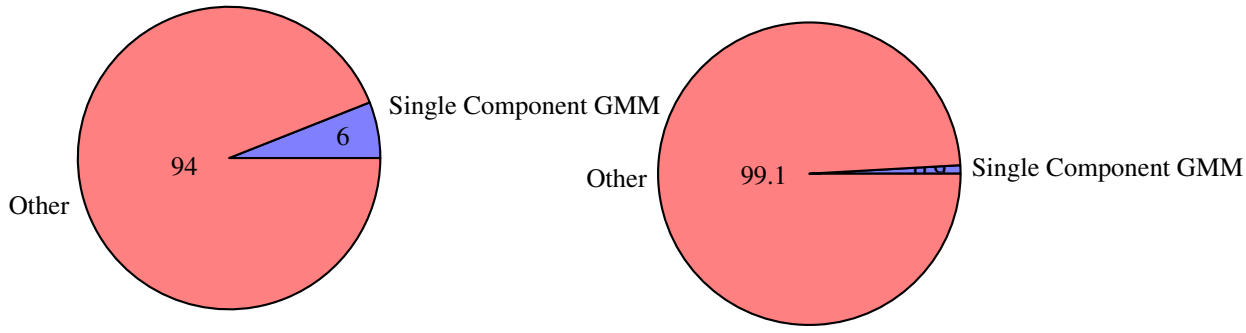
Histogram of a subset of the data optimally modeled using one component with the Gaussian distribution fit from the Gaussian Mixture Model.



Evaluation

In this section we compare the results at each step of the process outlined in the implementation section, making reference to the BIC Score in particular as the metric used to judge any possible improvement. It is worth noting once again that a decrease of more than 10 in the BIC Score corresponds to a very strong improvement in the model[20]. Figure 12 below shows the fraction of all subsets which were identified to be optimally modelled using only one GMM component, with fixed values for the threshold = 100 and depth = 4. What is interesting to note is that, under the same conditions, the target variable as the process durations yields considerably fewer such subsets. This goes a long way in showing that the nature of the dataset and underlying distribution of the data greatly impact the result of the method described under the implementation section.

Figure 12
Optimal GMM Components at Threshold = 100, Depth = 4



(a) *Start Times*

(b) *Durations*

Part 4: Comparative Analysis of Model Performance

Following our deep dive into the intricacies of Gaussian Mixture Models (GMM) for both event start times and process durations within the BPI 2017 dataset, Part 4 consolidates the findings through a comparative analysis of model performance. This section compares the Bayesian Information Criterion (BIC) scores obtained from the overall dataset model, segmented subsets model, and the subsets optimally modeled with a single GMM component.

Analytical Framework

The three different cases being compared in this part are as follows:

1. **Overall Data Model**: This model, applied to the entire dataset, serves as the baseline.
2. **Segmented Subsets Model**: Here, we delve into the granularity introduced by segmenting the dataset based on the most important feature identified by the Decision Tree Regressor, analyzing the average BIC score across all partitions.
3. **Single-Component Subsets Model**: Focusing on subsets which are optimally modelled using only a single normal distribution component when running the GMM on it, this model delves into small pockets of homogeneous data among the many subsets generated through segmentation.

Shown below are the main findings of comparing the three implementations using the two different target variables mentioned, process start times and process durations. Both Figure 14 and Figure 13 are log-scaled for better visualisation. The average BIC Score of all relevant subsets was used for the 'segmented subsets' and 'single-component-optimal subsets' totals. The choice of using the average is to account for the total effect, but other metrics such as the median BIC Score or more complex metrics might be more appropriate in certain circumstances. The 'overall data' value is of course the BIC Score of the model derived from the overall dataset.

Figure 13
Comparison of BIC Values for Different Implementations using the time of day for event start times as the target variable

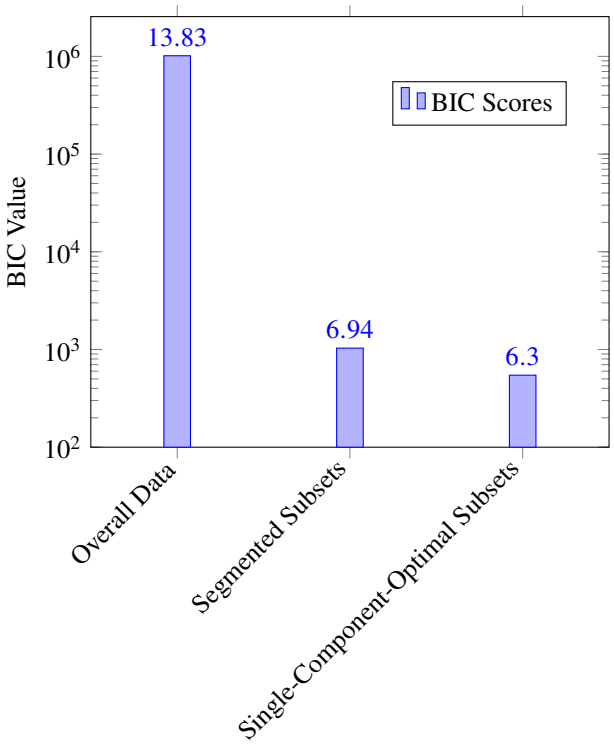
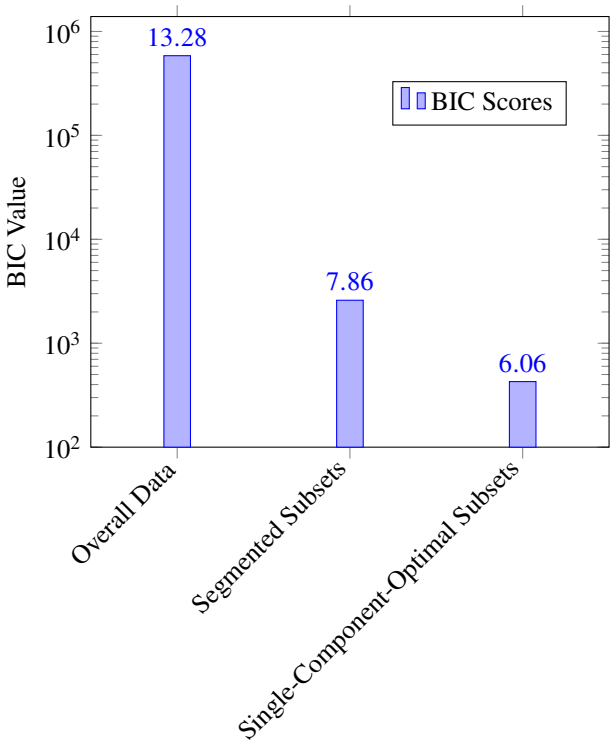


Figure 14
Comparison of BIC Values for Different Implementations using the process durations as the target variable



Key Insights

A clear improvement in the BIC score is noted for the segmented subset models in both Figure 14 and Figure 13, with an even greater improvement for those subsets that can be modelled using only one normal distribution component in both cases as well.

- **Impact of Segmentation**: A clear improvement in the BIC score is noted for the segmented subset models in both Figure 14 and Figure 13. The segmented subsets are more homogeneous and produce better models since the underlying pattern in the data is better discernible.
- **Simplicity vs. Complexity**: The single-component subsets model presents the lowest average BIC scores among the segments it applies to, highlighting scenarios where the data is broken down to a very homogeneous set that conforms to some underlying normal distribution.
- **Subset Modeling Relevance**: The choice between a naive full dataset, segmented dataset, or normal-distribution dataset modeling approach must align with the analytical objectives and the specific characteristics of the datasets under consideration, since trying to isolate very

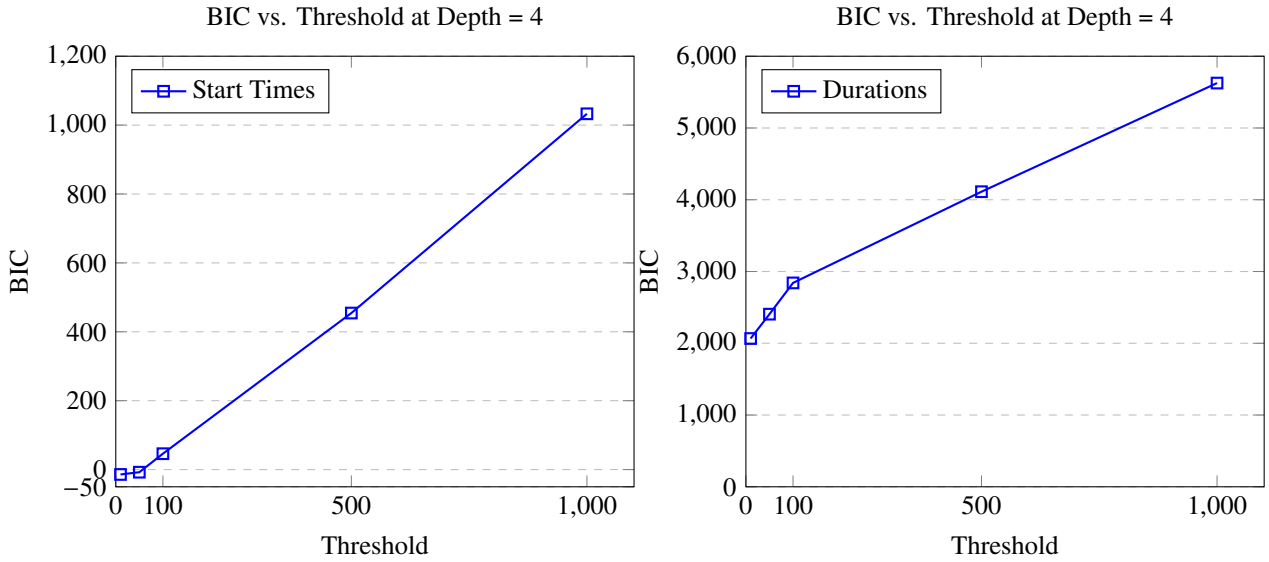
homogeneous subsets in the data, is not always possible or useful. It is only when the subset of data is actually useful to decision-making that it would make sense to do so.

Part 5: Hyperparameter Tuning and Complexity Analysis

The final step is dedicated to optimizing the recursive partitioning process through meticulous hyperparameter tuning and conducting a basic complexity analysis to ascertain how viable the implementation is for larger datasets. The objective here is not just to enhance the model's precision but also to gauge its scalability and efficiency across larger datasets and more intricate process mining tasks, and identify potential optimal values that may be generally applicable across datasets.

The tuning process helps in pinpointing the optimal blend of threshold values and maximum recursion depth, minimizing the Bayesian Information Criterion (BIC) score in executing the recursive partitioning function. The figures below refer to the BIC score as it varies with different values for the threshold and recursion depth. Figure 15 indicates that the BIC score decreases as the recursion depth increases, stagnating after a recursion depth of 4.

Figure 15
BIC vs. Threshold at Depth = 4



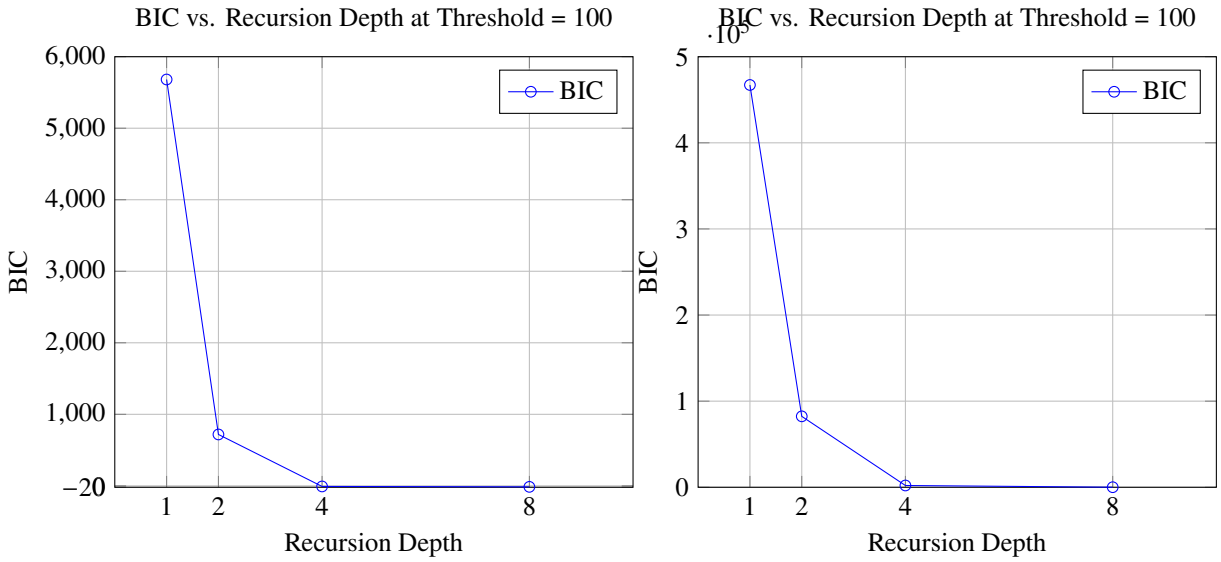
(a) *Start Times* **(b)** *Durations*

In Figure 15a a local minimum exists between 0 and 100 before increasing, while in Figure 15b the BIC score seems to increase as the threshold increases with no clearly visible local minima. Both

show the general trend that the higher the threshold value is, the higher the BIC score will be. The intuition behind why this might be the case could be that the higher the threshold value, the larger the subsets are before the recursive partitioning is stopped. These larger subsets then might still contain relatively more heterogeneous data.

In Figure 16, we look at the relationship between the recursion depth of the recursive partitioning function used to decompose the overall dataset into subsets, and the BIC Score. The trend here is again consistent between the two cases (the cases with start time or process duration as the target variable, respectively). Here we see that the BIC Score decreases exponentially with an increase in the recursion depth. The recursion depth describes how many times decomposition is allowed to happen. Let's take a quick example once more to cement the underlying logic. Take a dataset being decomposed using the recursive partitioning function and a recursion depth of 2. The first decomposition might happen on the basis of the 2 possible values of some feature A, and thereafter each subsequent subset might be decomposed on the basis of the 3 possible values of some feature B, resulting in 6 subsets in the end. Each round of decomposition or partitioning results in more and more homogeneous subsets that would have a better and better BIC Score, since any GMM fitted would fit better - as hypothesized.

Figure 16
BIC vs. Recursion Depth



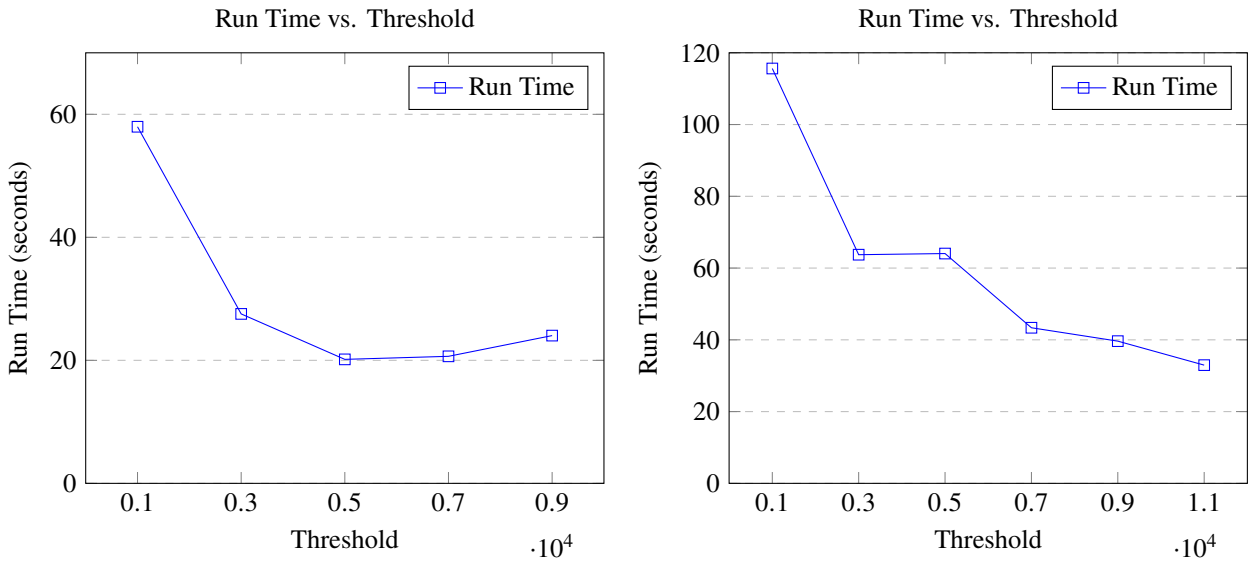
(a) *Start Times*

(b) *Durations*

Complexity Analysis

The complexity analysis illuminates the computational implications of varying te hyperparameters mentioned in the previous subsection, providing insights into the scalability and efficiency of our partitioning strategy. The graphs below in Figure 17 clearly show that the run time decreases with a decrease in the threshold value used, and stagnates at a point. This is largely explained by the fact that the higher the threshold, the quicker the recursive partitioning function will be stopped, and so the run time vs. threshold shows a logarithmic relationship between these variables.

Figure 17
Run Time vs. Threshold of the recursive segmentation

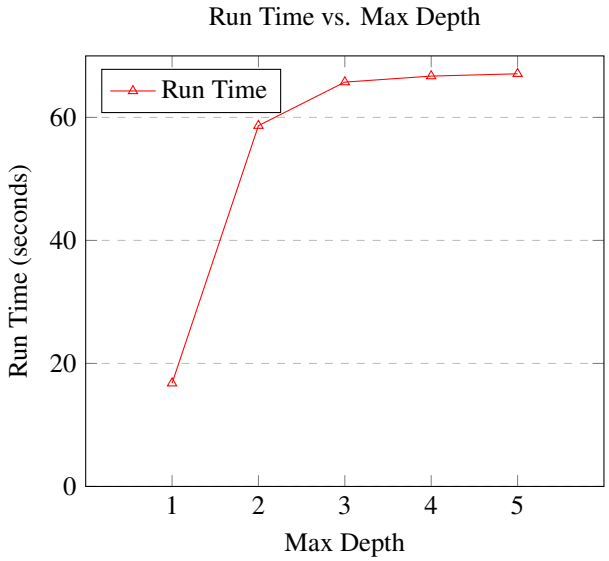


(a) *Start Times*

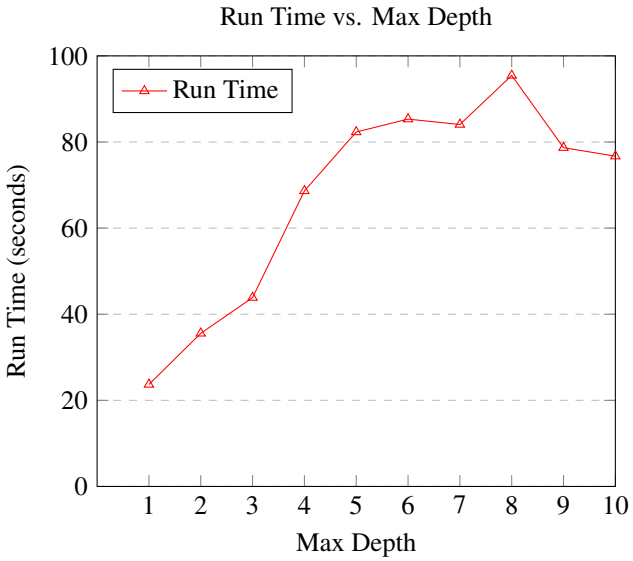
(b) *Durations*

In Figure 18 we show the relationship between the processing time and the maximum recursion depth. Here, the trend is consistent between the two cases and shows that the processing time increases with an increase in the maximum recursion depth - the higher the maximum recursion depth the longer the function is allowed to run, but this has an increasingly negligible effect the higher this maximum depth value is, since the threshold discussed earlier will likely cause the function to return a value by this point.

Figure 18
Run Time vs. Max Depth



(a) *Start Times*



(b) *Durations*

Discussion

This thesis has developed and evaluated a method for automating the analysis and prediction of execution and starting times within business processes in a generalized way. Below we discuss some of the main points on what this thesis has been able to deliver.

Contributions

By utilizing Gaussian Mixture Models (GMMs) and Decision Trees, it successfully identified patterns and relationships in business process data with greater accuracy than the naive approach that employs statistical distributions. The approach aims to enhance decision-making and resource optimization, particularly for resource-constrained organizations, and acknowledge the underlying randomness that forms part of all data from the real world, rather than making predictions in a way that assumes uniformity. The combination of GMMs for data segmentation and Decision Trees for feature identification has proven effective in generating accurate and generalizable predictive models. This method democratizes access to advanced business process analytics, enabling organizations without extensive data science expertise to benefit from these techniques. By using distributions rather than close-fitting functions to build a model, the result is a robust framework that can automatically uncover relationships between execution times and various parameters in business processes, and produce more robust models as a result.

Evaluation and Validation of Predictive Models

We conducted a rigorous evaluation of the predictive models using metrics such as the Bayesian Information Criterion (BIC). The results showed a significant improvement in model accuracy and generality at each step of the outlined method, though this has not been compared to other approaches that may be in use in the field. The decreasing BIC scores indicate that our approach effectively captures the underlying patterns in the data better than a naive approach based on GMMs would, leading to more reliable predictions.

Implications

The ability to predict execution times, along with many other temporal data features, and optimize resource allocation could have substantial implications for business process management. The use of

GMMs and Decision Trees provides a robust framework for analyzing complex business processes, enabling data-driven decision-making that can improve operational efficiency and effectiveness. This is not to say that the idea and method outlined in this thesis will always be well-suited for data exploration. We saw that for the duration times the data remained complex despite efforts to partition the dataset into more homogenous subsets. There will also be instances where business processes will be modelled better using an array of other methods out there. One such a case might be data that are so uniform and constant that they might be better modelled using supervised learning algorithms, since overfitting might not cause major issues in such a case. What this thesis, however, aimed to show is simply that the generality and accuracy of distributional regression based modelling techniques can be greatly improved and be relevant tools for prediction, especially in working towards the idea that that benchmark models can help less well-resourced players in an industry also make use of such insights and data analysis without much additional work.

Practical Applications

The method outlined in this thesis could be applied to various business scenarios, providing valuable insights that inform better decision-making by helping identify the overall distribution of target values in an accurate and well-defined model. This is particularly beneficial for smaller organizations that may lack the resources to conduct in-depth data analysis and might be able to rely on distributional models such as these, which might be able to serve as benchmark models, to base their decisions upon. For this to be possible, industry-level efforts to create and update such models would be necessary, and the feasibility would need to be assessed on an industry basis. By offering a cost-effective avenue for process optimization, the research in this thesis supports the broader adoption of advanced analytics in business process management, and the adoption of distribution-based analytics in particular.

Limitations

The accuracy and generality of the models depend on the quality, quantity and nature of the input data. We have seen that the method outlined in this thesis yields quite different results depending on the overall data - the starting times were modelled much easier than the duration times. In cases where data is sparse or highly irregular, the models may require further refinement and validation, or it might be appropriate to introduce further features in the dataset to be able to then employ my method more easily, since additional features would allow for the partitioning of the data into a

greater number of subsets. Additionally, the computational complexity associated with GMMs and recursive partitioning may present challenges for real-time applications, especially in large-scale business environments, though this is dependent on the parameter values used and the nature of the dataset to a large extent.

Future Research Directions

Future research should explore several key areas to enhance the robustness and applicability of this approach. It might be useful to introduce further features into a dataset to improve the partitioning of the dataset into more homogenous subsets, and develop techniques to evaluate when to do this. Integrating other clustering algorithms and supervised learning methods, rather than those used here, could provide further validation and refinement, and potentially lead to more useful models. Additionally, tweaking the ways in which the 'most important feature' is selected after each round of decomposition might enhance the results. Fuuture research could also compare a distribution-based approach like the one outlined in this thesis with other approaches used in business process management. Overall I believe future research that builds on the idea of combining statistical distributions with machine learning algorithms in a chain of steps that ultimately produces general rather than dataset-specific models could be a useful direction of research. Another potential area of future research might focus on the ability of this method or similar frameworks to help identify concept drift within particular organisations. Finally, a practical application not mentioned up to now may be for managers or decision-makers within organisations to distinguish between processes that are and are not well-modeled using a single gaussian component, and possibly conclude that those which are not well-modeled in this way may have structural inefficiencies causing this effect, rather than noise. Additional research into the viability of this hypothesis will however be necessary.

Conclusion

The research presented in this thesis sought to develop and validate a method for automating the generalized analysis and prediction of execution and starting times within business processes. By leveraging Gaussian Mixture Models (GMMs) and Decision Trees, we sought to uncover patterns and relationships in business process data that could enhance decision-making and resource optimization, particularly for resource-constrained organizations. Our approach promotes generality, ensuring models are not overfitted to specific datasets.

This method involves segmenting the data based on the most important features in a recursive approach and modeling the resulting subsets with GMMs. This improves the generality and accuracy of the models compared to a naive approach that uses the entire dataset. Evaluation results, using metrics such as the Bayesian Information Criterion (BIC), showed significant improvements in model accuracy and generality at each step. Fewer GMM components led to better BIC scores, indicating more generalizable insights.

The initial hypothesis was that subsets of homogeneous data, optimally modeled using normal distribution components, would provide more generalizable insights. This was confirmed by the decrease in BIC scores at each step, demonstrating that these data pockets produced better models than considering the entire dataset. This approach aims to identify trends that transcend individual datasets, aiding decision-making within organizations, and the data from the BPI 2017 Challenge can possible help to establish general rules of thumb for process execution and start times for the business cases to which the dataset relates.

Future research should explore the integration of additional or composite features to improve data partitioning, the use of alternative clustering algorithms and supervised learning methods, and further refinement of feature selection techniques. Comparing this distribution-based approach with other methods in business process management could also provide valuable insights, and identify which cases distribution-based frameworks might perform better than other frameworks employed in the field of business process management.

Overall, this thesis has shown that combining statistical distributions with machine learning algorithms in a systematic way can produce general models that are valuable for process optimization of temporal data.

Bibliography

[1] M. Weske, *Business Process Management*. Springer Berlin Heidelberg, 2012, ISBN: 978-3-642-28615-5. DOI: 10.1007/978-3-642-28616-2.

[2] M. Kajba and B. Jereb, "Process optimization of the selected business using a process approach," *European Journal of Studies in Management and Business*, vol. 23, pp. 1–17, Dec. 2022. DOI: 10.32038/mbrq.2022.23.01.

[3] W. van der Aalst, *Process Mining*. Springer Berlin Heidelberg, 2016, ISBN: 978-3-662-49850-7. DOI: 10.1007/978-3-662-49851-4.

[4] W. Aalst, "Process discovery from event data: Relating models and logs through abstractions," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 8, e1244, Dec. 2018. DOI: 10.1002/widm.1244.

[5] A. Rozinat, R. S. Mans, M. Song, and W. M. van der Aalst, "Discovering simulation models," *Information Systems*, vol. 34, pp. 305–327, 3 May 2009, ISSN: 0306-4379. DOI: 10.1016/J.IS.2008.09.002.

[6] L. Baier, J. Reimold, and N. Kühl, "Handling concept drift for predictions in business process mining," May 2020. DOI: 10.1109/CBI49978.2020.00016.

[7] Yuliya, Z. E. S. Konstantin, and Gaidamaka, "Analysis of business process execution time with queueing theory models," Alexander, N. Anatoly, Y. R. D. Alexander, and Gortsev, Eds., Springer International Publishing, 2016, pp. 315–326, ISBN: 978-3-319-44615-8.

[8] T. Schmitt, M. Bundscherer, R. Drechsel, and T. Bocklet, "Machine learning based optimization of a ceramic bushing manufacturing process," Dec. 2022, pp. 1–4. DOI: 10.1109/SENSORS52175.2022.9967124.

[9] IEEE Task Force on Process Mining, *2017 bpi challenge*, Accessed: 2023-11-26, 2017. [Online]. Available: https://data.4tu.nl/articles/dataset/BPI_Challenge_2017/12689204/1.

[10] K. Peffers, T. Tuunanen, C. Gengler, *et al.*, "The design science research process: A model for producing and presenting information systems research," *Proceedings of First International Conference on Design Science Research in Information Systems and Technology DESRIST*, Dec. 2006.

[11] A. Berti, S. van Zelst, and D. Schuster, "Pm4py: A process mining library for python," *Software Impacts*, vol. 17, p. 100 556, 2023, ISSN: 2665-9638. DOI: https://doi.org/10.1016/j.simpa.2023.100556. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2665963823000933.

[12] M. Dumas, M. L. Rosa, J. Mendling, and H. A. Reijers, *Fundamentals of Business Process Management*. Springer Berlin Heidelberg, 2018, ISBN: 978-3-662-56508-7. DOI: 10.1007/978-3-662-56509-4.

[13] M. H. Kutner, C. J. Nachtsheim, J. Neter, and W. Li, *Applied Linear Statistical Models*, 5th ed. McGraw-Hill, 2005, ISBN: 978-0073108742.

[14] "Nist/sematech e-handbook of statistical methods." (2022), [Online]. Available: https://www.itl.nist.gov/div898/handbook/.

[15] D. C. Montgomery, E. A. Peck, and G. G. Vining, *Introduction to Statistical Quality Control*, 7th ed. John Wiley & Sons, 2008.

[16] B. S. Everitt and T. Hothorn, *An introduction to generalized linear models*, 3rd ed. Chapman & Hall/CRC, 2016.

[17] K. Murphy, *Machine Learning: A Probabilistic Perspective* (Adaptive Computation and Machine Learning series). MIT Press, 2012, ISBN: 9780262304320. [Online]. Available: https://books.google.de/books?id=RC43AgAAQBAJ.

[18] W. Kratsch, J. Manderscheid, M. Röglinger, and J. Seyfried, "Machine learning in business process monitoring: A comparison of deep learning and classical approaches used for outcome prediction," *Business & Information Systems Engineering*, vol. 63, no. 3, pp. 261–276, 2021.

[19] O. López-Pintado and M. Dumas, "Business process simulation with differentiated resources: Does it make a difference?" In *Business Process Management - 20th International Conference, BPM 2022*, ser. Lecture Notes in Computer Science, vol. 13420, Springer, 2022, pp. 361–378.

[20] A. Berchtold, "Sequence analysis and transition models," in *Encyclopedia of Animal Behavior*, M. D. Breed and J. Moore, Eds., Oxford: Academic Press, 2010, pp. 139–145, ISBN: 978-0-08-045337-8. DOI: https://doi.org/10.1016/B978-0-08-045337-8.00233-3. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780080453378002333.

Appendix

Table 1
BIC Values for Different Implementations with Depth = 4 and Threshold = 1000

Implementation	BIC (Start Times)	BIC (Durations)
Overall Data	1014417.2828	584247.7657
Segmented Subsets	1032.8322	2592.4091
Single-Component-Optimal Subsets	545.5497	427.5077

Table 2
BIC vs. Recursion Depth at Threshold = 100

Recursion Depth	BIC (Start Times)	BIC (Durations)
1	5681.24	467168.71
2	718.33	82360.80
4	-8.55	2066.19
8	-14.18	67.97

Table 3
BIC vs. Threshold at Depth = 4

Threshold	BIC (Start Times)	BIC (Durations)
10	-14.18	2066.19
50	-7.66	2405.46
100	46.07	2841.18
500	454.40	4112.22
1000	1032.83	5626.01

Table 4
Run Time vs. Max Depth

Max Depth	Run Time (Start Times)	Run Time (Durations)
1	10.05	16.78
2	31.63	58.66
3	34.50	65.74
4	36.31	66.72
5	36.21	67.09

Table 5
Run Time vs. Threshold

Threshold	Run Time (Start Times)	Run Time (Durations)
1000	35.50	115.62
3000	14.40	63.72
5000	10.75	64.05
7000	10.56	43.37
9000	10.59	39.63